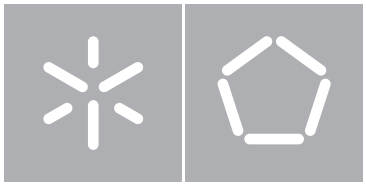
Universidade do Minho
Escola de Engenharia

Paulo Renato Dias Rodrigues

Data Warehouses suportados por Nuvens

Outubro de 2013



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Paulo Renato Dias Rodrigues

Data Warehouses suportados por Nuvens

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de
Professor Doutor Orlando Manuel de Oliveira Belo

Outubro de 2013

Anexo 3

DECLARAÇÃO

Nome

Paulo Renato Dias Rodrigues

Endereço electrónico: prdrodrigues@gmail.com Telefone: 253 273 576 / 917 950 468

Número do Bilhete de Identidade: 13768454

Titulo dissertação ☑/tese ☐

Data Warehouses suportados por Nuvens

Orientador(es):

Professor Doutor Orlando Manuel de Oliveira Belo

Ano de conclusão: 2013

Designação do Mestrado ou do Ramo de Conhecimento do Doutoramento:

Mestrado em Engenharia Informática

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE/TRABALHO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, 22/10/2013

Assinatura: Paulo Renato Dias Rodrigues

Agradecimentos

Ao meu orientador, professor Orlando Belo, que, graças à sua excelente capacidade de comunicação, conhecimento, experiência, dedicação e profissionalismo, me conduziu ao longo desta dissertação de forma sustentada e saudável. De realçar a sua disponibilidade, acompanhamento e preocupação demonstrada, bem como as boas orientações que sempre me concedeu.

Ao professor António Luís Sousa, por diversas vezes ter clareado as nuvens que sobre mim pairavam relativamente a parte do tema desta dissertação. Um agradecimento especial pela preciosa e indispensável ajuda, por vezes em horas extra, que me deu na parte prática deste projeto.

Aos meus amigos, que me acompanharam desde a licenciatura até a esta fase de mestrado. Pelos bons e inesquecíveis momentos vividos, de divertimento, partilha e colaboração, indispensáveis para ultrapassar as adversidades que foram surgindo neste trajeto. Uma menção particular ao Rui, ao States e ao Bernardo, que acompanharam mais de perto a realização desta dissertação.

Aos meus pais, pelo natural apoio demonstrado e esforço realizado, para que eu conseguisse culminar mais uma etapa da minha vida académica e pessoal.

Por último, a todos aqueles que, apesar de não estarem aqui explicitamente referidos, me ajudaram e contribuíram direta ou indiretamente para a realização desta dissertação.

Resumo

Data Warehouses suportados por Nuvens

O universo das Tecnologias de Informação está a assistir a grandes mudanças desde o surgimento do conceito de *cloud computing*. A *cloud computing* revela-se como um meio que possibilita a fácil aquisição e liberação (elasticidade) de recursos computacionais, que disponibiliza infraestruturas altamente escaláveis e dispendiosas com o mínimo de configuração possível e, ainda, pelo facto de ser um serviço com custos reduzidos comparativamente a uma solução *in-house*, pois tipicamente utiliza um modelo *"pay-as-you-go"*. Claro que com a delegação de toda a infraestrutura e dos dados para um provedor de *clouds*, questões como a segurança e privacidade dos dados começaram a ser equacionadas, apresentando-se assim como desvantagens para soluções em *cloud*. No entanto, além da *cloud computing*, outras variáveis, como o grande aumento do volume de dados nas empresas e os avanços tecnológicos alcançados nas redes de banda larga, têm "exigido" a adaptação das bases de dados para um ambiente em *cloud*, o que originou, pouco a pouco, o paradigma de *Database as a Service*. Atualmente ainda existem dúvidas relativamente às bases de dados SQL, sobre se estas serão as mais indicadas para ambientes *cloud*. Este modelo de bases de dados tem dominado o mercado mas apresenta diversas limitações (por exemplo a nível de escalabilidade e garantia das propriedades ACID) quando confrontadas com implementações num ambiente *cloud*. Por outro lado, o ecossistema de aplicações desenvolvidas com bases de dados SQL é demasiado grande para ser modificado para outro modelo. Apesar desta indefinição, a *cloud* parece ser um cenário ideal para *data warehouses* pois são bases de dados que albergam usualmente enormes volumes de dados e são essencialmente de leitura.

Com esta dissertação pretendeu-se estudar a viabilidade da implementação e migração de um sistema de *data warehousing* para um ambiente *cloud* e apresentar um protótipo que expusesse a utilidade do mesmo face a uma típica implementação *in-house*.

Abstract

Cloud Based Data Warehouses

The world of Information Technology is witnessing major changes since the appearance of the cloud computing concept. Cloud computing reveals itself as a means to allow easy acquisition and release, in other words elasticity, of computing resources, provides highly scalable and costly infrastructures with minimal configuration and also because it is a service with reduced costs compared to an in-house solution, because it typically uses a "pay-as-you-go" model. Of course, with the delegation of the entire infrastructure and the data to a cloud provider, issues such as security and privacy of data began to be addressed, becoming drawbacks to cloud solutions. However, in addition to cloud computing, other variables, such as the large increase of enterprise data volumes and the technological advances in broadband networks, have required the adaptation of databases to a cloud environment, which led, step by step, to the Database as a Service paradigm. Currently there are still doubts whether SQL databases will be the most suitable model for cloud environments. While this database model has dominated the market, it has several limitations (eg. in terms of scalability and assurance of the ACID properties) when confronted with implementations in a cloud environment. On the other hand, the ecosystem of applications developed under SQL databases is too large to be changed to another model. Despite this uncertainty, the cloud seems to be an ideal environment for data warehouses because these are databases that usually house huge volumes of data and are essentially used for reading purposes.

The purpose of this dissertation was to study the feasibility of implementation and migration of a data warehousing system to a cloud environment and to develop a prototype that would expose its usefulness compared to a typical in-house implementation.

Índice

Índice de Figuras

Índice de Tabelas

Siglas e Acrónimos

API	Application Programming Interface
BI	Business Intelligence
CLI	Command Line Interface
DaaS	Data as a Service
DBaaS	Database as a Service
DW	Data Warehouse
DWaaS	Data Warehouse as a Service
IaaS	Infraestructure as a Service
JADE	Java Agent Development Framework
MDX	MultiDimensional eXpressions
MPP	Massively Parallel Processing
NIST	National Institute of Standards and Technology
OLAP	On-line Analytical Processing
OLTP	Online Transaction Processing
PaaS	Platform as a Service
SaaS	Software as a Service
SAN	Storage Area Network

SD	Shared-Disk
SDW	Sistema de Data Warehousing
SGBD	Sistema de Gestão de Bases de Dados
SLA	Service-Level Agreement
SN	Shared-Nothing
SO	Sistema Operacional
SQL	Structured Query Language
XML	Extensible Markup Language

Capítulo 1

Introdução

1.1 A gestão de dados numa *cloud*

As bases de dados relacionais têm dominado o mercado ao longo das últimas décadas, apresentando-se ainda hoje como principal solução para o armazenamento de dados resultantes dos processos de negócio das empresas. Com o enorme crescimento da informação digital, as empresas começaram a encarar estes dados como sendo de análise, utilizando-os para a resolução de problemas, tomadas de decisão ou planeamento de estratégias de negócio. Assim, passou a ser necessário a utilização de bases de dados (como os *data warehouses*) que fossem capazes de armazenar toda a informação atual e de histórico proveniente de diversos sistemas operacionais (SOs) empresariais. A construção deste tipo de bases de dados é vulgarmente muito dispendiosa, pois estas necessitam dum elevado número de recursos para armazenar os volumes de dados envolvidos e fazer a sua consequente disponibilização de uma forma expedita. É nesta perspetiva que o aparecimento da *cloud computing* – computação baseada em nuvens - tem levado a uma mudança de paradigma não só no panorama tecnológico, mas também no panorama das bases de dados ditas convencionais. A *cloud* permite às empresas utilizar infraestruturas poderosas a custos acessíveis, obter ou libertar facilmente recursos computacionais a pedido e ausentar-se de quaisquer questões relacionadas com a manutenção de infraestruturas, de modo a que tenham apenas de se preocupar com o mais importante: os dados. Daí cresceu, naturalmente, a necessidade de adaptação das bases de dados como um serviço através da *cloud*. Como

consequência, surgiram diversos conceitos relacionados com a gestão de dados neste ambiente, como *Data as a Service* (DaaS) ou *Database as a Service* (DBaaS). DaaS é uma estratégia de *cloud* que disponibiliza dados referentes a um dado negócio, de modo a poderem ser utilizados para análise e estudo de tendências num contexto específico. Permite assim aceder a dados úteis de negócio de forma rápida, segura, com custos reduzidos e a partir de qualquer lugar. Apesar de relacionado, o nosso foco está na estratégia DBaaS pois permite aceder e implantar bases de dados na *cloud*, oferecendo ainda todas as suas funcionalidades e serviços inerentes. Além destes, também existem soluções que se caracterizam por disponibilizar uma dada quantidade de armazenamento na *cloud*, na qual se pode fazer uma gestão básica de dados e, por exemplo, armazenar cópias de segurança.

Em princípio, numa *cloud* qualquer podem-se implantar as tradicionais bases de dados como PostgreSQL[1], MySQL[2], Oracle[3], IBM DB2[4], etc. Neste caso, os provedores da *cloud* vendem máquinas virtuais aos utilizadores para que estes possam instalar, configurar e gerir as suas bases de dados. Uma alternativa a este tipo de oferta é a escolha de uma solução utilizando DBaaS. Nesta alternativa, a instalação, a configuração e a manutenção da base de dados fica ao cargo do provedor do serviço, pagando o utilizador o serviço de acordo com a sua utilização. O Amazon RDS[5] (*Amazon Relational Database Service*) é um exemplo de uma solução DBaaS que oferece acesso a bases de dado MySQL, Oracle ou SQL Server[6].

Apesar das vantagens já evidenciadas, a implantação de bases de dados numa *cloud* levou ao aparecimento de novos problemas. Por norma, um sistema de bases de dados pode utilizar uma das seguintes arquiteturas (Lee, 2011):

- *shared-nothing,* na qual cada servidor de base de dados possui o seu próprio disco e gere uma parte única dos dados;
- *shared-disk*, na qual diferentes servidores partilham o mesmo disco, o que faz com que cada um tenha acesso a todos os dados.

[1] http://www.postgresql.org/
[2] http://www.mysql.com/
[3] http://www.oracle.com/us/products/database/
[4] http://www-01.ibm.com/software/data/db2/
[5] http://aws.amazon.com/en/rds/
[6] http://www.microsoft.com/sqlserver/

Como os sistemas de bases de dados transacionais utilizam tipicamente uma arquitetura *shared-disk*, estes não são os mais adequados para serem instalados numa *cloud,* pois não são escaláveis e porque é difícil de garantir as propriedades ACID em ambientes distribuídos, como é o caso da *cloud*. Além disto, problemas de desempenho, de segurança ou de privacidade de dados também são referenciados. Uma arquitetura *cloud* certamente introduzirá um maior *overhead* nas comunicações do que numa solução típica *in-house*. Relativamente à segurança e privacidade dos dados, estes são confiados e guardados em infraestruturas fornecidas pelos provedores de serviços na *cloud*. Como as bases de dados transacionais registam informação relativa a processos de negócio, estas podem conter alguma informação sensível e privada, que precise de ser submetida a mecanismos de segurança. Apesar dos provedores de serviços na *cloud* poderem fornecer mecanismos de segurança, como a encriptação dos dados, o processo de desencriptação iria, assim, aumentar a carga no CPU, podendo mesmo aumentar significativamente os tempos de resposta do sistema. Atendendo a outra perspectiva, os sistemas de gestão de dados analíticos, como é o caso dos sistemas de *data warehousing* (SDWs), beneficiam da utilização de uma arquitetura *shared-nothing* devido à sua escalabilidade, que por sua vez, e graças a essa mesma característica, é uma arquitetura adequada para utilização na *cloud*. Além disso, como sabemos, um DW possui características especiais quando comparados com as bases de dados transacionais. Primeiro, os dados considerados sensíveis ou privados não são guardados num DW pois não serão importantes para a análise, diminuindo-se assim os problemas de segurança dos dados numa implementação em *cloud*. Segundo, um DW é considerado como sendo apenas de leitura, sendo o sistema atualizado apenas com a execução dos típicos processos ETL. Em terceiro lugar, como o sistema é atualizado com informação proveniente dos sistemas de dados transacionais, esses dados estão à partida consistentes, não havendo assim necessidade de verificar as propriedades ACID sobre um DW (Abadi, 2009). Por estes motivos, parece que os sistemas de gestão de dados analíticos serão mais facilmente implementados em ambientes *cloud*.

1.2 Motivação e objetivos

O crescente aumento de informação e a necessidade de análise dos dados para suporte a atividades de tomada de decisão têm levado as empresas a optar pela implementação de DWs de modo a conseguirem "transformar" a sua informação em conhecimento útil. A construção deste tipo de sistemas passa, tipicamente, por um conjunto de fases bastante diverso devido à complexidade envolvida e porque se pretende que este seja um processo rigoroso de modo a

evitar o insucesso do mesmo. Tratando-se de um sistema complexo, que gere usualmente enormes quantidades de dados, a sua construção exige que sejam disponibilizados imensos recursos, quer estes sejam *hardware*, *software* ou mesmo humanos. Muitos dos custos envolvidos com o projeto e implementação de um DW podem ainda ser considerados como custos iniciais, necessários uma vez na instalação e arranque do sistema, ou então considerados como custos recorrentes, quando associados à manutenção do mesmo (Inmon, 2000). A envolvência de todos estes recursos faz com que a construção de DWs seja um processo muito dispendioso, o que impede muitas pequenas e médias empresas de optarem pela sua implementação. Mesmo independentemente da dimensão da empresa, a construção de um DW deve ser devidamente justificada e planeada devido aos custos envolvidos, pois quanto maior forem as necessidades, maior serão também os custos e os riscos associados. Atendendo a estes problemas, seria vantajoso para as empresas que o processo de construção de um DW visse os seus custos e riscos reduzidos sem afetar a qualidade de serviço do sistema desenvolvido.

É, assim, baseada nesta perspetiva que a *cloud* pode também revolucionar a área de *Business Intelligence* (BI). Seria óptimo poder trazer as vantagens da *cloud* para o domínio dos DWs e, à semelhança do conceito DBaaS, utilizar uma solução de *Data Warehouse as a Service* (DWaaS). Com isto, poder-se-ia libertar as empresas praticamente da totalidade dos custos de implementação relacionados com *software* e *hardware* e reduzir significativamente os custos dos recursos humanos, uma vez que seriam apenas necessários profissionais que tratassem da gestão do projeto, dos requisitos do sistema, do desenho dos dados e finalmente da implementação. Qualquer outro tipo de questões (nomeadamente aquisição, configuração e manutenção de infraestruturas físicas) e respetivos recursos envolvidos ficariam apenas ao cargo dos provedores da *cloud*. Além dos custos, também o tempo de implementação seria reduzido drasticamente pois a *cloud* providenciaria rapidamente a infraestrutura necessária.

As vantagens apresentadas estão sobretudo relacionadas com o custo de construção de um DW. Porém as características como a escalabilidade ou elasticidade da *cloud* são também mais valias durante a execução deste tipo de sistemas. Claro que, com a possibilidade de uma solução DWaaS, não se pretende eliminar ou diminuir a necessidade de fundamentação e justificação da construção deste tipo de sistemas, mas sim diminuir o impacto dos custos da implementação e gestão de um DW em qualquer organização.

1.3 Estrutura da dissertação

Tendo em conta os dois assuntos principais abordados, *cloud computing e data warehousing*, justifica-se que esta dissertação esteja organizada de forma a possibilitar ao leitor uma suave e gradual transição entre estes dois conceitos, começando por uma explicação e definição do conceito de *cloud computing* em geral, seguindo depois por uma abordagem que especifique a sua aplicação no âmbito dos sistemas de suporte à decisão, mais concretamente no contexto dos SDWs. Neste sentido, a presente dissertação está estruturada e dividida em três principais capítulos.

A introdução e explicação geral do conceito de *cloud computing* é relatada no segundo capítulo. Este começa com uma breve referência à história e ao aparecimento deste conceito, aparecendo na secção seguinte uma definição para o mesmo. As restantes secções são utilizadas para fazer uma desmistificação do conceito. Inicialmente é feita uma comparação deste modelo de computação com outros previamente surgidos, como *utility computing*, *cluster computing* e *grid computing*. De seguida é abordado o ambiente em torno da *cloud computing*, identificando os principais atores envolvidos, características e benefícios que levam à sua adoção, bem como os diversos serviços e formas que permitem a sua disponibilização. De referir ainda que, todas as fontes utilizadas para rechear este capítulo foram maioritariamente artigos científicos e técnicos, essencialmente de investigação num meio académico, publicados em diversas conferências relacionadas com as *clouds* e problemática subjacente.

O terceiro capítulo tem como finalidade fazer a ligação entre SDWs e a *cloud*, abordando assim a questão específica de implementação deste tipo de sistemas nesse ambiente. Como o propósito desta dissertação não era a explicação pormenorizada dos SDWs, a primeira secção limita-se a expor algumas noções gerais sobre este tipo de sistemas. Em termos práticos, além da principal questão dos DWs, esta dissertação abrange um problema muito específico e crucial que é a implementação de bases de dados relacionais na *cloud*, e, como tal, é feita a distinção e descrição dos dois preponderantes sistemas que utilizam esta forma de armazenamento: sistemas transacionais e sistemas de gestão de dados analíticos. Os aspetos evidenciados nesta primeira secção são cruciais para, nas seções seguintes, se perceber quais as necessidades, vantagens e alguns dos problemas existentes, quando estes sistemas são confrontados com uma implementação na *cloud*. Posto isto, nas duas seções seguintes, é exposta a necessidade da utilização deste ambiente computacional para os DWs, sendo apresentadas diversas soluções do

mercado propícias à gestão de dados na *cloud*. Esta última secção cobre praticamente a totalidade do espectro dos diversos tipos de soluções de gestão de dados existentes na *cloud*, começando por apresentar os serviços que permitem a gestão simplificada de dados e terminando naqueles que realmente revelam interesse no âmbito desta dissertação, ou seja, as soluções para a utilização de bases de dados e DWs na *cloud*. De seguida é tratada de forma específica o problema da mudança destes sistemas para a *cloud*. Como são um componente essencial da sua base, a primeira secção começa por identificar um conjunto de requisitos e características que se pretendem ver nas bases de dados, quando implementadas num ambiente em *cloud*. Os entraves e questões que dificultam a garantia destas características são apresentados na secção seguinte, sendo abordadas questões relacionadas com o tipo de arquitetura de base de dados a utilizar, questões de desempenho, estabilidade, privacidade e segurança de informação neste ambiente, etc. A mudança para a *cloud* culmina com o contraste de todos os problemas evidenciados e o seu impacto em sistemas transacionais e nos DWs, de forma a perceber-se qual deste tipo de sistemas está mais adequado para um mudança para este novo ambiente. Este capítulo termina com a apresentação de uma mais valia e possibilidade para a implementação dos DWs na *cloud*, mencionando a utilização de uma *cloud* privada de forma a reduzir ou minimizar alguns dos problemas e riscos previamente apresentados nas secções transatas. De referir que, como a tentativa de implantação de DWs na *cloud* nasceu e cresceu, principalmente, das necessidades existentes no ambiente empresarial e não num meio académico, na sua maioria, os artigos utilizados na elaboração deste capítulo caracterizam-se por ser relatórios técnicos e guias elaborados precisamente pelas próprias empresas que enfrentaram/enfrentam esta problemática.

O caso prático, que se caracteriza essencialmente pela implementação de um sistema de suporte à decisão com recurso a ambientes em *cloud*, é retratado no capítulo quatro. Inicialmente é feita uma breve contextualização prática, na qual é explicada a origem, a motivação e o propósito que levaram à realização do projeto em causa. Esta parte "introdutória" termina com uma breve explicação do funcionamento geral do sistema desenvolvido, revelando os diversos componentes que o constituem e que serão descritos detalhadamente ao longo do resto do capítulo. Na sua parte restante, o capítulo está dividido em duas grandes partes. A primeira descreve todo o processo que foi necessário realizar para instanciar e configurar uma *cloud* e os recursos/componentes associados. Inclusive, são mesmo apresentados guias passo a passo das tarefas que foram necessárias realizar para este efeito. A segunda parte revela detalhes específicos da implementação do projeto no que diz respeito à interação e utilização da *cloud* como fonte de

informação global para o sistema desenvolvido. Essencialmente é descrito o seu habitual funcionamento através da explicação dos processos de dados a si associados, nomeadamente para a angariação, conciliação e provisionamento de dados no sistema.

Por fim, o quinto capítulo. Este apresenta algumas conclusões sobre o trabalho que foi realizado nesta dissertação, quer este seja de cariz teórico ou prático. Inicialmente é feita uma pequena revisão e constatação das vantagens estudadas mediante a utilização de soluções em *cloud*, assim como os problemas abordados referentes ao caso específico dos DWs. No final do capítulo encontra-se um resumo do projeto prático realizado e uma breve apreciação pessoal sobre o mesmo. Nele são ainda identificados alguns desafios e obstáculos enfrentados durante a execução deste caso prático, bem como algumas linhas de orientação a realizar num futuro próximo.

Capítulo 2

A Computação "na nuvem"

A *cloud computing* apresenta-se como um conceito que tem estado a revolucionar o mundo das TI. É o "conceito da moda" no domínio das TI. Um conceito que tem ganho cada vez mais destaque e importância tanto a nível empresarial como académico. Até para uso pessoal tem tido bastante aceitação, apesar deste tipo de utilizadores não serem, à partida, considerados *early adopters*. A *cloud computing* surge assim como uma solução aliciante e necessária num mundo que assiste ao contínuo e massivo aumento da informação digital. "*The world is online*" ou "*The world is mobile*" são algumas das frases que marcam esta era digital que, por sua vez, exige que as TI tenham uma elevada capacidade de resposta e adaptação de modo a ser possível apresentar soluções face às necessidades e problemas inerentes aos serviços disponibilizados hoje em dia. Entre outros, *Software as a Service* (SaaS), *Platform as a Service* (PaaS), *Infraestructure as a Service* (IaaS) e *Data as a Service* (DaaS) são alguns dos diferentes modelos de serviços que acompanham o conceito de *cloud computing* e que prometem ser uma solução para as necessidades existentes. De facto, foram muitas as vantagens anunciadas e, na prática, várias já verificadas como é o caso de tempos de implementação reduzidos, fácil e rápido escalonamento de recursos a pedido e custos de implementação também significativamente menores comparativamente com soluções tradicionais. A *cloud* evoluiu assim desde um mero conceito idealizado e publicitado como uma solução promissora para uma tecnologia real e emergente.

2.1 A concretização de visões antigas

Embora pareça recente, a *cloud computing* não é um conceito novo. Ao longo da história, muitas foram as visões enunciadas que podiam ser relacionadas com a origem deste conceito. Uma possível teoria data dos anos 60 (Buyya et al., 2009) (Timmermans et al., 2010). Nessa altura, à medida que os computadores se foram tornando cada vez mais importantes e necessários, começaram-se a estudar técnicas que permitissem disponibilizar capacidade de computação em grande escala e para diversos utilizadores através da partilha do tempo de processamento. Em 1961 numa palestra dada no Instituto de Tecnologia de Massachusetts (MIT) sobre *"Time-Sharing Computer Systems"*, John McCarthy enunciava um novo paradigma de computação defendendo que, um dia, a capacidade de computação poderia e seria fornecida como um serviço público à semelhança de outros já existentes, como água, eletricidade ou gás (McCarthy, 1961). Este serviço passaria assim a provisionar o nível de computação necessário para satisfazer as necessidades gerais de cada utilizador. Em 1969, também Leonard Kleinrock, que esteve envolvido no projeto ARPANET[7], partilhava desta visão referindo que, apesar de na altura as redes de computadores ainda estarem num estado embrionário, à medida que fossem crescendo e se desenvolvendo, poder-se-ia assistir à propagação de serviços públicos de computação que iriam servir casas individuais e escritórios em todo o país (Kleinrock, 2003). Até aos dias de hoje, vários paradigmas de computação surgiram e prometeram respeitar estes ideais, como é o caso da *Utility Computing* e da *Grid Computing*.

Os avanços tecnológicos verificados desde os anos 60 e as recentes melhorias nos serviços de banda larga facilitaram o aparecimento da *cloud computing* e o seu provisionamento em grande escala. É difícil indicar quando foi a primeira vez que se ouviu este termo, mas é possível apontar alguns marcos que certamente ajudaram na sua definição e evolução globalmente (DataZion, n.d.). Em 1999, o aparecimento da empresa Salesforce.com[8] introduziu, pela primeira vez, a perspetiva de fornecimento e acesso a software empresarial[9] através dum simples *browser* (SaaS). Um dos seus objetivos era também introduzir um novo modelo de negócio baseado em subscrições no qual os clientes pagariam apenas o que necessitassem e utilizassem[10]. Mais tarde, em 2002, foi

[7] *Advanced Research Projects Agency Network (ARPANET),* foi a primeira rede operacional desenvolvida com base no paradigma de comutação de pacotes e considerada o progenitor da Internet.
[8] http://www.salesforce.com/
[9] Tradução do inglês: *Enterprise Software.*
[10] Também conhecido em inglês como modelo *pay-as-you-go.*

a vez da Amazon lançar o Amazon Web Services[11], que era um conjunto de serviços de computação fornecidos remotamente através da Internet. Quatro anos mais tarde seguiu-se o lançamento dos serviços Amazon Simple Storage Service[12] (S3) e Elastic Compute Cloud[13] (EC2), sendo que o primeiro possibilitava a disponibilização de espaço para alojamento de dados na *cloud*, enquanto que o segundo permitia aos utilizadores alugarem a infraestrutura necessária para instanciarem máquinas virtuais nas quais pudessem executar as suas próprias aplicações (IaaS). Foi também em 2006 que a Google anunciou o Google Docs[14], um serviço para gestão e partilha de documentos que certamente direcionou as atenções para a *cloud*. Aliás, apesar de não haver consenso sobre quem estabeleceu o termo *cloud computing*, muitos parecem acreditar que Eric Schmidt, o CEO da Google, deu um grande contributo para tal ao associá-lo em 2006 aos seus produtos (Willis, 2009). A Amazon contribuiria também para a aceitação do termo ao colocá-lo no nome do seu serviço EC2 que fora lançado de seguida nesse mesmo ano. Os anos seguintes despoletaram diversas investigações neste ramo. Em 2008, como resultado de um projeto de investigação, assistiu-se ao lançamento do OpenNebula[15], o primeiro *software open-source* que auxilia e agiliza a criação de *clouds* privadas e híbridas. O Windows Azure[16] (PaaS e IaaS) ditou em 2009 a entrada da Microsoft no mundo da *cloud*. Outros gigantes da informática, como a Oracle, HP, ou Teradata, também apesentaram posteriormente as suas soluções, dinamizando e auxiliando a mesma enquanto tecnologia emergente. Em Portugal, o CloudPT[17] é um serviço de alojamento de ficheiros na *cloud* e uma das mais recentes soluções a nível nacional, tendo sido lançado no final de 2012 pela Portugal Telecom.

A adoção da *cloud* por parte de grandes empresas de informática e a explosão de serviços disponibilizados através dela ajudaram a *cloud computing* a ser uma das principais tendências no panorama da informática atual, fazendo ainda disparar as expectativas em torno do mesmo. Também nos últimos anos, novas empresas (*startups*) começaram a aderir aos serviços da *cloud,* contribuindo para o aumento da sua notoriedade e importância. Essas empresas começaram a usufruir de várias vantagens anunciadas, abrindo assim uma nova janela de oportunidades, na qual é possível uma rápida implementação de ideias inovadoras capazes de rivalizarem com

[11] http://aws.amazon.com/
[12] http://aws.amazon.com/en/s3/
[13] http://aws.amazon.com/en/ec2/
[14] http://docs.google.com/
[15] http://opennebula.org/
[16] http://www.windowsazure.com/
[17] https://cloudpt.pt/

soluções de outras organizações há muito estabelecidas em diversos mercados. No entanto, à medida que a *cloud* demora a cumprir todas as expectativas anunciadas, diversas questões e problemas vão surgindo, revelando-se assim inibidores deste tipo de soluções. Estudos recentes revelam mesmo que o conceito geral de *cloud computing* está a perder interesse e entusiasmo entre a comunidade (Cohen, 2012) (Gartner, 2012). Relativamente a outras vertentes associadas ao *cloud computing*, mais concretamente SDWs na *cloud*, 2012 foi um ano marcado pela apresentação de diversas soluções sonantes (ex.: Amazon Redshift[18] e Treasure Data Cloud Data Warehouse[19]) e 2013 promete também ser, no mínimo, recheado de eventos e novidades que dão resposta à necessidade natural de implementação/migração dos SDWs para a *cloud*.

2.2 O que é a *cloud computing*

Desde um sonho de longa data que ditava o fornecimento de poder de computação como um serviço até à palavra da moda e mais recente tendência no mundo das TI, a *cloud computing* afirmou-se como uma solução credível e vantajosa. Devido a estes aspetos, muito se tem falado e discutido sobre este conceito, fazendo com que, por vezes, não haja uma opinião consensual em relação a si. Ao longo do seu ainda curto período de vida, foram aparecendo várias definições e interpretações para o termo *cloud computing*, e muito se especulou sobre aquilo que pode ser ou não uma *cloud*.

De uma maneira geral, a *cloud computing* pode ser simplesmente descrita como o provisionamento de recursos escaláveis e elásticos a consumidores através da Internet. Por outro lado, e de um ponto de vista de utilização, pode ser descrito como a possibilidade do utilizador aceder a ficheiros, dados, aplicações e outros serviços através da Internet (Kim, 2009). Segundo a Gartner, é um estilo de computação através do qual se disponibilizam funcionalidades das TI como um serviço a consumidores externos através da Internet (Plummer et al., 2008). Por se tratar de um conceito bastante abrangente e, de uma maneira geral, se basear no provisionamento de recursos de computação, isto poderá ter ajudado a esta explosão de definições e consequente variedade de opiniões, levando mesmo à sua associação com outros conceitos similares que surgiram anteriormente - por exemplo, o termo *cloud computing* é muitas vezes confundido ou referido como sendo *Grid Computing* ou *Utility Computing*.

[18] http://aws.amazon.com/en/redshift/
[19] http://www.treasure-data.com/

De forma a clarificar o conceito, o Instituto Nacional de Normas e Tecnologia (National Institute of Standards and Technology - NIST) elaborou uma proposta que teve como objetivo definir uma base para discussão sobre *cloud computing* e para a comparação geral dos diversos serviços de *cloud* existentes bem como das respetivas estratégias de implantação. Esta proposta teve boa aceitação na indústria, e muitas empresas já a referenciaram e a utilizaram em alguns dos seus artigos (Oracle, 2010) (Madsen, 2012). De acordo com o NIST, a *cloud computing* é um modelo que visa permitir, através da rede, o acesso a pedido, ubíquo e conveniente a um conjunto de recursos partilhados e configuráveis (ex. redes, servidores, armazenamento, aplicações e serviços). Esses recursos podem ainda ser rapidamente provisionados e libertados com o mínimo esforço de gestão por parte do fornecedor do serviço (NIST, 2011). Relativamente à *cloud*, esta pode ser simplesmente interpretada como sendo um ambiente de execução no qual todos estes recursos residem e através da qual se disponibilizam os serviços requeridos. A *cloud* caracteriza-se ainda por ser um ambiente elástico e fiável que coloca os seus recursos à disposição do utilizador de acordo os requisitos existentes no momento.

2.3 Conceitos ou paradigmas similares

Já desde 1961 que as palavras de John McCarthy levaram ao aparecimento de diversos paradigmas, todos com o objetivo de concretizarem a mesma visão e provisionarem assim capacidade de computação como um serviço. Essencialmente, esta diversidade foi uma consequência dos limites existentes tanto a nível de hardware e software, sendo que muitos paradigmas tentavam concretizar a mesma visão mas com implementações distintas. Apesar de susceptível, a *cloud computing* não deve ser considerada um substituto de qualquer paradigma anterior só porque a sua origem e tendência são mais recentes. Inclusive, na literatura, há quem considere a *cloud computing* como um conceito que engloba e abrange alguns dos paradigmas que surgiram anteriormente e ao qual lhe são associados.

Dos diversos paradigmas e conceitos que foram surgindo, *utility*, *grid* e *cluster computing* parecem ser os mais semelhantes e dos mais associados quando se fala de *cloud computing*. Em comum, todos estes paradigmas se caracterizam por serem exemplos de computação paralela ou distribuída, e por oferecerem elevada capacidade de computação necessária para a execução de tarefas pesadas e dispendiosas a nível computacional. Por estes motivos, de seguida são

apresentadas algumas noções gerais sobre estes paradigmas, de forma a perceber o que são e o que os assemelha e/ou distingue da *cloud computing*.

2.3.1 *Utility Computing*

O termo *utility computing* é exatamente a visão proferida por John McCarthy em 1961. É simplesmente um conceito que enuncia o provisionamento de capacidade de computação como uma utilidade ou um serviço à semelhança de outros serviços públicos já existentes. Assim como a base da *cloud computing*, caracteriza-se por providenciar um serviço ao cliente consoante as suas necessidades e a custos reduzidos. O cliente consegue assim aceder a soluções computacionais, através da Internet ou de uma rede privada, sempre que quiser e quando quiser. A *cloud computing* contempla e está assente nestes mesmos princípios, pelo que facilmente se percebe o porquê de muitas vezes ser confundido ou associado como sendo um sucessor ou uma versão mais recente de *utility computing*. Na realidade, e na literatura, considera-se que *utility computing* seja mais um modelo de negócio do que propriamente uma tecnologia ou um paradigma específico.

2.3.2 *Cluster Computing*

Durante vários anos, os designados supercomputadores foram os líderes no domínio da computação. De uma maneira geral, são máquinas autónomas com grande capacidade de computação, bastante dispendiosas e que precisam de uma grande quantidade de energia para terem um funcionamento correto (Abah & Ogwueleka, 2013). Por estes motivos, mais tarde começaram a ser substituídos por *clusters*, pois estes também permitem obter grande poder de computação, mas de uma forma mais barata. Um *cluster* é um conjunto de máquinas com capacidade de computação paralela ou distribuída que estão interligados entre si tipicamente através de uma rede local (LAN[20]). Caracterizam-se ainda por serem máquinas homogéneas, ou seja, que têm o mesmo tipo de hardware e sistema operativo (Buyya et al., 2009) (Sadashiv & Kumar, 2011). Estas máquinas cooperam na execução de tarefas que não seriam possíveis de realizar nos tradicionais computadores individuais, como por exemplo tarefas que necessitem de grande poder de computação ou que exigem o processamento de grandes volumes de dados. Do

[20] LAN – Local Area Network

ponto de vista do utilizador existem diversas máquinas que funcionam como se se tratassem de uma só.

Apesar de ambos os paradigmas possibilitarem grande capacidade computacional, existem algumas diferenças entre *cluster* e *cloud computing.* Por exemplo, e ao contrário daquilo que acontece nos *clusters*, as máquinas que suportam a infraestrutura de uma *cloud* podem estar separadas geograficamente e caracterizam-se normalmente por serem máquinas heterogéneas. Além disto, o foco da *cloud* está no provisionamento de recursos computacionais sob a forma de um serviço, ao passo que os *clusters* são normalmente implementados para aumentar o desempenho e a disponibilidade de um dado serviço ou processo.

2.3.3 *Grid Computing*

O paradigma de *grid computing* é bastante semelhante ao de *clusters* já apresentado, sendo muitas vezes considerado como uma sua evolução. De uma perspectiva geral, *grid computing* consiste em aplicar os recursos de diversas máquinas existentes numa dada rede para a resolução de um único problema. Um exemplo de aplicação deste paradigma é a iniciativa SETI@home[21] que, através da Internet e com a execução de um pequeno software, possibilita que cada utilizador contribua com recursos computacionais do seu próprio computador pessoal para a procura de inteligência extraterrestre. Apesar de semelhante, no caso dos *clusters* assume-se que as máquinas envolvidas são homogéneas e estão distribuídas localmente e acessíveis através de uma LAN. Porém, na computação em *grid* podem ser utilizadas máquinas de qualquer parte do mundo, o que possibilita a sua caracterização como um paradigma que utiliza máquinas heterogéneas, que podem estar distribuídas globalmente (Buyya et al., 2009). Todo o *software*, incluindo os sistemas operativos, que está a ser executado nessas máquinas pode também ser diferente. Aliás, para o utilizador, tipicamente, a sua máquina apenas tem de conseguir executar um *software* específico que será o responsável por coordenar e realizar a execução da tarefa pretendida. Normalmente, esse software é fornecido pela entidade que está a tirar proveito deste tipo de computação. Além disso, como uma dada entidade não "paga" por tirar proveito da capacidade das máquinas dos utilizadores, se estas estiverem interligadas através da Internet praticamente não existem custos de infraestrutura para este paradigma, nestes dois aspectos (Abah & Ogwueleka, 2013).

[21] http://setiathome.berkeley.edu

Segundo Ian Foster (Foster et al., 2008), os paradigmas *grid* e *cloud computing* partilham a mesma visão: reduzir custos de computação ao mesmo tempo que se pretende aumentar a qualidade, flexibilidade e fiabilidade à custa da delegação de serviços para uma terceira entidade. Porém, entre outras características e detalhes técnicos (Foster et al., 2008) (Buyya et al., 2009) (Sadashiv & Kumar, 2011), estes dois paradigmas apresentam diversas diferenças apesar da similaridade em algumas funcionalidades e possibilidades. Por exemplo, *grid computing* é um modelo essencialmente descentralizado no que diz respeito à localização e distribuição dos recursos envolvidos. Estes recursos e respetivas máquinas são ainda propriedade de diversos e diferentes utilizados, ao passo que na *cloud* a infraestrutura que a suporta é normalmente propriedade de uma única entidade: o provedor de serviços da nuvem. Apesar destes aspectos, muitas vezes a computação em *grid* acaba por ser considerada como sendo uma versão limitada da computação em *cloud* devido a alguns problemas que existem no primeiro paradigma, em que, por exemplo, a falha de um nodo é mais difícil de colmatar do que num ambiente em nuvem, no qual se recorre a técnicas como a redundância para resolver tais situações.

2.4 Os atores num ambiente em *cloud*

A *cloud* trouxe novas possibilidades para os seus utilizadores, mas também originou um novo segmento de mercado abrindo portas para novas oportunidades de negócio na indústria das TI. Muitas organizações montaram o seu negócio em torno da *cloud*, não apenas para usufruir dos seus serviços, mas também para oferecerem soluções assentes nesse ambiente. Como em qualquer serviço que seja disponibilizado, pode-se sempre identificar, no mínimo, dois atores fundamentais:

- O Utilizador ou Consumidor, também designado noutros contextos como cliente final, é aquele que utiliza diretamente as funcionalidades e recursos providenciados por uma dada *cloud*. É a entidade ou organização que utiliza os serviços de *cloud computing*, sejam estes relacionados com *software,* plataformas ou de infraestruturas.

- O Provedor, que é a entidade ou organização responsável por disponibilizar os serviços da *cloud* para os consumidores, quer seja através de uma interface própria (PaaS) ou através de máquinas virtuais (IaaS). Os serviços também podem ser disponibilizados na forma de SaaS, porém existe alguma ambiguidade sobre se estes provedores devem ser designados provedores de *clouds* ou provedores de serviços. Os provedores são também os

responsáveis por adquirir e gerir toda a infraestrutura necessária para suportar os serviços fornecidos.

Embora estes dois atores sejam os principais intervenientes num ambiente em *cloud*, estes não representam a totalidade do elenco. Claro que à medida que este conceito foi crescendo, diversas entidades procuraram entrar neste mundo oferecendo inúmeras atividades relacionadas com este processo de provisionamento entre consumidores e provedores. Desde entidades que realizam auditoria às implementações dos serviços na *cloud*, a intermediários e revendedores que acrescentam alguma funcionalidade a serviços já existentes ou mesmo organizações que providenciam ferramentas de suporte para a implementação/execução de serviços, todos estes desempenham um papel neste ambiente e têm a sua importância no processo de provisionamento. Talvez devido à sua diversidade, até hoje apenas surgiram propostas que caracterizassem estes intervenientes, não existindo por agora nenhum consenso assumido (Comissão Europeia, 2010) (NIST, 2011).

2.5 Características e benefícios

É simples de perceber porque é que a *cloud computing* tem tido bastante aceitação, procura e notoriedade no mundo das TI. A *cloud* é um ambiente elástico que permite a fácil libertação e aquisição de recursos considerados, teoricamente, infinitos. Além disto, são os provedores dos serviços de *cloud* que ficam responsáveis por tratar das questões e encargos relacionados com infraestrutura, desde aquisição, configuração ou manutenção. Isto possibilita aos utilizadores acederem a infraestruturas poderosas, altamente escaláveis e dispendiosas, tipicamente através dum modelo *"pay-as-you-go"*[22]. O facto desta infraestrutura ser suportada por terceiros, faz com que os seus utilizadores se concentrem apenas nos seus propósitos de utilização para este ambiente. No caso de empresas, a *cloud computing* permite que estas se foquem única e exclusivamente no seu processo de negócio e consigam rapidamente pôr o mesmo em prática.

[22] Modelo de negócio no qual o cliente paga à medida que vai utilizando um determinado serviço. Tipicamente, o cliente é cobrado consoante os recursos computacionais que utiliza e/ou o tempo durante o qual os utilizou.

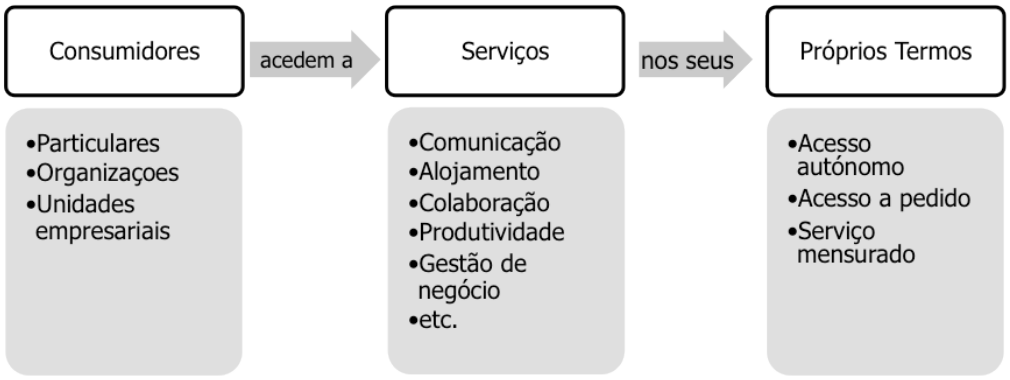


Figura 2.1 Perspectiva da *cloud* por parte dos consumidores[23]

Nos últimos anos tem-se assistido a um fenómeno crescente designado por *cloud washing*. Este é a tentativa, por vezes intencional, de um provedor associar ou (re)distribuir um antigo produto associado agora à *cloud* e ao conceito de *cloud computing*, apenas porque apresenta algumas das semelhanças e características de um serviço disponibilizado através de uma *cloud*. É assim fundamental conhecer as características essenciais que definem a *cloud computing*, de modo a evitar que este tipo de situações aconteçam. Na sua proposta para definição do conceito de *cloud computing*, o NIST indicou que este é composto por cinco características essenciais que devem estar presentes na implantação e fornecimento dos serviços disponibilizados. Essas características são:

- Acesso através da rede: os serviços devem ser fornecidos através de uma rede e acedidos através de mecanismos padrão, de modo a que praticamente qualquer dispositivo consiga usufruir dos mesmos. De realçar que a rede em causa não tem, necessariamente, que ser a Internet. Por exemplo, no caso de *clouds* privadas pode ser utilizada uma rede local sem acesso ao exterior.

- Acesso autónomo e a pedido**:** os consumidores podem utilizar os serviços sempre que necessitarem/entenderem e sem existir qualquer interação humana da parte do provedor do serviço.

[23] Imagem adaptada e extraída de (EMC Consulting, 2010)

- Agrupamento e partilha de recursos[24]: permite aos provedores agruparem diversos recursos, sejam físicos ou virtuais, e partilharem os mesmos por diferentes consumidores de acordo com as suas necessidades no momento. Por outra perspetiva, significa que os dados ou aplicações de diferentes consumidores estão alojados ou suportados pelos mesmos recursos, existindo assim a premissa de que o consumidor não tem conhecimento nem controlo da localização dos recursos fornecidos. Por vezes, e por questões legais, se o provedor possibilitar tais opções, o consumidor pode limitar a localização dos recursos com um certo nível de abstração, indicando por exemplo o país ou cidade onde estes devem residir.

- Rápida elasticidade: sendo um ambiente escalável, a *cloud* permite a aquisição e libertação de recursos de forma rápida, fácil e a pedido. Do ponto de vista do consumidor, os recursos disponíveis são infinitos e podem ser solicitados a qualquer altura e na quantidade que necessitar.

- Serviço avaliado e mensurado: a *cloud* controla e monitoriza os recursos utilizados por um dado serviço. De seguida, o consumidor é cobrado de acordo com a utilização dos recursos que teve. Os recursos cobrados são determinados de acordo com o tipo e natureza do serviço disponibilizado (ex. medição do espaço utilizado em disco, tempo de processamento, etc.).

Apesar de não estarem explicitamente evidenciadas acima, existem outras características que estão associadas com a *cloud computing* e que são típicas dos processos de fornecimento de serviços ditos tradicionais. Termos como fiabilidade, disponibilidade ou adaptabilidade estão subentendidos nas cincos características fundamentais identificadas pelo NIST.

Atendendo a todas as características acima apresentadas, e aliado ao facto dos serviços serem cobrados consoante os recursos gastos, a elasticidade deve ser a característica que mais é considerada por parte dos consumidores e aquela que mais contribui para que estes adoptem serviços de *cloud*. Tome-se como exemplo uma empresa que tem a sua plataforma *web* de venda de vestuário *online* assente numa *cloud*. Desde logo, a empresa tem a vantagem de não ter de se preocupar com qualquer infraestrutura envolvida nem com os custos inerentes à mesma. Se

[24] Adaptado do inglês: *resource pooling*

precisasse, a empresa teria de planear muito bem o que iria fazer. Teria de ter alguma forma de medir a afluência que iria existir ao *website*, bem como garantir uma infraestrutura com maior capacidade do que a necessária/calculada de modo a acolher alguma variação não esperada. Mesmo com esta situação resolvida, a empresa poderia novamente sofrer de questões de provisionamento, caso sucedessem picos inesperados de acesso à sua plataforma *web*. Até ao momento de ocorrência, seria sempre uma incerteza se a infraestrutura iria ou não estar à altura da exigência. Além disso, como a afluência ao *website* poderia não ser regular, existiriam sempre recursos inutilizados. Claro está que, aliado a estes problemas de provisionamento, estão sempre associados os custos de aquisição e gestão da infraestrutura que, tipicamente, crescem consoante as necessidades existentes. No caso de pequenas e até médias empresas, este tipo de situações de previsão de provisionamento podem ser incomportáveis devido aos custos envolvidos. Neste caso, como a plataforma é suportada por uma *cloud*, não existe a possibilidade de ocorrência de provisionamento em excesso ou em défice pois os recursos podem ser rapidamente ajustados a pedido e consoante a necessidade.

A *cloud computing* é assim um serviço que apresenta vantagens óbvias do ponto de vista do utilizador, seja este particular, uma empresa, organização, etc.. No entanto, tudo isto não seria possível sem o contributo dos provedores de serviços que ficam encarregues da gestão da infraestrutura necessária, dos respetivos encargos e, entre outros, da promoção de uma alegada disponibilidade infinita de recursos computacionais. Para isto, os provedores necessitam de garantir infraestrutura poderosa e em grande escala, de modo a que seja capaz de servir múltiplos consumidores e ainda contratar técnicos especializados em diferentes áreas, de modo a poderem operar com os diversos tipos de serviços disponibilizados e com os recursos associados. Não obstante, a *cloud* é uma realidade pois é também um negócio, e os provedores também têm benefícios com isso (Armbrust et al., 2009).

2.6 Modelos de serviço

Desde o momento em que a *cloud computing* começou a ser uma realidade, este tem sido apresentado como a próxima melhor solução para qualquer área ou tecnologia das TI. Anuncia-se que tudo pode ser fornecido como um serviço, de forma mais vantajosa do que as ofertas ditas tradicionais. Apesar disto, apenas três modelos são frequentemente mencionados: infraestrutura, plataforma e *software* como um serviço. De forma a simplificar e perceber os diversos tipos,

definiu-se um esquema em pirâmide ilustrando a forma como se posicionam esses três modelos típicos. A natural percepção de hierarquia foi o principal objetivo na escolha da pirâmide. Desta forma existe a noção de que cada camada é construída uma em cima da outra, e de que as camadas mais altas estão a um nível de abstração maior que as camadas inferiores. Esta abstração diz respeito à gestão e manutenção de recursos, sendo que as camadas superiores delegam maior responsabilidade para o provedor dos serviços do que para o consumidor. Apesar das camadas superiores estarem diretamente relacionadas com as inferiores, não existe necessariamente uma relação de interdependência. Cada camada pode, assim, existir por si própria.

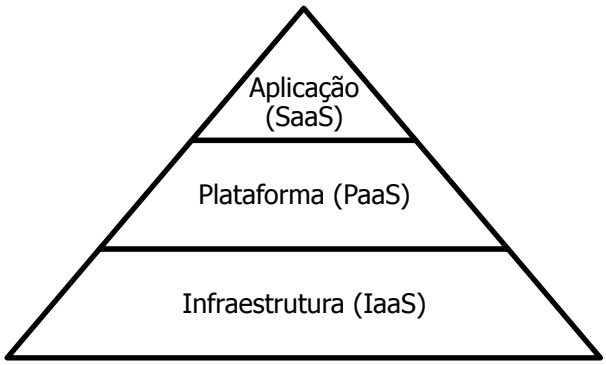


Figura 2.2 Pirâmide de modelos de serviço na *cloud*[25]

Um modelo de serviço é definido, nada mais nada menos, por aquilo que é fornecido a um dado consumidor. Basicamente, significa que num hipotético modelo XaaS o recurso designado por X é alojado, gerido e fornecido pelo provedor. Como já foi dito, os principais tipos de modelos de serviço são IaaS, PaaS e SaaS. Porém, do ponto de vista desta dissertação faz sentido falar também de modelos que possibilitem a gestão de dados, por exemplo através da disponibilização de bases de dados ou mesmo SDWs como um serviço. Estes termos aparecerão mais a frente nesta dissertação, para que, nesta secção, o foco se mantenha apenas nos três principais tipos já indicados.

2.6.1 *Infraestructure as a Service* (IaaS)

IaaS é o primeiro nível da pirâmide e aquele que permite maior liberdade aos consumidores do ponto de vista de utilização dos recursos. Neste modelo, o provedor disponibiliza capacidade de

[25] Imagem extraída e adaptada de (Abah & Ogwueleka, 2013)

processamento, armazenamento, memória, redes e outros recursos de computação primordiais. Tipicamente, estes recursos são disponibilizados através de máquinas virtuais instanciadas pelo provedor, no qual o consumidor pode de seguida instalar sistemas operativos ou quaisquer outros aplicativos. Nesta forma, o consumidor não controla diretamente a infraestrutura que suporta e fornece os recursos disponibilizados, mas tem total controlo de gestão de tudo o que implementar e montar com os mesmos. Comparativamente a uma solução tradicional, ao optar por IaaS o consumidor beneficia do acesso a infraestrutura avançada e disponibilidade infinita de recursos. O Google Compute Engine[26], Amazon EC2 e Windows Azure (*virtual machines*) são alguns exemplos de IaaS.

2.6.2 *Platform as a Service* (PaaS)

Este modelo caracteriza-se por disponibilizar um ambiente de desenvolvimento para o consumidor no qual este pode implementar e alojar as suas aplicações. Este ambiente é também designado por plataforma de computação, que incluí um sistema operativo e todos os componentes necessários para o desenvolvimento de uma dada aplicação. Por exemplo, se o objetivo fosse desenvolver um *website*, a plataforma poderia incluir, por exemplo, além do sistema operativo, um servidor *web*, uma base de dados, uma linguagem de programação e as respetivas bibliotecas, etc. Toda a manutenção da plataforma e dos seus componentes nativos fica a cargo do provedor. Por outro lado, o consumidor tem controlo total das aplicações que implementa sobre a plataforma. Em certas soluções de PaaS, os consumidores podem aceder e gerir as definições dos componentes do ambiente de desenvolvimento, podendo ainda usufruir do escalonamento automático de recursos consoante as necessidades das aplicações implementadas. A ideia principal por trás deste modelo, é possibilitar aos consumidores um desenvolvimento simplificado e rápido de aplicações, libertando-os da complexidade de instalação e configuração de infraestrutura e de ambientes de programação. Como exemplos de soluções que se baseiam em PaaS, existem o Google App Engine[27], o Force.com[28] ou o Windows Azure.

[26] https://cloud.google.com/products/compute-engine
[27] https://cloud.google.com/products/
[28] http://www.force.com/

2.6.3 *Software as a Service* (SaaS)

No topo da pirâmide está, talvez, o modelo de serviço mais popular de *cloud computing* - SaaS -, muito provavelmente por culpa da explosão de soluções que se verificou desde que este conceito passou a ser uma realidade. De uma maneira simples, caracteriza-se pela possibilidade do consumidor aceder e utilizar *software* de um provedor. Isto permite ao consumidor libertar-se dos custos de aquisição do *software* ou de licenças para o mesmo, pagando o seu uso através de subscrições, por exemplo, baseadas no número de utilizadores do *software*. Toda a instalação, manutenção e gestão do software é da responsabilidade do provedor, o que incluí quaisquer requisitos ou necessidades que a aplicação tenha ou venha a precisar. A única configuração que o consumidor tem acesso é aquela que é disponibilizada pela aplicação, que tipicamente são definições ao nível da aplicação. Os consumidores deste tipo de soluções beneficiaram ainda da crescente utilização de tecnologias *Web* para distribuição destes serviços de software, o que fez com que estas aplicações possam estar acessíveis através de um browser e, consequentemente, em diferentes dispositivos. Tal como já foi referido, existem diversos exemplos de soluções SaaS. O Google Apps[29] é um caso, sendo que se trata de um conjunto de aplicativos alojados na *cloud*, como o Gmail[30], Google Docs ou Google Drive[31]. Outros exemplos são o caso do CRM[32] da Salesforce ou o Microsoft Office 365[33].

2.6.4 Soluções tradicionais e serviços de *cloud*: uma breve comparação

A *cloud computing*, além de poder ser caracterizada por todas as funcionalidades e vantagens que providencia, pode também ser vista como uma separação das responsabilidades entre os típicos provedores e consumidores. Esquecendo a *cloud*, e todos os seus serviços e soluções, de um ponto de vista tradicional, as empresas tinham que adquirir o *hardware* e *software* necessário para o seu negócio e lidar com todas as questões a si inerentes, desde manutenção, segurança, atualizações etc. Esta abordagem é frequentemente designada na literatura por *On-Premises* ou *In-House*. Ao longo deste documento já se apresentaram diversas vantagens que a *cloud*

[29] http://www.google.com/enterprise/apps/business/
[30] https://mail.google.com/
[31] https://drive.google.com
[32] Sigla para *Customer Relationship Management*
[33] http://office.microsoft.com/

proporciona comparativamente às ditas soluções tradicionais. A nível de responsabilidades, a Figura 2.3 permite dar-nos uma ideia bem mais clara da diferença entre uma abordagem tradicional e os três modelos de serviço já apresentados.

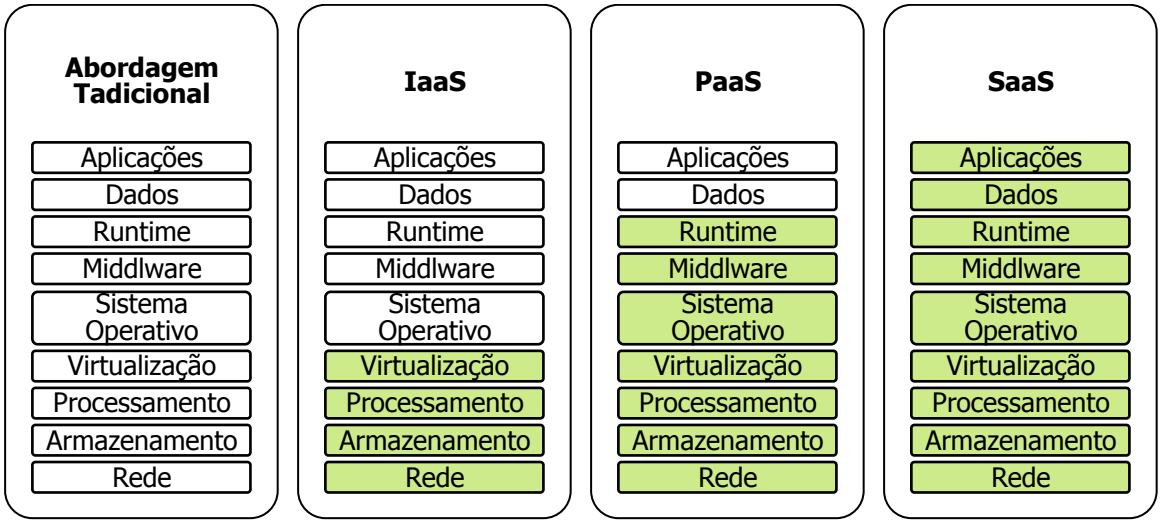


Figura 2.3 Comparação da divisão de responsabilidades entre uma abordagem tradicional e os modelos de serviços da *cloud*[34]

De uma forma simplista, a Figura 2.3 apresenta uma estrutura típica de um sistema que as empresas teriam de adquirir numa abordagem tradicional. Essa estrutura é também baseada nos recursos que um consumidor pode usufruir com serviços de *cloud*. Na figura assume-se que o sistema seria composto pelos nove níveis acima apresentados, que cada nível com fundo branco é da responsabilidade do consumidor e que os restantes, que estão preenchidos a cor, são da responsabilidade do provedor do serviço. Assim como foi dito aquando da apresentação da pirâmide de serviços de *cloud*, pode-se observar que, à medida que se vai subindo na hierarquia da mesma, a responsabilidade de gestão e da manutenção de recursos vai aumentando para o provedor e diminuindo para o consumidor. O consumidor abstrai-se cada vez mais dos recursos disponibilizados à medida que a sua escolha recai sobre os níveis superiores da pirâmide.

[34] Imagem recolhida e adaptada de (Comparison of the conventional software-system management model with the cloud services, 2012?)

2.7 Modelos de implantação

Os modelos de implantação são os diferentes meios através dos quais os consumidores acedem aos serviços de *cloud*. Estes modelos, também designados comummente por *clouds*, diferenciam-se e caracterizam-se, essencialmente, pelo público alvo que servem. Se se atender a este pressuposto, a definição de um modelo público e outro privado contempla todas as alternativas possíveis. No entanto, de acordo com a recomendação do NIST, existem quatro tipos de *clouds*: privada, pública, comunitária e híbrida. No que diz respeito à definição e caracterização das *clouds comunitária* e híbrida, a proposta do NIST foi considerada confusa, inconsistente e por vezes ambígua (Chou, 2011). Do ponto de vista desta dissertação, aceitou-se a *cloud* privada e pública como sendo os dois modelos essenciais de implantação. Quanto aos dois restantes, mais à frente, na secção 2.7.3 é apresentada a sua descrição e justificação do porquê de não serem considerados como modelos essenciais de implantação.

2.7.1 *Cloud* Privada

As *clouds* privadas são tipicamente utilizadas na distribuição dos serviços para uso exclusivo num meio fechado ou interno, por exemplo para os utilizadores de uma dada organização. Na literatura, por vezes é transmitida a ideia errada de que toda a infraestrutura tem de ser adquirida e gerida pelos próprios utilizadores da *cloud* privada. Na realidade, a infraestrutura que suporta a *cloud* privada pode ser delegada para um provedor de serviços e o seu acesso continuar a ser exclusivamente privado. Quando é gerida internamente pelos próprios utilizadores, estes têm total controlo dos processos, dados ou aplicações utilizadas na *cloud*, mas podem perder ou ver diminuídos alguns dos benefícios gerais de *cloud computing*, como o acesso a infraestrutura a preços reduzidos, a elevada elasticidade e disponibilidade de recursos, tempos rápidos de implantação, etc.

2.7.2 *Cloud* Pública

Contrariamente às *clouds* privadas, este modelo é utilizado para a distribuição dos serviços para o público em geral, tipicamente através da Internet. Toda a infraestrutura de suporte fica ao encargo do provedor do serviço, sendo inteiramente partilhada pelos diversos utilizadores desta *cloud*. Graças a esta partilha e com a optimização dos processos de gestão de recursos, os provedores conseguem maximizar a utilização dos mesmos e possibilitar o seu fornecimento a preços

reduzidos para os consumidores. A separação dos recursos por utilizadores é meramente lógica e é feita por exemplo através do uso de credenciais de acesso.

2.7.3 Outros modelos presentes na proposta do NIST

O NIST alega que uma *cloud* comunitária tem como público alvo um conjunto específico de utilizadores de várias organizações que têm algum tipo de interesses em comum. Menciona ainda que a infraestrutura que suporta essa *cloud* pode pertencer (e ser gerida por) a uma ou mais das organizações envolvidas, por um provedor de serviços externo à comunidade ou por qualquer combinação destes. Atendendo a esta definição e independentemente do número de organizações ou utilizadores envolvidos, este modelo continua a ser na mesma uma *cloud* privada e por isso não deve ser distinguido como sendo um tipo diferente. A *cloud* comunitária poderia ser identificada como uma generalização da *cloud* privada, mas mesmo essa distinção parece ser pouco relevante.

Relativamente à *cloud* híbrida, o NIST descreve-a como sendo uma composição de diferentes tipos de *clouds* mencionados (privada, pública e comunitária). As diversas *clouds* são interligadas através de meios standard ou proprietários de forma a permitir a portabilidade de dados ou aplicações entre as mesmas. Acima de tudo, o propósito de classificação – o público alvo – deveria ser consistente entre os diferentes tipos de *clouds* identificados. Independentemente de qual seja a combinação que dá origem a uma *cloud* híbrida, esta pode ter apenas um dos dois públicos alvo diferentes: um privado ou um público. Por este motivo, torna-se também confuso a definição de um novo tipo de *cloud*. Yung Chou (Chou, 2011) refere que este termo de *cloud* híbrida pode ter sido originado por força do meio empresarial. Para ele, uma *cloud* híbrida é meramente uma extensão de uma *cloud* privada ou pública que beneficia da introdução/cooperação com outros recursos tecnológicos que sejam necessários a uma dada organização.

2.7.4 Comparação de *clouds* privadas e públicas

Após a revisão dos diversos tipos de *clouds*, foi possível esclarecer que, na prática, apenas existem dois modelos essenciais: privado e público. Além disto, foi ainda dito que, ao contrário do que é frequentemente mencionado na literatura, é possível identificar duas abordagens no que diz respeito ao alojamento de *clouds* privadas. Uma *cloud* é privada consoante o seu público alvo, e não por questões de propriedade, localização ou responsabilidade de gestão. Esta falsa premissa de que a infraestrutura de uma *cloud* privada é gerida pelos próprios utilizadores pode muito bem

ter surgido do facto de muitas empresas terem desenvolvido as suas próprias *clouds* internamente, aproveitando a infraestrutura existente e respetivos investimentos feitos anteriormente. Devido a este equívoco, para fazer a comparação dos dois tipos de *clouds* far-se-á a distinção entre *clouds* privadas geridas internamente pelos seus próprios utilizadores e *clouds* privadas geridas externamente por um provedor de serviços.

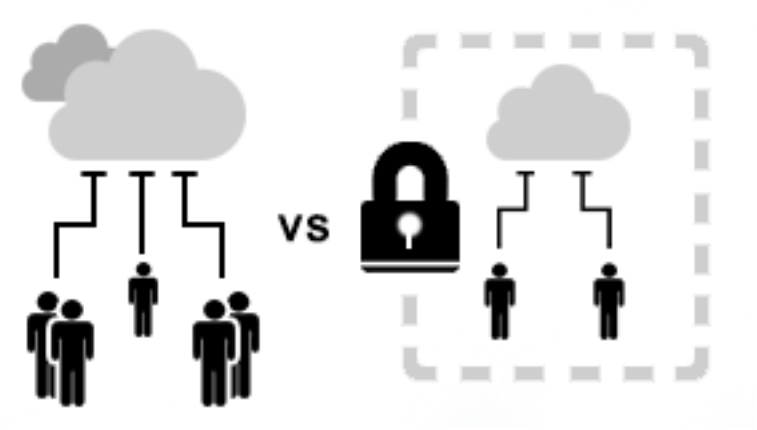


Figura 2.4 Comparação entre *clouds* privadas e públicas - modelos essenciais de implantação

Quando comparadas, uma *cloud* pública e uma *cloud* privada apresentam algumas diferenças, sempre que esta última for gerida internamente, ou seja, pelos seus próprios utilizadores (Oracle, 2010). Além do público alvo, a diferença de proprietário no que diz respeito à gestão de toda a infraestrutura da *cloud* é o motivo principal para a existência de diversas vantagens e desvantagens entre estes dois modelos de implantação. Como é gerida internamente, os utilizadores de uma *cloud* privada têm total controlo sobre tudo o que esta contém. Desta forma, este tipo de *clouds* privadas proporciona:

- maior controlo, segurança e privacidade;
- possibilidade de conformidade com restrições governamentais, por exemplo relativas à localização e residência dos dados;
- aumento da qualidade de serviço, por exemplo graças à utilização de redes locais que permite a diminuição de perdas de desempenho nas comunicações;
- maior facilidade de integração de aplicações tradicionais com aplicações executadas na *cloud*.

Estas *clouds* privadas caracterizam-se ainda por a longo prazo terem um custo total reduzido relativamente às *clouds* públicas. Apesar de terem um enorme investimento inicial, a infraestrutura adquirida é propriedade do utilizador e chegará a um ponto em que todo o seu custo será compensado, ao passo que com o aluguer dos serviços de uma *cloud* pública, o utilizador paga um montante mais baixo, mas mensal, o que a longo prazo se pode traduzir num custo acumulado superior ao custo inicial realizado numa *cloud* privada. As *clouds* públicas são ainda consideradas ambientes *"one size fits all",* ou seja, os utilizadores perdem alguma personalização pois as soluções são fornecidas de modo a abrangerem o maior número de potenciais consumidores. Apesar disto estas *clouds* apresentam vantagens:

- inexistência de aquisição, configuração e manutenção de infraestrutura o que proporciona aos utilizadores começarem rapidamente a usufruir dos serviços;
- custos iniciais reduzidos, pois os utilizadores são cobrados apenas consoante a utilização que fazem dos recursos disponibilizados;
- acesso a infraestrutura tecnologicamente avançada e a recursos teoricamente infinitos, pois, como o seu propósito é servir diversos utilizadores, à partida um provedor terá melhores condições a oferecer do que uma organização que opta por montar uma infraestrutura para uso pessoal;
- acesso aos serviços a partir de qualquer lugar, pois tipicamente estes são distribuídos através de uma rede pública, a Internet.

Relativamente à segunda comparação, esta é sobre *clouds* públicas e *clouds* privadas geridas externamente. No geral, estes dois tipos apresentam praticamente as mesmas características, sendo assim muito semelhantes. A principal diferença anunciada é o agrupamento e partilha de recursos entre diversas organizações que é feito no caso das *clouds* públicas, uma vez que nas privadas os recursos são dedicados a uma única organização. Como não há partilha, o conjunto de recursos disponíveis nas *clouds* privadas poderá ser menor. Por outro lado, perante tal exclusividade, a *cloud* privada posiciona-se como sendo um ambiente que disponibiliza maior segurança e privacidade. Não devemos esquecer, porém, que as questões de segurança e de privacidade continuam a ser tratadas pelo provedor, o que pode também causar um aumento nos custos duma solução privada relativamente à mesma solução pública. De resto, em ambos os casos os consumidores beneficiam do facto das *clouds* estarem sob a alçada de um provedor,

aproveitando assim as diversas vantagens da *cloud computing* equitativamente. Na prática, a escolha de um ambiente é definida pela necessidade de ter maior controlo sobre o mesmo, quer seja por questões regulamentares, quer por requisitos específicos ou exigências existentes na indústria.

Capítulo 3

Data Warehousing na *Cloud*

A *cloud* tem sido proposta como a melhor solução para a implementação de serviços de informática. Devido às vantagens referidas no capítulo anterior, muitas organizações têm optado por implantar ou migrar os seus sistemas, processos e dados, de forma a usufruírem dos benefícios da elasticidade que este tipo de ambiente propiciona. No contexto de BI, os SDWs não são uma exceção e muito se tem falado na emergência, vantagens e novas possibilidades que adviriam graças à sua implementação na *cloud*. De facto, devido às vantagens da *cloud,* este parece ser um ambiente ideal para DWs. Mais uma vez, o "velho" requisito relativo a infraestruturas altamente escaláveis e poderosas volta a ser indispensável para uma implementação bem sucedida. Aliás, este requisito é muitas vezes o factor decisivo que leva as empresas a direcionar as suas atenções para a *cloud*, pois esta permite a obtenção dos recursos usualmente necessários a preços reduzidos. Não obstante, a *cloud* apresenta também algumas desvantagens que fazem levantar algumas preocupações para seus os consumidores. À parte das questões técnicas, as empresas têm primeiro de encontrar resposta para questões como (Salazar & Jiming, 2012): "É razoável e seguro confiar os dados para um provedor de serviços externo?", "Que garantias existem de que o provedor não desaparece um dia? O que aconteceria aos dados?", "Se se mudasse o provedor, os dados seriam perdidos?", "Existe algum risco de perder os dados armazenados na *cloud*?" ou "A transferência dos dados para a *cloud* é segura?". O que é certo, é que a implementação de SDWs na *cloud* parece obrigatória e inevitável perante as vantagens que isso traria. Inclusive, em 2012 foram dados passos importantes nesse sentido, pois

assistiu-se ao lançamento de soluções sonantes como o Redshift, lançado pela Amazon, e o serviço de *Data Warehousing* na *cloud* lançado pela Treasure Data.

3.1 Noções gerais sobre *Data Warehouses*

De uma maneira geral, assentes sob as típicas bases de dados relacionais, existem duas vertentes principais utilizadas para o armazenamento e gestão de dados: os sistemas de gestão de dados transacionais e os de gestão de dados analíticos. Os primeiros caracterizam-se por serem utilizados para registar a informação resultante das transações ocorridas nos diversos processos de negócio existentes, por exemplo, dentro de uma dada organização ou empresa. Como exemplo de aplicações que lidam com dados transacionais, existem as utilizadas para o registo de vendas, para a marcação de reservas em companhias aéreas ou hotéis, etc.. Este tipo de sistemas registam maioritariamente operações de escrita[35] e dependem da aplicação e verificação das propriedades ACID sobre as bases de dados que utilizam, de modo a garantirem a consistência e integridade dos dados (Abadi, 2009). As bases de dados destes sistemas revelam assim taxas de crescimento acentuadas, tendo mesmo tendência a alcançar estados de saturação que, face a interrogações mais complexas, reduzem a capacidade de resposta do sistema e aumentam os períodos de espera dos utilizadores habituais. Dada a necessidade em apresentarem tempos de resposta reduzidos para que o normal funcionamento dos processos de negócio não seja afetado, estes sistemas não são assim apropriados para a análise e exploração de dados, nos quais muitas vezes são realizadas interrogações que envolvem cálculos complexos e agregações de dados. É com este objetivo que surgem os sistemas de gestão de dados analíticos. Como são essencialmente sistemas de leitura[36] e albergam grandes volumes de dados, seja informação histórica ou atual, estes são essencialmente utilizados para o planeamento de novas estratégias de negócio, resolução de problemas e para o suporte à decisão (Abadi, 2009) - os DWs são exemplos de sistemas que permitem a gestão de dados analíticos.

Características	**Gestão de dados transacionais**	**Gestão de dados analíticos**
Objetivo	Suportar habituais processos de negócio	Resolução de problemas, tomada de decisão e planeamento de estratégias de negócio

[35] Na literatura é utilizada a expressão *write-intensive* ou *write-only* para descrever estes sistemas
[36] Na literatura é utilizada a expressão *read-only* para descrever estes sistemas

Volume de dados	Na ordem dos Megabytes/Gigabytes	Na ordem dos Gigabytes/Terabytes
Precisão	Dados atuais	Dados atuais e de histórico
Consultas	Simples, afetam poucos resultados e são à partida conhecidas e repetitivas pois suportam processos habituais de negócio	Normalmente são complexas e não são pré-estabelecidas (exploração ad hoc)
Desenho dos dados	Esquemas normalizados, de forma a garantir a consistência e integridade dos dados	Uso de esquemas desnormalizados com o objetivo de diminuir o tempo de satisfação das consultas
Operações	Essencialmente de escrita	Essencialmente de leitura
Exemplos	Sistemas de registo de vendas ou de reservas.	Data Warehouses

Tabela 3.1 Resumo de diferenças existentes entre sistemas de gestão de dados transacionais e de gestão de dados analíticos

Face ao crescente aumento de informação resultante de sistemas transacionais e à necessidade de análise desses mesmos dados, as empresas têm optado pela implementação de SDWs de modo a utilizar a sua informação para a tomada de decisão. Um DW é um conjunto de dados orientado para um dado assunto, integrado, variante ao longo do tempo e não volátil, especialmente orientado para o suporte de processos de tomada de decisão (Inmon, 2005). São sistemas que necessitam de um elevado número de recursos para poderem gerir os volumes de dados envolvidos e fazer a sua consequente disponibilização de forma expedita para os seus utilizadores. Estes utilizadores são vulgarmente designados por agentes de decisão e são os responsáveis por interrogar o sistema e explorar os dados contidos no mesmo.

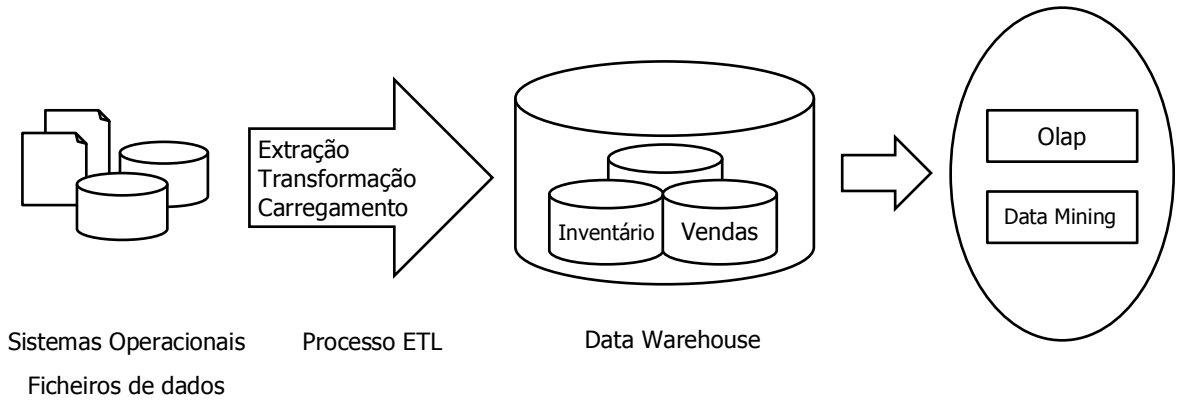


Figura 3.1 Arquitetura geral de um sistema de *Data Warehousing*

Normalmente, toda a informação contida num SDW é proveniente de um ou mais sistemas transacionais, comummente designados na literatura por sistemas operacionais (SO). Estes são considerados e utilizados como sendo as fontes de dados do sistema, se bem que outros tipos de fontes podem ser utilizadas, como é o caso de ficheiros de dados Excel, XML[37], entre outros. É nestas fontes que se encontram os dados relativos às transações realizadas nos diversos processos de negócio empresariais. De forma a levar esta informação para o DW, é iniciado um processo de ETL[38] que está encarregue de extrair os dados das fontes, transformá-los de forma a adaptarem-se ao DW e carregá-los posteriormente no mesmo. Como muitas das vezes a introdução dos dados nas fontes não é cuidada, de forma a garantir a qualidade dos dados no SDW, é necessário tratar e resolver algumas anomalias e conflitos existentes, quando se unifica informação proveniente de diversas fontes, como é o caso de inconsistências, duplicados, valores nulos, etc., havendo assim a necessidade de realizar este processo de transformação e limpeza de dados (Rahm & Do, 2000). Como se trata do processamento e carregamento de grandes volumes de dados, o processo ETL é geralmente exigente e moroso, sendo que o principal problema surge, precisamente, na última fase quando se dá a interação com o DW. Dependendo do contexto, pode ser obrigatório que o DW tenha que estar disponível 24/7, sendo assim um desafio carregar os dados e manter ao mesmo tempo o desempenho no sistema na satisfação de interrogações. Noutros casos pode existir uma janela de oportunidade para se realizar o carregamento dos dados, por exemplo a uma determinada hora durante a noite ou aos fins de semana, sendo que nestes períodos o DW, salvo algumas exceções, não estará disponível para consulta por parte dos agentes de decisão. Como foi dito anteriormente, o DW é um sistema essencialmente de leitura, sendo que os únicos momentos de escrita e atualização dos dados ocorrem precisamente durante a execução dos processos ETL. Assim que os dados estiverem carregados no DW, estes estão imediatamente disponíveis para os agentes de decisão e para serem usados por diversas aplicações, como por exemplo ferramentas OLAP[39] ou de Data Mining. De forma semelhante aos SOs, no DW os dados também podem ser armazenados nas típicas bases de dados relacionais, obtendo-se assim a escalabilidade já reconhecida destas bases e facilitando os processos de integração dos dados através da utilização de técnicas SQL convencionais. Porém, dada a finalidade do DW, os esquemas de dados são

[37] XML - Extensible Markup Language
[38] ETL (*Extract*, *Transform and Load*) – Extração, Transformação e Carregamento
[39] OLAP - On-line Analytical Processing

desenhados de forma orientada à decisão, a um dado assunto e não orientados ao processo como no caso dos SO.

Apesar da utilidade que tem e das possibilidades que permite, as empresas têm de realizar um esforço financeiro considerável de modo a concretizarem a implementação de um SDW. O seu processo de construção é moroso e muito dispendioso, envolvendo custos iniciais e recorrentes resultantes da gestão do sistema (Inmon, 2000). Este processo passa, tipicamente, por um conjunto de fases bastante diverso, que inclui o planeamento e justificação do projeto, levantamento de requisitos, desenho dos dados, análise das fontes de dados, planeamento e configuração da infraestrutura, definição de mecanismos de segurança, instalação, implementação, análises de desempenho, testes, etc. Mesmo após a sua implementação, o DW requer constante monitorização e atualização dos seus dados, obrigando assim a contínuos esforços de gestão e manutenção não só do próprio sistemas mas também da sua infraestrutura associada. Todas estas fases são parte de uma metodologia conhecida e aceite na área que tem como objetivo impulsionar a construção de um DW para o sucesso (Kimball et al., 1998). É assim um processo bastante complexo e que exige a disponibilização de diversos recursos, sejam estes *hardware*, *software* ou mesmo humanos. Além disto, a implementação deste tipo de sistemas revela-se um projeto arriscado pois não existe qualquer garantias de retorno do massivo investimento realizado. Por outro lado, quando bem construídos, estes sistemas apresentam elevado desempenho na satisfação de interrogações sem afectar o funcionamento normal de cada uma das fontes de dados envolvidas, e contêm ainda informação que as mesmas nem sempre possuem, como é o caso de dados de histórico ou agregados.

3.2 A promessa da *cloud* para *Data Warehousing*

Hoje em dia, para muitas empresas os dados representam um dos recursos mais valiosos que estas possuem. Através de processos e tecnologias de BI, estes dados podem ser "transformados" em conhecimento útil imprescindível para as empresas melhorarem os seus negócios. Assim, quer seja localmente de uma maneira tradicional (*on-premises*) ou na *cloud*, a área de BI é uma das áreas mais interessantes de *software*. Mesmo quando a economia não é favorável e é necessário cortar custos e diminuir as verbas disponíveis para as TI, as organizações continuam a melhorar e investir nos seus processos de BI, pois, como estes ajudam a prever tendências de mercado e auxiliam na definição de novas estratégias de negócio para o mesmo, permitem rapidamente a

recuperação do investimento realizado. Com o aparecimento da *cloud computing* e a forte adoção ao paradigma, é quase obrigatório e natural que se comece também a tentar migrar as diversas aplicações e sistemas de BI para a *cloud*. No caso específico dos DWs, essencialmente é necessário perceber se este é o ambiente certo para estes sistemas e se as organizações estão dispostas a adotar esta mudança.

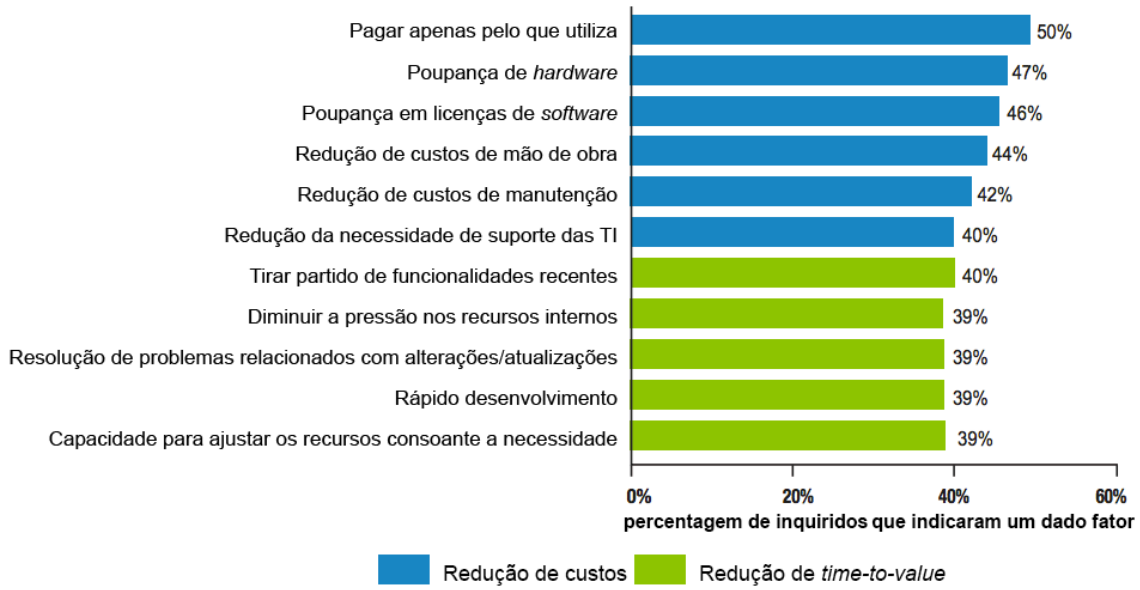


Figura 3.2 Razões indicadas que levam à migração para um ambiente em *cloud*[40]

Os benefícios proporcionados e as vantagens associadas à *cloud* são já conhecidas e comprovadas na indústria. De facto, através de um estudo realizado pela IBM onde foram inquiridos diversos agentes de decisão das diversas áreas das TI (IBM, 2010), foi possível identificar alguns motivos que estes consideram como sendo chave para a migração e adopção de soluções em *cloud*. A Figura 3.2 apresenta alguns dos motivos recolhidos desse estudo, sendo que mesmo atendendo a todos os resultados, é possível perceber-se que os benefícios mais considerados e apelativos se enquadram na redução de custos e na redução de *time-to-value*[41]. Apesar do estudo realizado ser sobre a adoção do conceito de *cloud computing* em geral, estes dois factores também se aplicam e são decisivos no contexto dos DWs. Como se sabe, um DW assume-se como um sistema muito

[40] Imagem recolhida e adaptada de (IBM, 2010)
[41] Expressão utilizada na literatura para designar o período de tempo entre o pedido de um valor específico e o momento de entrega desse mesmo valor.

dispendioso quer seja no início ou no final do seu processo de construção, sendo também muito exigente durante o seu funcionamento. Assim, com a implementação de SDWs na *cloud*, é possível diminuir-se os custos de implementação e manutenção deste tipo de sistemas, acelerar o seu processo de desenvolvimento e consequentemente reduzir tempos de implementação, e ainda obter-se à medida todos os recursos necessários para o seu funcionamento (Madsen, 2012).

Numa instalação *on-premises*, uma organização tem de planear e adquirir a infraestrutura necessária para a construção e utilização do DW, considerando e prevenindo-se sempre para possíveis picos de utilização e para o seu crescimento futuro. Não só serão contabilizados custos de infraestrutura, mas também de mão de obra especializada necessária para instalar e manter todo o sistema. Mesmo prevenindo-se para o seu crescimento, haverá seguramente uma altura em que será necessário aumentar ou atualizar a infraestrutura existente, obrigando à aquisição de novos componentes, como é o caso de servidores poderosos e dispendiosos. Através do agrupamento e partilha de recursos, um provedor de serviços na *cloud* consegue aumentar o rendimento e utilização desses mesmos recursos, possibilitando a sua disponibilização a baixos preços. Com preços reduzidos e segundo um modelo *pay-as-you-go*, a *cloud* retira assim toda a necessidade de aquisição de infraestrutura bem como os elevados custos iniciais envolvidos. Dada a elasticidade deste ambiente, uma organização não tem ainda que se preocupar com o provisionamento em défice ou em excesso, nem com a subutilização dos recursos adquiridos. Qualquer necessidade existente será satisfeita com a aquisição e libertação de recursos à medida, não sendo necessário proceder à atualização de infraestrutura por parte da organização. Utilizando a *cloud* para obter infraestrutura para a implantação de bases de dados, consegue-se assim remover e ultrapassar muitas das barreiras existentes que atrasam o desenvolvimento de projetos. Este aspeto ganha particularmente importância nas grandes organizações em que é necessário obter-se aprovação de orçamentos, sendo que, muitas vezes, a espera pela mesma pode atrasar ou inviabilizar a execução de um dado projeto (Madsen, 2012). Com o acesso autónomo, simplificado e a pedido a recursos computacionalmente poderosos é possível acelerar o processo de construção de um DW, pois toda a infraestrutura adquirida está imediatamente pronta a ser utilizada. É igualmente fácil e rápido adquirir recursos para aumentar a capacidade ou o desempenho do sistema. Aliado a este facto, e sabendo que se é cobrado apenas consoante os recursos utilizados, é mais fácil aprovar-se e obter-se as verbas necessárias para satisfazer o sistema, pois os habituais e elevados custos de infraestrutura são substituídos por custos menores e operacionais.

A *cloud* promete ainda bom desempenho, escalabilidade e agilidade para os DWs (Madsen, 2012). Numa abordagem *on-premises*, pode ser um desafio garantir o desempenho e a escalabilidade do sistema, seja na sua construção inicial com o planeamento e definição da infraestrutura necessária, ou durante a sua execução de modo a que este consiga gerir e disponibilizar enormes volumes de dados atempadamente. Segundo Mark Madsen, estas garantias que muitas vezes eram alcançadas com bastante esforço através da gestão das cargas de trabalho[42] existentes ou com a afinação[43] dos sistemas, podem agora ser alcançadas com a definição de níveis de serviço[44] entre o provedor da *cloud* e a organização, possibilitando assim que o sistema se adapte e adquira os recursos necessários mesmo durante o seu funcionamento. Estas garantias só são possíveis graças à rapidez existente na disponibilização a pedido dos recursos necessários. De facto, características como a elasticidade, custos reduzidos e rapidez de implementação possibilitada pela *cloud* trazem maior agilidade a esta área dos SDWs. Por vezes, muitos projetos não são realizados pois têm elevados custos iniciais, prazos curtos de realização ou porque simplesmente o tempo no qual geram algum valor é muito limitado e curto. Está-se assim a falar de projetos resultantes de pedidos inesperados, pouco planeados ou cujo seu benefício é pouco claro e difícil de justificar, o que consequentemente torna também difícil a obtenção de aprovação para a sua realização. Com um ambiente em *cloud*, estes projetos podem ser rapidamente postos em prática e realizados a custos reduzidos e, mesmo que não se obtenha o retorno esperado, podem ser imediatamente encerrados, sendo que não existirão quaisquer despojos associados ao mesmo, quer seja *hardware* ou *software* e respectivas licenças, etc. Mark Madsen refere ainda que, graças à *cloud*, como o custo é estático, ou seja, é igual realizar uma dada operação numa máquina durante cem horas e realizar a mesma operação numa hora em cem máquinas, a avaliação e decisão de realizar um projeto poderá agora ser feita com base no seu valor e oportunidade e não consoante os custos associados.

Reconhecidas algumas das vantagens da *cloud*, é necessário ainda perceber até que ponto as organizações estarão dispostas em adoptar este tipo ambiente. Isto porque, mesmo havendo vantagens, no caso de uma possível migração de soluções existentes, a área de *data warehousing* é um pouco diferente da maior parte das áreas existentes nas TI visto que tudo acontece em maior escala. Essencialmente trata-se de uma questão de custos e do enorme investimento que

[42] Na literatura é utilizado o termo *workloads*
[43] Na literatura é utilizada a expressão *"system tuning"*
[44] Na literatura é utilizado o termo SLA – Service-Level Agreement

poderá já ter sido realizado. Para uma organização que já tenha as suas aplicações de DW a correr com sucesso, com enorme infraestrutura e investimentos realizados, poderá não ser tão urgente ou necessária a mudança para a *cloud*. No caso da adoção de serviços de *cloud* de um provedor externo, seria mesmo necessário perceber o que se iria fazer com toda a infraestrutura própria que tinha sido adquirida para o efeito até ao momento e que estava a suportar o DW. Mesmo que venha a acontecer, essa mudança poderá não ser suave e rápida pois tipicamente envolve a mudança de todo o sistema e em alguns casos de diversas aplicações existentes em torno do mesmo. Porém, isto não inviabiliza a adoção do conceito de *cloud computing*, pois pode ser possível que a organização aproveite a infraestrutura que tem para criar internamente a sua própria *cloud* privada e tirar assim proveito das diversas vantagens inerentes. Por outro lado, uma nova oportunidade surge sim para aquelas organizações que procuram fazer uma primeira implementação de um SDW desde raiz na *cloud*. Tome-se como exemplo as pequenas e médias empresas que não têm o tempo, a mão de obra especializada e o investimento necessário para levarem a cabo a implementação e gestão de todo o ambiente em torno de um SDW, nem mesmo para monitorizar ou manter os diversos serviços associados dentro das suas próprias instalações (Barsch, 2012). Estas organizações vêem a *cloud* como uma solução através da qual conseguem delegar as responsabilidades existentes relacionadas com os SDWs para uma entidade externa, de modo a que não tenham de se preocupar com questões de propriedade, administração ou suporte da totalidade ou parte do sistema. Com a *cloud*, as empresas deixam assim de se preocupar com a aquisição de recursos e com os contínuos investimentos nas TI, tendo assim acesso a infraestrutura poderosa e atual e a mão de obra especializada com base numa subscrição. No caso das pequenas e médias empresas, a *cloud* é ainda um ambiente muito apelativo e estratégico pois, conhecidas as vantagens dos SDWs e das aplicações de BI, estas já não estão apenas ao alcance das grandes empresas com capacidade e estrutura para executar este tipo de projetos.

3.3 Gestão de dados na *cloud*: modelos e soluções existentes

Face à necessidade de mudança para a *cloud* e às oportunidades daí resultantes, começaram a surgir, com bastante naturalidade, soluções que permitem a gestão de dados neste ambiente. Inclusive, como em torno da *cloud computing* existe a ideia de que tudo pode ser fornecido como um serviço, muitas destas soluções deram origem a novos termos que passaram a ser

referenciados como sendo novos modelos de serviço disponíveis através da *cloud*. Mesmo existindo na literatura pouca consensualidade na sua definição, DaaS e DBaaS são os mais relevantes do ponto de vista desta dissertação sendo que, recentemente, começou também a surgir o conceito de DWaaS. Além destes novos modelos, os modelos já conhecidos e consensuais, como é o caso dos três tipos previamente apresentados nesta dissertação, podem também ser utilizados para alojar sistemas ou disponibilizar aplicações que permitam a gestão de dados, aumentando assim o leque de alternativas existentes.

Sendo um dos novos modelos que apareceram, DaaS é uma estratégia de *cloud* utilizada para disponibilizar dados úteis e específicos de negócio de forma rápida, segura e a custos reduzidos. DaaS respeita o princípio de que os dados podem ser fornecidos aos utilizadores sempre que estes os requisitarem, independentemente da separação organizacional ou geográfica entre as empresas e os provedores de serviços (Janssen, n.d.). Essencialmente, este serviço é usado por empresas ou organizações para acederem a dados geridos por terceiros, porém pode também ser utilizado para que estas mesmas entidades façam o outsourcing dos seus próprios dados (Delphix, 2011). Um exemplo típico de utilização é o acesso a conjuntos de dados referentes a um negócio específico, contexto ou problema de forma a realizar estudos e visualizar estatísticas sobre os mesmos. O Google Public Data[45] e o Freebase[46] são exemplos de serviços que disponibilizam conjuntos de dados referentes a diversos assuntos. Num caso de outsourcing, este modelo permite que uma organização aceda aos seus dados atualizados, de forma rápida, conveniente e sem prejudicar o funcionamento dos seus SOs. Como os dados estão alojados na *cloud*, estão ainda acessíveis através de diversos sistemas e dispositivos. Além disto, esta solução caracteriza-se por possibilitar a recolha de dados em diferentes formatos, por tipicamente guardar os mesmos em bases de dados relacionais e por consequentemente também permitir a sua disponibilização para os utilizadores em diversos formatos conhecidos na indústria. Talvez uma das principais desvantagens deste modelo é de que, geralmente, os dados acedidos não estão disponíveis para *download*.

Como outra possibilidade, existem algumas aplicações que também permitem o armazenamento e gestão de dados na *cloud* apesar de serem disponibilizadas sob a forma de *software as a service*. O Dropbox[47] ou o Google Drive[48] são exemplos deste tipo de aplicações que são normalmente

[45] http://www.google.com/publicdata/directory
[46] http://www.freebase.com
[47] https://www.dropbox.com
[48] https://drive.google.com

utilizadas para o armazenamento de cópias de segurança e gestão básica de ficheiros, de forma semelhante a um tradicional sistema de ficheiros. De facto, atualmente estas aplicações estão bem integradas nos diversos sistemas operativos, inclusive para dispositivos móveis, o que torna muito fácil a sua utilização. Duas das mais valias deste tipo de serviço são a sincronização dos dados e o preço de utilização. Com uma simples ligação à Internet, os utilizadores conseguem ver os seus ficheiros sincronizados e acessíveis a partir de praticamente qualquer dispositivo que possuam. Além da partilha dos dados entre dispositivos, estas aplicações também simplificam a partilha de ficheiros entre diferentes utilizadores. Claro está que o modelo de negócio é definido consoante a quantidade de espaço disponibilizada para cada utilizador. Normalmente é disponibilizada uma quantidade inicial de forma gratuita que é suficiente para uma utilização "doméstica". Os utilizadores apenas são cobrados se pretenderem uma maior capacidade de armazenamento, sendo que o valor vai aumentando quanto maior for a quantidade de espaço requisitada.

Apesar das soluções descritas acima e do vasto número de exemplos das mesmas já existentes no mercado, o principal foco está na implantação e utilização de bases de dados através da *cloud*. Como as bases de dados relacionais têm sido a principal solução para o armazenamento de dados no mundo empresarial, um modelo como DBaaS revela-se mais necessário, urgente e num patamar acima das soluções apresentadas. Atualmente, demasiadas organizações dependem de bases de dados relacionais para as suas aplicações, sendo que o ecossistema em torno destas bases é extremamente rico e vasto. Como curiosidade, Navneet Joneva, um gestor de produto do Google Cloud SQL[49] que é uma solução do tipo DBaaS, afirma que um dos pedidos mais frequentes dos utilizadores era para que existisse uma solução que lhes permitisse desenvolver de forma fácil as tradicionais aplicações suportadas por bases de dados (Claburn, 2011). Com estas soluções, o objetivo é disponibilizar as funcionalidades de uma base de dados relacional pela *cloud*, de forma a que os utilizadores possam armazenar e aceder às suas bases de dados a qualquer momento e a partir de qualquer lugar, usufruindo assim da escalabilidade, disponibilidade e fiabilidade que este ambiente consegue oferecer. Para isto, atualmente existem duas abordagens distintas que permitem a implantação de bases de dados na *cloud* (Arora & Gupta, 2012):

- Através da utilização de máquinas virtuais instanciadas numa *cloud*, nas quais o utilizador pode instalar uma base de dados tradicional, como Oracle, MySQL, PostgreSQL, etc. De

[49] https://developers.google.com/cloud-sql/

forma a simplificar e acelerar este processo, algumas empresas disponibilizam imagens de máquinas virtuais com os seus sistemas de bases de dados já pré-configurados e optimizados, de forma a que o utilizador apenas tenha de importar a imagem para a *cloud*. Esta abordagem pode ser facilmente alcançada através da utilização de uma solução IaaS ou PaaS, nas quais o utilizador tem controlo sobre as aplicações e sistemas que instalar nas máquinas virtuais disponibilizadas.

- Seguindo um modelo do tipo DBaaS, no qual o utilizador acede a uma base de dados providenciada por um dado provedor. Amazon RDS (Amazon Relational Database Service), o Microsoft SQL Azure ou o Heroku Postgres[50] são exemplos de soluções existentes que disponibilizam as tradicionais bases de dados na forma DBaaS.

Ambas as abordagens permitem aos utilizadores beneficiar do acesso a infraestrutura mantida por um provedor e de serem cobrados consoante os recursos que utilizarem. No entanto, a abordagem DBaaS é a mais cómoda e interessante para o utilizador pois, para além da infraestrutura física, a instalação, configuração e manutenção do motor da base de dados é também da responsabilidade do provedor. Praticamente, o utilizador apenas tem que se preocupar com os dados, os respectivos esquemas e as instâncias de bases de dados que criar. Para isto, é habitual os provedores disponibilizarem uma interface *web* para o cliente poder configurar, aceder e visualizar estatísticas de funcionamento das bases de dados disponibilizadas. Inclusive, os formulários e as opções existentes nestas interfaces são bastante semelhantes às disponibilizadas nos tradicionais assistentes de configuração, de forma a transmitir alguma familiaridade a um utilizador que tenha pouca experiência com soluções na *cloud*.

[50] https://postgres.heroku.com/

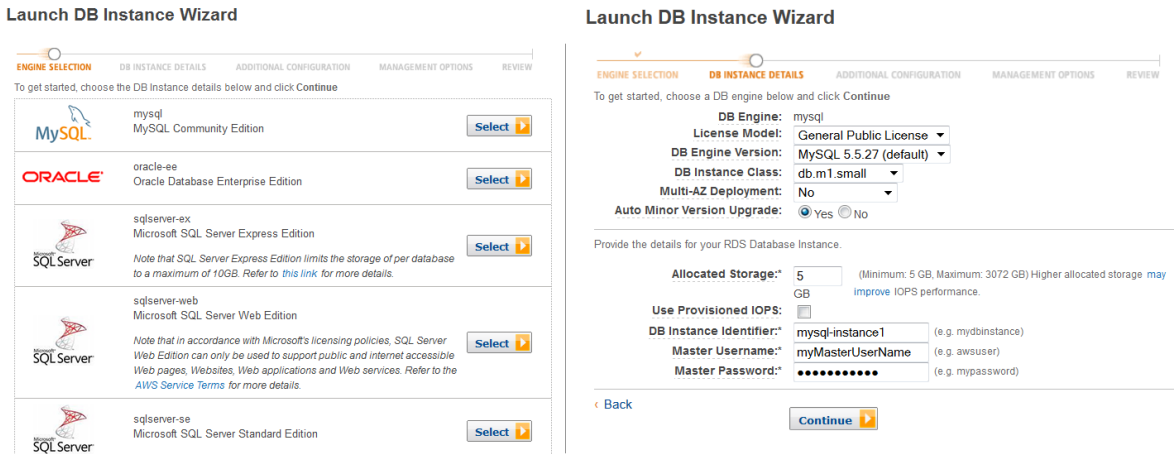


Figura 3.3 Exemplo da interface disponibilizada com o Amazon Relational Database Service

Como foi dito, uma solução do tipo DBaaS é necessária de modo a que seja possível migrar para a *cloud* as aplicações existentes que utilizam bases de dados. Contudo, e apesar de um DW poder ser implementado sobe uma típica base de dados relacional, mais recentemente começou-se a falar num novo modelo, DWaaS, capaz de disponibilizar componentes específicos e necessários para este tipo de sistemas, consistindo assim numa solução integrada de *hardware*, *software* e serviços relacionados. Este tipo de soluções DWaaS podem incluir funcionalidades de monitorização, segurança, manutenção e suporte para todo o ambiente do DW, desde a infraestrutura e arquitetura do sistema, à integração de dados e mesmo até à inclusão de aplicações e utilitários de BI (Barsch, 2012). Atualmente, espera-se que no presente ano de 2013 se assista a novos desenvolvimentos nesta área, já que o final de 2012 ficou marcado pelo lançamento de, pelo menos, duas soluções sonantes como é o caso do Amazon Redshift e do Treasure Data Cloud Data Warehouse. Como diminuem substancialmente o processo tradicional de *data warehousing* (Figura 3.4), estas soluções possibilitam que em pouco tempo se obtenha um DW pronto a funcionar e contam já com algumas histórias e casos de sucesso a seu favor (MobFox, 2013?) (ContextLogic, 2013?). Outro ponto forte nestas soluções é o facto de possibilitarem uma interação com os dados através da bem conhecida e já exaustivamente utilizada linguagem SQL, mesmo que na sua arquitetura subjacente não sejam estritamente utilizados os típicos motores ou sistemas de bases de dados relacionais. Em suma, DWaaS assemelha-se assim a uma evolução de DBaaS adaptada a uma área específica, a de *data warehousing*. Não obstante, a base de ambos passará sempre pela implantação de bases de dados num ambiente em *cloud*, sendo que este problema é ainda um dos maiores desafios a ultrapassar.

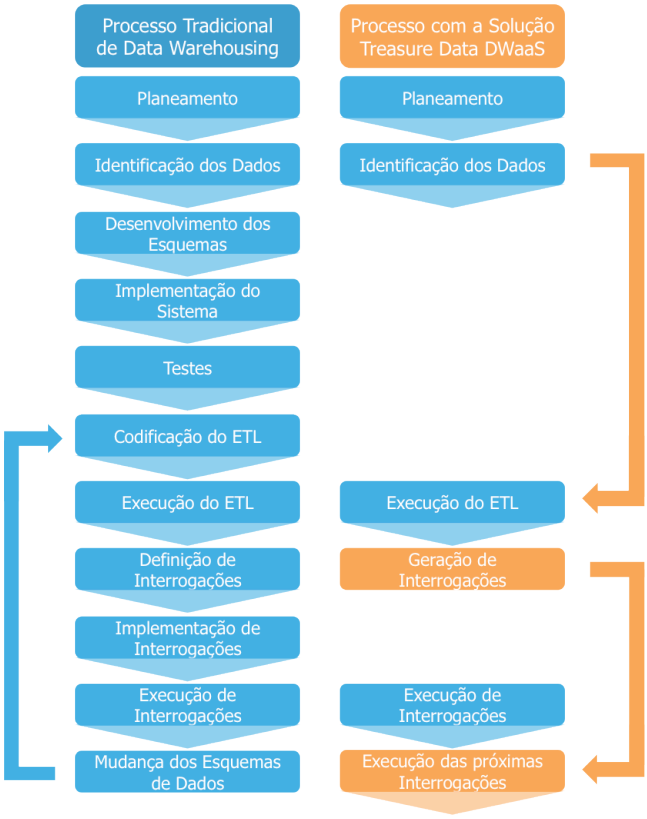


Figura 3.4 Comparação entre o processo de *data warehousing* tradicional e o de DWaaS[51]

3.4 A mudança para a *cloud*

3.4.1 Requisitos e características desejadas nas bases de dados

Face ao domínio das bases de dados na indústria das TI, e à adoção verificada ao *cloud computing*, era normal e expectável que começassem a surgir as primeiras tentativas de implantação deste tipo de sistemas na *cloud*. Apesar de não se ter a certeza se as bases de dados relacionais são o modelo ideal para uma implementação na *cloud*, face ao enorme ecossistema de aplicações existentes e desenvolvidas sobe estas bases, este modelo é aquele que recolhe maior atenção e interesse quando se fala numa possível migração para este ambiente. Na *cloud*, pressupõe-se então que uma base de dados tire proveito dos diversos benefícios deste ambiente

[51] Imagem recolhida e adaptada de (Traditional Data Warehouse and Treasure Data Process Comparison, 201-)

elástico e que simultaneamente ofereça as funcionalidades tradicionais de uma base de dados relacional, de forma a suportar os diferentes tipos de sistemas e aplicações existentes que dependem destas. Assim, entre outros aspectos, quando implementadas na *cloud* estas bases de dados devem (Mathur et al., 2011) (Arora & Gupta, 2012):

- Garantir a consistência e integridade dos dados. Estas duas características são requisitos críticos e essenciais para os sistemas transacionais. Para as garantir, cada transação realizada sobe a base de dados deve estar de acordo com as já conhecidas propriedades ACID[52], mantendo-se assim o estado correto da base após a escrita. Assim, se não existirem garantias da aplicação destas propriedades na *cloud*, não se deve negligenciar a consistência e integridade dos dados em prol das vantagens e benefícios deste ambiente.

- Garantir a segurança e privacidade dos dados. Estes dois aspetos são igualmente cruciais como a consistência e integridade dos dados. Frequentemente os dados são cifrados de forma a proteger a informação sensível e privada dos utilizadores. As bases de dados devem assim possuir mecanismos que garantem aos utilizadores que os seus dados estarão protegidos contra acessos não autorizados.

- Caracterizar-se por serem um sistema tolerante a falhas[53] e com elevada disponibilidade. Sejam sistemas transacionais ou sistemas cujo objetivo é a gestão de grandes volumes de dados para análise, uma base de dados deve estar sempre disponível para os utilizadores, em muitos casos críticos 24/7 durante todo o ano.

- Permitir a portabilidade dos dados. Um dos principais obstáculos na adoção da *cloud* é a dependência existente de um utilizador no seu provedor de serviços, também designada na literatura por *vendor lock-in*. É uma mais valia para os utilizadores que estes consigam trocar de provedor sem quaisquer complicações ou custos de mudança. Para isto, é obrigatório que exista uma fácil portabilidade dos dados entre provedores e que existam APIs[54] padrão, quer seja para o acesso e gestão das diferentes bases de dados existentes na *cloud* ou para a interação com aplicações já existentes no mercado, como é o caso de ferramentas de BI.

[52] ACID (Atomicidade, Consistência, Isolamento e Durabilidade)
[53] Na literatura são designados por Fault Tolerant Systems
[54] API - Application Programming Interface

- Ser escaláveis e apresentar bom desempenho. Uma das principais características da *cloud* é a sua elasticidade que permite aumentar ou diminuir o número de recursos afetados a um serviço sem causar qualquer interrupção na execução do mesmo, fazendo com que o sistema mantenha um bom desempenho face aos pedidos recebidos. Pretende-se assim que as bases de dados sejam capazes de suportar o aumento do volume de dados e o acesso simultâneo de um número ilimitado de utilizadores. Esta escalabilidade pode ser alcançada com a adição de mais máquinas de forma a disponibilizar-se mais recursos.

- Apresentar uma interface simples para interrogação dos dados. Na prática, uma base de dados implementada na *cloud* é distribuída. Assim, para retornar todos os resultados existentes e esperados, pode acontecer que uma interrogação tenha de aceder a dois ou mais nodos da base de dados. É por isso importante que exista uma interface simples e se possível familiar que esconda do utilizador toda a complexidade existente.

Figura 3.5 Características desejadas nas bases de dados implantadas na *cloud*

3.4.2 A problemática da implantação de bases de dados na *cloud*

Muitos utilizadores estão ainda reticentes quando confrontados com uma possível mudança para a *cloud*. Este é um ambiente distribuído bastante diferente do ambiente centralizado no qual se executam as tradicionais bases de dados. A mudança para a *cloud* trouxe assim diversos desafios

que dificultam a concretização das características desejadas para uma base de dados. Como as bases de dados relacionais são já uma tecnologia "antiga", diversos aspectos técnicos, como por exemplo a nível da sua arquitetura, ou questões relacionadas com o seu contexto e modo de operação foram surgindo e revelando-se como inibidores de uma possível migração para a *cloud*. Assim, surgiram novos requisitos necessários para uma correta e convidativa utilização de bases de dados relacionais na *cloud.* Abaixo são abordados alguns desses aspectos, indicando alguns dos problemas inerentes aos mesmos bem como possíveis soluções.

A procura pela arquitetura ideal: *shared-nothing* ou *shared-disk*

As tradicionais bases de dados relacionais, como Oracle ou Microsoft SQL Server, foram primariamente desenhadas para funcionarem em servidores extremamente poderosos e caros que têm total controlo sobre todo o *software* e *hardware* envolvidos. Nesta arquitetura clássica, os problemas de escalabilidade e elasticidade surgem quando o servidor de base de dados começa a ficar sobrecarregado devido ao trabalho existente. Neste caso, a solução passa por dimensionar verticalmente[55] o sistema para aumentar o seu desempenho, ou seja, basicamente são adicionados recursos, como capacidade de processamento ou memória, ao servidor existente ou então é comprada uma máquina ainda melhor e mais poderosa, havendo assim elevados custos no *upgrade* do sistema (Kossmann et al., 2010). Mesmo antes da *cloud*, de forma a atingir novos níveis de escalabilidade, desempenho, eficiência ou disponibilidade, começaram-se a implementar os designados *clusters* de bases de dados, passando assim dum ambiente centralizado para um ambiente distribuído (Hogan, ?a). Ora, as *clouds* são também ambientes distribuídos que oferecem características semelhantes aos *clusters*. De facto, uma das vantagens mais sonantes, a elasticidade, é conseguida com a paralelização das cargas de trabalho existentes e do dimensionamento horizontal[56] do sistema (Abadi, 2009), fazendo com que seja necessário atender-se às arquiteturas existentes de bases de dados paralelas quando estas bases são confrontadas com uma possível migração para a *cloud*. Normalmente, as duas principais arquiteturas utilizadas são a *shared-disk* (SD) e a *shared-nothing* (SN).

[55] Referenciado na literatura como *scale-up* ou *vertical scale*, consiste na adição de recursos a um único nodo ou máquina de um sistema, de forma aumentar o desempenho do mesmo.
[56] Referenciado na literatura como *scale-out* ou *horizontal scale*, consiste na adição de mais nodos a um sistema de modo a distribuir e paralelizar as cargas de trabalho existentes. Normalmente, esta abordagem envolve menores custos que um dimensionamento vertical, pois tira proveito de máquinas comuns, menos poderosas e mais baratas.

Na arquitetura SD a base de dados está acessível por todos os nodos e respectivos sistemas de gestão de base de dados (SGBD), ou seja, existe apenas um único espaço de armazenamento que é partilhado por toda a rede. Como alternativa, na arquitetura SN a base de dados é particionada em fragmentos, sendo que cada subconjunto originado é gerido e acedido exclusivamente por um SGBDs pertence à rede. Assim, cada nodo é independente e autossuficiente e, sabendo que estes não partilham os discos entre si, não existe um ponto de contenção na rede. Atualmente, os principais vendedores de bases de dados já disponibilizam versões dos seus produtos compatíveis com estas arquiteturas.

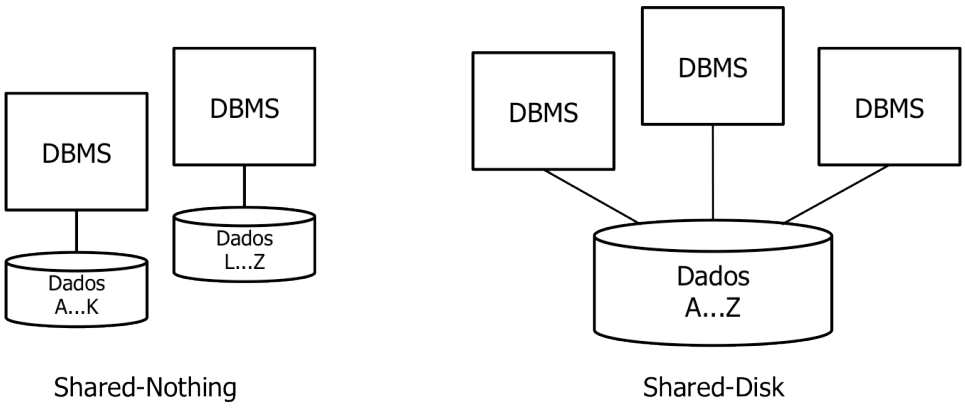


Figura 3.6 Representação das arquiteturas *shared-nothing* e *shared-disk*

Na arquitetura SN, como o armazenamento está distribuído por toda a rede e cada subconjunto de dados é da responsabilidade de um único SGBD, é necessário um *middlware* responsável por encaminhar os pedidos dos utilizadores para o servidor apropriado, ou seja, aquele que tem a informação pretendida. Claro está que em caso de falha de um nodo perder-se-ia por completo o acesso a uma determinada parte dos dados, por isso, em tais situações, o elo que falhe é substituído por outro secundário (nodo *slave*) e que contém uma réplica do mesmo subconjunto de dados existente no primeiro (nodo *master*). Tipicamente, esta promoção é um processo manual e como tal pode ser moroso, enquanto que, em caso de falha, na arquitetura SD basta reencaminhar automaticamente os pedidos para o próximo nodo que esteja disponível. Em termos de escalabilidade, como na *cloud* o dimensionamento horizontal do sistema é a abordagem mais conveniente, a arquitetura SN parece apresentar vantagens sobe a arquitetura SD. Na arquitetura SD existe a passagem de mensagens entre os nodos da rede com o objetivo de alertar cada um sobre o estados dos restantes. Assim, à medida que se aumenta o número de nodos, demora mais

tempo a passar o testemunho entre todos os pontos da rede, resultando assim no aumento do estado de espera em cada nodo e num atraso na resposta aos pedidos. Em teoria, a arquitetura SN apresenta enorme escalabilidade pois é possível aumentar o número de nodos sem causar qualquer interferência ou conflito, visto que cada ponto da rede é independente. Porém, a adição de um novo nodo requere o particionamento cuidado dos dados, de modo a minimizar o efeito de *data shipping*[57]. A ideia é diminuir a necessidade de executar transações e junções de dados entre nodos, pois isto aumentaria a latência e originaria pontos de estrangulamento na rede. A independência de cada novo faz com que seja difícil fazer um efetivo e dinâmico balanceamento de carga na arquitetura SN. Nesta, cada nodo é responsável por acomodar toda a carga de trabalho existente para os seus dados, o que faz com haja a necessidade de investir em mais máquinas de forma a dividir o trabalho existente. Se a nova máquina adicionada for um *master*, é necessário proceder ao particionamento dos dados, porém podem ser adicionados *slaves* com cópias dos dados para acomodar apenas pedidos de leitura. Já no caso da arquitetura SD, como todo o conjunto de dados está partilhado, qualquer servidor de base de dados pode satisfazer um pedido, sem ser necessário realizar qualquer mudança. Assim, o balanceamento de carga é feito de uma forma mais suave e adapta-se melhor às alterações nas cargas de trabalho. Estas duas arquiteturas diferem ainda num aspeto muito importante para as bases de dados: a consistência dos dados (Hogan, ?b). Na arquitetura SD, todos os nodos são alertados e esperam pelo término de uma transação antes de poderem ler os dados. Como apenas é utilizada uma cópia dos dados, o armazenamento partilhado, não existe qualquer possibilidade de um nodo ler dados desatualizados ou inconsistentes. Os problemas surgem assim na arquitetura SN quando se replicam os dados para dar origem às máquinas *slaves*. Numa rede de longa distância, na prática não é possível garantir a consistência dos dados e a disponibilidade do sistema simultaneamente (Gilbert & Lynch, 2002) (Abadi, 2009). Normalmente, além de promoverem alta disponibilidade na substituição de um *master* em caso de falha, estas máquinas auxiliam na distribuição das cargas de trabalho, respondendo assim apenas aos pedidos de leitura. Em caso de escrita num *master*, os dados são replicados pelos *slaves* de forma assíncrona, não sendo assim possível garantir que, numa próxima leitura, os dados lidos duma destas máquinas estejam já atualizados e consistentes. Uma replicação síncrona pode trazer sérios problemas de desempenho e de disponibilidade, pois pode-se tratar de uma operação demorada devido à distante distribuição da rede e ao volume de dados envolvidos na replicação.

[57] *Data shipping* – passagem de dados duma máquina para outra para serem processados.

Arquitetura *Shared-Disk*	**Arquitetura *Shared-Nothing***
Existe apenas um único espaço de armazenamento que contém todo o conjunto de dados. Todos os nodos têm acesso a toda a informação existente.	Conjunto de dados é fragmentado e distribuído pela rede, havendo um único nodo responsável por cada parte. Necessidade de particionamento dos dados.
Tempo de propagação de mensagens e sinais limita o número de nodos existentes.	Permite escalabilidade ilimitada quando os dados são eficientemente particionados.
Como o número de máquinas é limitado, pode ser necessário dimensionar verticalmente.	Dimensionamento horizontal permite a utilização de máquinas menos dispendiosas.
Balanceamento de carga dinâmico e suave pois qualquer nodo pode responder aos pedidos.	Maior investimento em máquinas para melhorar o balanceamento de carga.
Garante a consistência dos dados.	Devido à replicação, poderá ocorrer inconsistência nos dados.

Tabela 3.2 Resumo comparativo das arquiteturas *shared-disk* e *shared-nothing*[58]

Nas arquiteturas SN há a tendência para utilizar o disco local do próprio servidor para alojar o fragmento de dados que este gere, ao passo que na SD o armazenamento está partilhado e acessível através da rede. Assim, anteriormente quando existia o problema de comunicações lentas pela rede, a arquitetura SN apresentava benefícios consideráveis em termos de desempenho. Hoje em dia tal premissa não é verificável e, como foi possível ver em cima pelos aspetos discutidos, é difícil de indicar qual destas duas arquiteturas tem vantagens em relativamente à outra. A verdade é que existem argumentos sólidos e diversas vantagens de ambos os lados. Em suma, qual destas arquiteturas é a melhor ou qual é que deve ser usada, é uma questão sem resposta imediata que depende muito do tipo de aplicações que se pretende implementar sobe as bases de dados e do tipo de cargas de trabalho existentes.

Instabilidade e imprevisibilidade da *cloud*

No seio de uma empresa, cada vez mais os dados assumem um papel de maior relevo e suportam os processos críticos de negócio, onde a mínima falha pode desencadear grandes prejuízos. A durabilidade e disponibilidade dos dados deve ser assegurada pois, assim como qualquer outro ambiente ou sistema, a *cloud* está sujeita a falhar. Normalmente, estas características são alcançadas com a replicação automática dos dados por grandes distâncias geográficas, como por exemplo *datacenters* espalhados por todo o mundo (Abadi, 2009). Sabendo que o utilizador não

[58] Tabela construída com base em (Hogan, ?b), (Lee, 2011), (Arora & Gupta, 2012).

tem qualquer controlo ou conhecimento da replicação dos dados nem da sua localização exata, esta distribuição geográfica pode ser problemática ou mesmo proibitiva na adoção de uma solução na *cloud* nos países em que existem leis que impõem restrições relativas à residência dos dados, pois os utilizadores podem não ter a oportunidade de exigir que estes permaneçam dentro dos limites legais estipulados. A *cloud* pode revelar-se ainda como um ambiente imprevisível no que diz respeito aos níveis de desempenho oferecidos. Como os nodos computacionais podem ser heterogéneos, estes não apresentam todos as mesmas características e desempenho, pelo que a execução degradada de um pode baixar significativamente o desempenho global do sistema. Também o acesso concorrente de um número elevado de utilizadores, especialmente para operações de escrita em discos tradicionais partilhados, ou a constatação de velocidades irregulares nas comunicações, resultantes por exemplo do afastamento geográfico entre nodos, são exemplos de situações nas quais se pode observar uma diminuição do desempenho (Gelder, 2011?). Uma maneira que os utilizadores têm de se precaver destas contrariedades da *cloud* é através da contratação dos SLAs para base de dados, porém atualmente existem poucos provedores a disponibilizarem estes acordos para os seus serviços.

Perda de controlo sobre os dados

Sempre que se fala de dados, são sempre equacionadas as questões de segurança e privacidade dos mesmos, sendo que na *cloud* estes dois aspectos ganham ainda maior importância. Isto porque com a adoção deste ambiente, o utilizador está a confiar os seus dados a uma terceira entidade, perdendo assim algum controlo sobre os mesmos. Como são utilizadas para registar e executar diversas operações de negócio, é frequente e necessário guardar-se informação sensível e operacional nas bases de dados, como é o caso de números de cartão de crédito ou informações privadas. Nestes contextos, um aumento dos riscos de segurança e de violação de privacidade é inaceitável. O utilizador necessita de ter garantias de privacidade dos seus próprios dados, por exemplo, com recurso a técnicas de encriptação e de segurança, de modo a que todos os meios de comunicação com a sua informação estejam apenas acessíveis e utilizáveis por si, independentemente do espaço de armazenamento poder ser comum a diversos clientes de diferentes organizações. A privacidade e segurança na Internet é já um tema exaustivamente abordado e estudado, principalmente no que diz respeito à segurança nas transmissões e comunicações de dados. Alguns problemas técnicos podem assim surgir quando for necessário lidar com dados armazenados cifrados, pois normalmente são necessárias abordagens diferentes

de análise e processamento dos mesmos (Gelder, 2011?). Este é um mal necessário pois, mesmo que os dados sejam extraviados, é preciso garantir que os seus "novos donos" não consigam entender ou interpretar a informação que têm em mãos. Outra questão preocupante resultante da mudança para a *cloud*, apesar de não ser uma operação frequente nem fazer sentido em diversos casos de aplicação, é a capacidade do utilizador em ter a certeza se consegue efetiva e definitivamente apagar os seus dados deste ambiente quando e sempre que pretender (Slack, 2011).

Problemas nas comunicações de dados

Como se acede à *cloud* através de uma rede, por exemplo a Internet no caso de ambientes públicos, este meio tem de estar constantemente funcional e tem de oferecer boas velocidades de transmissão, de forma a contribuir também para a disponibilidade do sistema. Como estes meios de comunicação não são dedicados e, como tal, não são tão rápidos como numa conexão local, podem ocorrer estrangulamentos na rede quando se estiverem a realizar transferências de grandes volumes de dados, agravando-se a situação sempre que se tratarem de transferências entre grandes distâncias geográficas. Além das velocidades de transmissão e apesar da *cloud* estar frequentemente associada à disponibilização de serviços a preços reduzidos, também é preciso ter em conta os valores cobrados na transferência de dados para a *cloud*. Novamente, quando estão envolvidos grandes volumes de dados, esta transferência pode tornar-se um processo bastante caro, havendo mesmo registos de que, no passado, por vezes a solução mais barata passava mesmo pelo transporte físico dos dados em disco até ao destino (Gelder, 2011?). É claro que se podiam aplicar técnicas de compressão de forma a diminuir os volumes transferidos, porém, como não existem soluções perfeitas, pagar-se-ia sempre um preço pelo processamento adicional realizado.

3.4.3 As diferenças entre os *Data Warehouses* e os sistemas transacionais na *cloud*

No início da secção 3.1 foi possível perceber que existem dois principais tipos de sistemas ou aplicações diferentes que são utilizadas para a gestão de dados. Como existem diferenças nas suas necessidades e no seu modo de operação apesar de ambas utilizarem bases de dados relacionais, nessa secção foram ainda apresentadas algumas dessas divergências que caracterizam de uma forma geral o leque de aplicações de ambos os tipos. Ora, sendo os DWs exemplos específicos de

sistemas de gestão de dados analíticos, estes apresentam algumas peculiaridades próprias que não foram apresentadas e exploradas até agora, e que são factores importantes a considerar aquando de uma possível mudança para a *cloud*. Isto porque apesar dos problemas apresentados na secção 3.4.2 afetarem bases de dados em geral, pode ser que alguns destes não se verifiquem ou possam ser minimizados consoante o tipo de aplicação que se pretenda implementar na *cloud,* exatamente devido às diferenças existentes entre DWs e outros sistemas transacionais, como é o exemplo das aplicações OLTP (Online Transaction Processing).

Como os DWs necessitam de dar resposta imediata aos agentes de decisão, as questões de desempenho são consideradas logo na fase de desenho dos dados. Assim, os esquemas são desnormalizados dando origem a esquemas em estrela ou em floco de neve[59]. Desta forma, e ao contrário do que acontece nos esquemas normalizados, diminui-se ao máximo a necessidade de junção de tabelas e, consequentemente, melhora-se o tempo de resposta das interrogações realizadas sobre os dados. Como se sabe, a desnormalização é um problema quando se pretende realizar escrita ou atualização dos dados (Connolly & Begg, 2004a). Porém, outra diferença dos SDWs para os OLTP, é que os primeiros são bases essencialmente de leitura, reduzindo assim os problemas originados com a desnormalização dos esquemas. A nível de arquitetura, diversos vendedores na área dos DWs há muito estabelecidos no mercado, como a Teradata[60], Netezza[61] ou Greenplum[62], têm disponibilizado as suas soluções baseadas numa arquitetura SN (Abadi, 2009). Esta é frequentemente associada como sendo uma arquitetura de processamento paralelo massivo[63], utilizada para a satisfação rápida de interrogações complexas através da execução de técnicas paralelas de análise, junção e ordenação de forma a permitir que diversos processadores partilham as cargas de trabalho existentes (Connolly & Begg, 2004b). Devido ao bom desempenho e à possibilidade de escalabilidade infinita que oferece, esta arquitetura é ideal para aplicações *read-only*, como é o caso dos DWs, e dada a sua natureza distribuída também parece ser a mais adequada para a *cloud* (DeWitt et al., 200-) (Abadi, 2009) (Madsen, 2012) (Klopp, 2012).

Outro grande problema apontado às arquiteturas SN, a verificação das propriedades ACID, é minimizado com a implementação de DWs nesta arquitetura. Como o SDW é atualizado com

[59] Na literatura são conhecidos por *Star schema* e *Snowflake schema*, respetivamente
[60] http://www.teradata.com/
[61] http://www-01.ibm.com/software/data/netezza/
[62] http://www.greenplum.com/
[63] Massively Parallel Processing (MPP)

informação proveniente dos SOs, esses dados estão à partida consistentes. Além disto, aspectos como o reduzido número de escritas e o facto de, em muitos casos, ser suficiente a análise de dados recentes sem estarem atualizados até ao último segundo, eliminam a necessidade de verificar as propriedades ACID sobre um DW (Abadi, 2009). Na realidade, um dos principais desafios a ultrapassar com a implementação de DWs em arquiteturas SN na *cloud* diz respeito à replicação e ao particionamento dos dados. Tradicionalmente, estes dois processos são feitos de forma estática pelos administradores da base de dados, pelo que, na *cloud*, terão de ser feitos de uma forma mais dinâmica de modo a obter-se flexibilidade e estabilidade (Gelder, 2011?).

Outra grande diferença existente entre os DWs e os sistemas OLTP é o tipo de dados que estes utilizam e que são guardados nas bases de dados. Como já foi dito, os sistemas OLTP são aquele tipo de aplicações que necessitam de guardar e aceder a informação privada ou confidencial dos utilizadores, de forma a executarem habituais processos de negócio (execução de transferências bancárias, reservas online, etc.). No caso de um DW, normalmente este tipo de informação é deixada de parte, pois os dados confidenciais dificilmente serão considerados como preponderantes para análise. Mais uma vez, este é um aspecto que apenas permite diminuir um pouco as preocupações existentes, pois nem no caso dos DWs na *cloud* se podem descartar por completo as questões de privacidade e segurança dos dados. Por outro lado, invés de se sentirem menos seguras, se calhar as organizações deviam se sentir mais confiantes com a mudança para a *cloud*, pois os provedores podem ter melhores conhecimentos e capacidades para lidar com estas questões de segurança do que a própria organização numa situação *in-house*. Atendendo aos diversos motivos apresentados nesta secção, parece que os DWs estão mais propícios para serem implementados na *cloud*, pois, em alguns aspectos, a sua migração ocorrerá de uma forma mais suave do que para os sistemas transacionais. Não obstante, como nem todos os problemas têm ainda uma solução clara, terá de haver um esforço por parte dos provedores para garantirem aspectos como a qualidade do serviço, serviços de monitorização e modelos que permitam ao utilizador pagar apenas consoante o que utilizou, de forma que se consiga trazer todos os benefícios da *cloud* para os DWs.

3.5 *Data Warehouses* e *clouds* privadas

A migração de aplicações tradicionais para as *clouds* públicas pode não ser um processo simples, isto porque muitas dessas aplicações simplesmente não foram desenhadas para funcionarem num

ambiente público. Os requisitos de negócio existentes, o volume das cargas de trabalho ou a sensibilidade dos dados são alguns factores fundamentais para decidir qual dos ambientes, público ou privado, é o mais adequado para uma dada aplicação e com qual se pode obter mais benefícios. Claro que esta decisão depende sempre da indústria e do contexto no qual um dado negócio é executado. Por exemplo, na área dos serviços financeiros há a tendência para preferirem ambientes que possibilitem maior controlo sobre os dados, devido aos diversos requisitos legais, de segurança e de privacidade existentes. As *clouds* privadas são a sua escolha quando confrontados com este ambiente, quer seja para a implementação de sistemas transacionais ou de SDWs (Madsen, 2012). Além deste, outros estudos e inquéritos mostraram também preferência pela implementação de DWs em *clouds* privadas (Russom, 2011).

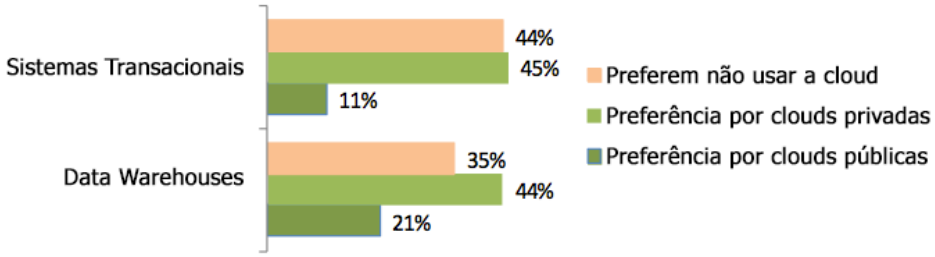


Figura 3.7 Preferência dos agentes de decisão por *clouds* privadas nos serviços financeiros[64]

Para o caso específico dos DWs, as *clouds* privadas possibilitam várias das vantagens e dos benefícios existentes nas *clouds* públicas, mas sem alguns dos problemas associados, principalmente quando se tratam de *clouds* que são propriedade da própria organização ou que são geridas internamente. Como se sabe, a infraestrutura utilizada nas *clouds* públicas é muito diferente do *hardware* convencional utilizado para *data warehousing*. A *cloud* é construída com máquinas comuns de baixo custo ao invés de servidores de topo e poderosos, sendo que tipicamente também não existem ligações extremamente rápidas e dedicadas entre nodos pois estes podem estar dispersos a grandes distâncias uns dos outros (Madsen, 2012). A existência de discos extremamente rápidos e diretamente ligados aos nodos também é rara pois, como por exemplo acontece numa arquitetura SD, o armazenamento está acessível através de uma rede, como uma SAN (Storage Area Network) (Arora & Gupta, 2012). Estas diferenças podem afetar e diminuir significativamente o desempenho dos SDWs na *cloud* quando comparados com uma

[64] Imagem extraída e adaptada de (Madsen, 2012)

execução *in-house*. A vantagem das *clouds* privadas é de que, tratando-se de um ambiente mais controlado, consegue-se disponibilizar recursos dedicados e consequentemente oferecer melhores condições para suportarem as diferentes e por vezes imprevisíveis cargas de trabalho existentes no âmbito dos DWs. O facto de ser um ambiente dedicado acaba ainda com o modelo *multi-tenant* utilizado nas *clouds* públicas, que muitas vezes obriga à execução de práticas adicionais de forma a promover confidencialidade, privacidade e segurança. De lembrar que com este modelo, como muito provavelmente um nodo na *cloud* não é um servidor físico mas sim uma de muitas máquinas virtuais que corre numa só máquina física, enquanto que a memória interna, algum espaço em disco e capacidade de processamento são alocados exclusivamente para cada instância virtualizada, outros componentes como redes ou armazenamento externo são partilhados muitas vezes entre organizações diferentes. Como já foi indicado (secção Perda de controlo sobre os dados), esta partilha não é um aspecto muito favorável aos olhos das organizações, pois daí deduzem alguns problemas e riscos de segurança e privacidade. De facto, a necessidade em manter algum controlo sobre os dados é um dos principais aspetos apontados para a escolha das *clouds* privadas (Russom, 2011) (Madsen, 2012). Desta forma os utilizadores conseguem por exemplo ditar onde os dados ficarão armazenados, podendo assim respeitar questões e limitações governamentais ou legais. Outro aspecto bastante apreciado é o facto de, se a *cloud* for gerida internamente não há mesmo a necessidade de entregar os dados a uma entidade terceira.

As *clouds* privadas oferecem o controlo necessário sobre o ambiente computacional, permitindo assim superar as principais questões e preocupações que inibem os utilizadores de escolher uma *cloud* pública, inclusive, permitem que se atinjam níveis semelhantes de escalabilidade e desempenho verificados num ambiente público. É assim normal que este ambiente comece a ser prioridade para o âmbito dos DWs enquanto que as *clouds* públicas não conseguirem transmitir a confiança necessária para a execução destes sistemas.

Capítulo 4

AC2DC - A *Cloud* como Fonte de Dados para um Sistema de *Dashboards* Auto-adaptáveis

4.1 Definição do caso

4.1.1 Contextualização prática

O tema desta dissertação teve origem num projeto realizado no âmbito de uma bolsa de iniciação científica financiada pela Portugal Telecom Inovação, SA, de acordo com um protocolo estabelecido com a Universidade do Minho para cooperação científica e técnica. O projeto era designado por AC2DC – Componentes Analíticos Auto-Adaptáveis para *Cloud Based Data Warehouses*, e tinha uma duração total de seis meses, nos quais se pretendia levar a cabo a especificação e implementação de componentes analíticos para a exploração de estruturas de dados multidimensionais mantidas em ambientes de *data warehousing* suportados por nuvens computacionais. O foco principal deste projeto consistia na implementação de técnicas que permitissem a reestruturação e adaptação automática de *dashboards* com base nas preferências de utilização dos agentes de decisão. Porém, um dos componentes essenciais de todo o sistema, a fonte de dados para os painéis de exploração, continha uma particularidade tecnologicamente recente e pouco habitual nos ambientes e aplicações de suporte à decisão: o facto dos dados serem disponibilizados a partir de um DW assente numa *cloud*. Assim, para abordar esta questão

em particular, o objetivo principal deste projeto centrou-se na investigação e desenvolvimento de novas formas de interação com DWs implantados em *clouds*, no sentido de que fosse possível angariar toda a informação pertinente para as diversas vertentes de análise e de tomada de decisão estabelecidas para os diversos agentes de decisão envolvidos. O projeto foi, assim, dividido em duas grandes partes: uma primeira relativa à implementação da *cloud*, angariação e provisionamento dos dados através da mesma, e uma segunda parte relativa à exploração e visualização dos dados através de painéis de *dashboards*.

De uma forma resumida, a secção 4.1.3 irá expor e descrever todo o sistema e respectivos componentes desenvolvidos, de modo a que seja possível perceber toda a sua constituição e funcionamento em geral. Contudo, em coordenação com o tema desta dissertação, nas restantes secções deste capítulo será dado enfâse apenas à implementação da nuvem computacional e dos processos diretamente relacionados com a mesma, que são fundamentais para a execução do sistema.

4.1.2 Objetivos e tarefas a realizar

No início da realização deste projeto foi definido um conjunto específico de tarefas com vista a concretizar a parte do sistema responsável pela angariação e distribuição de dados através da *cloud* implementada. Estas tarefas refletem funcionalidades diretamente relacionadas com os processos de exploração e manutenção de um SDW, realizados por intermédio de agentes computacionais. Com elas, o objetivo era o de demonstrar a viabilidade da instalação de um SDW numa *cloud*, revelando o seu correto funcionamento à semelhança do que acontece num ambiente tradicional. De forma mais detalhada, pretendia-se realizar um conjunto de tarefas específicas relacionadas, em particular, com a:

- Instalação, configuração, exploração e manutenção de um DW corporativo numa *cloud* privada de dados.
- Concepção e desenvolvimento de uma comunidade de agentes de *software* capazes de:
    - Angariar vistas de dados multidimensionais em vários sistemas fonte, conciliar essas vistas numa única estrutura de dados global e armazená-la na *cloud*, colocando-a disponível para exploração.

- Coletar na *cloud* as vistas multidimensionais necessárias ao povoamento e refrescamento das plataformas de processamento analítico.

De realçar que, apesar de este ser o cenário mais apelativo, o objetivo não era a realização de uma solução DWaaS. Primeiro, porque esta não era o tipo de solução desejada e, segundo, porque a elaboração dos diversos serviços extra associados à mesma (ex.: serviços de manutenção, monitorização, mecanismos de segurança, interfaces de utilização, etc.) seria um processo complexo e moroso, que só provocaria atrasos significativos no projeto e no protótipo pretendido, cuja necessidade era o provisionamento "simples" de dados através de uma *cloud*. Assim, o cenário implementado pode por exemplo assemelhar-se à criação de uma *cloud* utilizada e gerida pela própria organização pois, como se tratam de serviços para consumo interno, não é necessário o desenvolvimento de certo tipo de facilidades necessárias para comercialização externa.

4.1.3 Descrição e funcionamento geral do sistema desenvolvido

Como já foi brevemente referenciado, o projeto AC2DC visava a construção de componentes analíticos capazes de se auto-adaptarem às tendências de exploração dos agentes de decisão, com base nas suas preferências de utilização. O início do projeto começou então pela definição de um sistema capaz de suportar processos convencionais de um SDW, atendendo sempre ao facto de a fonte de dados principal, o DW, estar localizado e alojado numa *cloud* privada e não num sistema de dados convencional, como ainda acontece na generalidade dos casos, mesmo quando se fala imenso na emergência de novas abordagens para o armazenamento de dados neste tipo de sistemas. Ao longo desta dissertação já se falou dos benefícios e da motivação (secção 3.2) que leva à implantação/migração de SDWs para este tipo de ambientes elásticos e, como tal, é possível afirmar que com esta opção se está a garantir uma forma muito atual, expedita e robusta de providenciar a escalabilidade necessária a este tipo de sistemas de dados, não só no acolhimento das suas estruturas de dados e respetiva informação, mas também no número de acessos e interrogações que sobre eles podem ser lançadas.

A *cloud* do sistema foi então concebida como um repositório global de dados capaz de acolher informação proveniente de várias instâncias de um DW corporativo, à semelhança do que, por vezes, acontece na indústria, em que essas instâncias estão distribuídas ao longo das diversas instalações de uma empresa. Assim, no lugar dos típicos SOs que alimentam os DWs, neste caso existem instâncias de DWs a desempenhar esse papel. Apesar desta nova situação, os processos

de angariação de dados mantêm-se muito semelhantes aos realizados sobre os típicos SOs, sendo que neste caso a situação é um pouco mais facilitada, pois, à partida, é menor a necessidade de limpeza e tratamento dos dados durante os processos ETL uma vez que, como os dados são provenientes de DWs, estes já refletem algum cuidado e apresentam os níveis de qualidade adequados. No contexto deste projeto, a *cloud* atua assim como um sistema global de conciliação de dados, acessível a todos os agentes de decisão empresariais, independentemente do local em que cada um deles está situado. Perante isto, imediatamente foram identificados os processos necessários para alimentar a *cloud* e para realizar a sua exploração. Isto levou à divisão do sistema em dois subsistemas:

- O sistema montante, caracterizado por tratar de tudo o que fosse necessário realizar para fazer o povoamento correto da *cloud*. Aqui, essencialmente executam-se dois tipos de tarefas: a angariação de dados provenientes de diversas fontes de informação; e a conciliação desses mesmos dados numa única estrutura global localizada na *cloud*. O objetivo deste sistema é fazer com que a *cloud* seja auto suficiente, no sentido em que contém toda a informação necessária pra conseguir satisfazer todas as interrogações de dados realizadas pelos agentes de decisão nos painéis de exploração.

- O sistema jusante, responsável pela realização de todas as tarefas necessárias para a apresentação dos dados em painéis de exploração. Semelhante ao sistema montante, neste leque também se identificam duas tarefas principais: o provisionamento dos dados para estruturas multidimensionais que são processadas com informação diretamente extraída da *cloud* e que dão resposta às interrogações dos decisores; e a visualização de dados, caracterizada pela apresentação e funcionamento dos painéis de *dashboards*, armazenamento dos respetivos registos de utilização, além da reestruturação desses mesmos painéis.

O sistema montante e o jusante permitem também perceber o sentido unidirecional pelo qual os dados fluem no sistema. De forma alusiva a um rio, os dados correm no sistema desde as fontes de informação, passando pela a *cloud* e terminando a jusante, nos painéis de exploração.

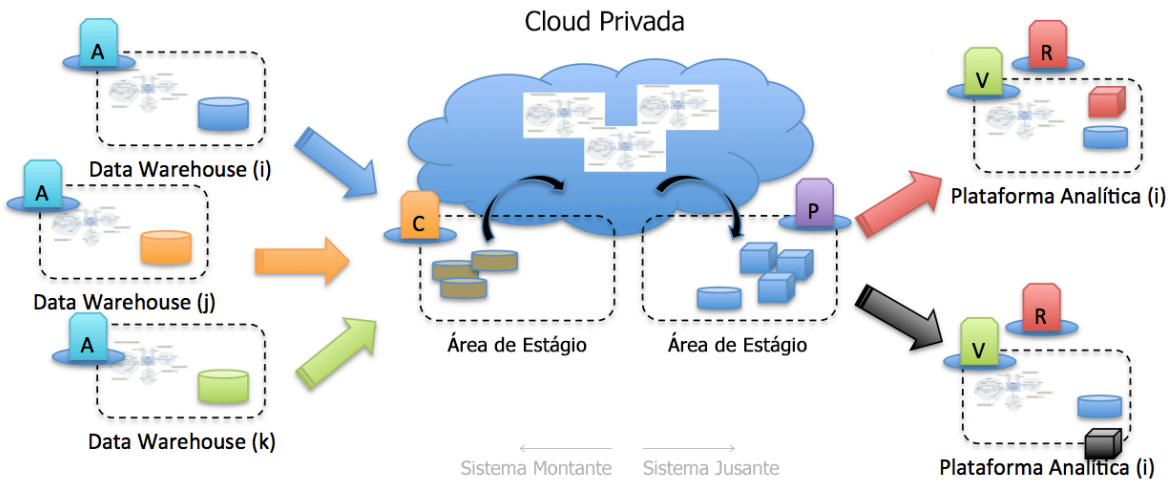


Figura 4.1 Arquitetura geral do sistema AC2DC

A Figura 4.1 Arquitetura geral do sistema AC2DC, representa um possível cenário de utilização do sistema AC2DC. Nela é possível visualizar um conjunto de unidades computacionais (agentes) identificados por letras do abecedário, e que fazem parte do sistema multiagente desenvolvido. Seguindo uma linha de concepção e desenvolvimento similar à que deu origem à *cloud*, optou-se pela implementação destes agentes de modo a que todo o sistema fosse escalável, para acompanhar, tal como a nuvem, o crescimento natural do sistema, robusto, no qual a operacionalidade dos seus componentes e serviços não fosse colocada, em nenhum momento, em causa, e autónomo, para que os seus processos pudessem executar de forma distinta e específica, quer sejam relativos aos pontos de recolha de dados ou às plataformas que suportam os painéis de exploração.

O sistema multiagente disponibiliza um meio capaz de acolher um conjunto de unidades computacionais heterogéneas que podem estar distribuídas por diferentes localizações físicas, em número e competências variáveis ao longo do tempo e que podem ser executadas em paralelo, partilhando recursos e cooperando entre si em prol de um objetivo comum. Por estes motivos, são um modelo de computação bastante adequado para o tipo de problemas presentes neste projeto e, como tal, foram desenvolvidas cinco comunidades de agentes para atuarem no sistema (ver Figura 4.1):

– (A) Agentes Angariadores: que são responsáveis por fazer a coleta dos dados necessários para alimentar os vários painéis de exploração dos utilizadores a partir das diversas fontes

de informação (neste caso, instâncias de um DW corporativo), de acordo com as suas agendas locais, e, depois, enviá-los para o respectivo agente de conciliação.

- (C) Agentes Conciliadores: são os agentes responsáveis pela conciliação da informação que é angariada nas diversas fontes de informação pelos agentes angariadores, transformando-a de acordo com as diretivas previamente definidas que determinam a forma como as estruturas multidimensionais de dados contidas na *cloud* são povoadas.

- (P) Agentes Provedores: atuam de forma muito similar aos conciliadores mas numa direção completamente oposta, pois coletam a informação da *cloud* e armazenam esses dados localmente nas suas bases multidimensionais em hipercubos específicos, para que, mais tarde, possam processar e satisfazer as interrogações enviadas pelos agentes de decisão.

- (R) Agentes Reestruturadores: são os componentes mais críticos do sistema, já que são eles os responsáveis por analisar regularmente - num intervalo de tempo definido previamente – os dados de registo relacionados com todas as consultas multidimensionais lançadas a partir dos painéis de *dashboards* durante uma ou mais sessões de exploração de um utilizador e, depois, definir quais os novos dados (e metadados) que se deverão mostrar posteriormente no mesmo sistema de painéis. Este tipo de ação de reestruturação é, ao fim e ao cabo, uma operação típica de personalização dos diversos painéis à disposição às necessidades (preferências) de exploração dos utilizadores.

- (V) Agentes Visualizadores: estes agentes são ativados sempre que uma plataforma de exploração analítica é colocada em execução. São os responsáveis pela estrutura e conteúdo dos vários painéis que a plataforma de exploração incorporar, tendo nas suas agendas de trabalho as ações e interrogações multidimensionais que devem lançar para alimentar os diversos *dashboards* integrados nos painéis de exploração da plataforma.

Em suma, os agentes apresentados são os responsáveis por realizar as interações necessárias entre os diversos componentes do sistema AC2DC. No final, o resultado apresentado aos agentes de decisão são painéis constituídos por *dashboards* que apresentam e disponibilizam os dados de negócio para exploração. Para isto, cada painel conta ainda com um conjunto de funcionalidades

que permitem a sua personalização, como é o caso da mudança do aspeto de um *dashboard* ou dos dados apresentados por este.

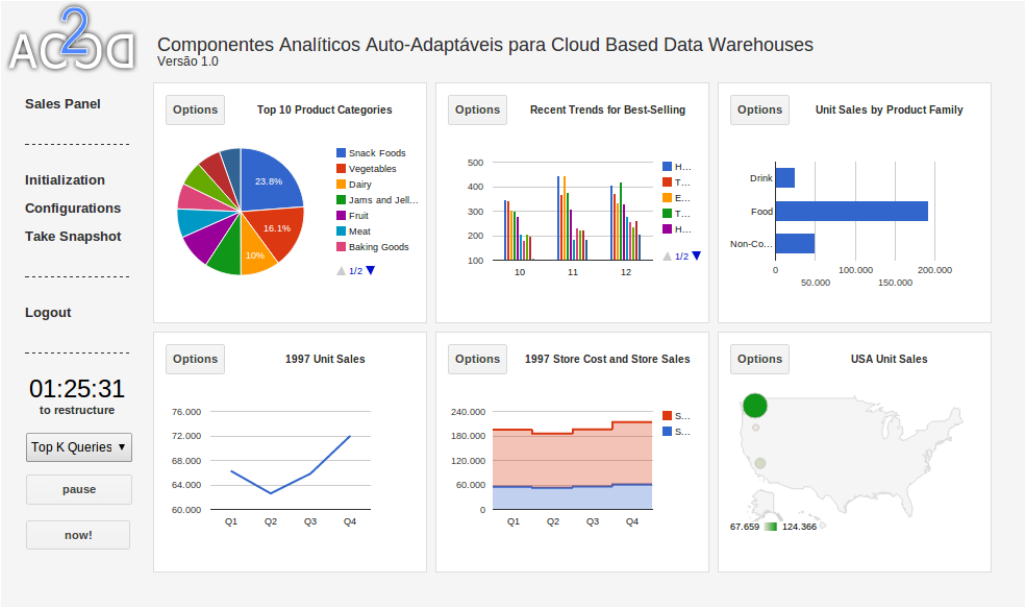


Figura 4.2 Exemplo de um painel de exploração disponibilizado para os agentes de decisão

4.1.4 Tecnologias e ferramentas utilizadas

Como este projeto surgiu no âmbito de uma bolsa financiada por uma entidade externa, houve um conjunto de regras e limitações que tivemos de respeitar durante a sua execução. Por exemplo, um dos requisitos iniciais para este projeto ditava que qualquer tecnologia utilizada fosse *open-source*. Além deste factor, tentamos sempre selecionar apenas tecnologias e ferramentas de referência, sendo muitas delas atuais e utilizadas na indústria em aplicações concretas. A tabela abaixo apresenta assim as principais ferramentas utilizadas no desenvolvimento da *cloud* e dos respetivos componentes e processos associados. Claro que, apesar de não estarem listadas, foram utilizadas diversas tecnologias secundárias que faziam parte dos pré-requisitos para o funcionamento de algumas ferramentas escolhidas.

Tecnologia/Ferramenta	Propósito
OpenNebula	Implementação, configuração e manutenção da *cloud* e respetivos recursos disponibilizados.

JADE[65] (Java Agent DEvelopment Framework)	Implementação e gestão do sistema multiagente.
Mondrian[66] – OLAP Server	Implementação dos sistemas de processamento analítico que contêm as estruturas multidimensionais de dados.
PostgreSQL	Implementação do *Data Warehouse* assente na *cloud*.
JAVA[67]	Implementação do comportamento e rotinas dos diversos agentes.

Tabela 4.1 Tecnologias e ferramentas utilizadas no projeto prático

4.2 A nuvem do sistema

4.2.1 O sistema escolhido: OpenNebula

Atendendo aos requisitos fundamentais do projeto e à necessidade de implementação de um SDW numa *cloud*, optamos por procurar um produto de *software* cujas características permitissem auxiliar e acelerar a construção dum ambiente deste tipo. A seleção de um produto avizinhava-se como sendo um desafio complicado pois, nesta área, as soluções de *software* existentes são relativamente recentes e estão frequentemente a ser atualizadas com o lançamento de novas versões e funcionalidades. Posto isto, de forma a facilitar este processo de escolha, inicialmente apenas foram colocados em "jogo" dois requisitos considerados cruciais para a realização das tarefas que tinha em mão, nomeadamente:

- o sistema para a criação e manutenção da *cloud* para o SDW deveria ser um software *open-source*;
- era fundamental a possibilidade de criação de *clouds* privadas, de modo a promover um acesso exclusivo aos dados, como usualmente é requerido no ambiente de um DW corporativo.

Como primeira abordagem, optamos então por procurar ferramentas de gestão de infraestruturas virtuais, vulgarmente conhecidas como *cloud toolkits*, que permitissem a criação de *clouds* privadas ou híbridas, sabendo que estas últimas são muito referenciadas na indústria. Assim,

[65] http://jade.tilab.com/
[66] http://mondrian.pentaho.com/
[67] http://www.java.com/

várias soluções e produtos foram identificados e estudados, muitas das quais pertencentes a empresas líderes na área de virtualização, como a Citrix[68], VMware[69] ou Microsoft[70], que se revelaram possíveis candidatas. Como seria de esperar, todas as soluções estudadas têm um princípio de funcionamento semelhante e contemplam as ditas funcionalidades básicas que caracterizam este tipo de ferramentas. Essencialmente, diferenciam-se pelas funcionalidades avançadas que disponibilizam e pela tecnologia específica e, por vezes, proprietária que utilizam e que pode facilitar o processo de integração numa dada organização. Como verificava a nossa pretensão de fazer uma implementação base de uma *cloud* privada através de uma ferramenta *open-source*, a solução escolhida para este projeto foi o OpenNebula. A Tabela 4.2 apresenta características de algumas das soluções alternativas encontradas e estudadas.

	Licença	**Versão Gratuita**	***Clouds* Híbridas**	**Gestores de Virtualização**	**Virtualização assistida por *hardware***
VMware vSphere	Proprietária	Não	Não	VMware	Não necessário
RHEV	Proprietária	Não	Não	KVM	Obrigatório
XenServer	Proprietária	Não	Não	XEN	Apenas para ambientes Windows
HyperV	Proprietária	Não	Não	HyperV	Não
Eucalyptus	BSD	Sim	Parcialmente	Xen, KVM, Vmware	Só com a utilização de KVM
Nimbus	Apache 2	Sim	Parcialmente	Xen, KVM	Apenas para ambientes Windows
Abiquo	Proprietária	Não	Não	VMware, Xen, KVM, HyperV	Só com a utilização de KVM

Tabela 4.2 Soluções alternativas ao OpenNebula[71]

O OpenNebula é uma solução *open-source* utilizada para a virtualização de *datacenters*, que disponibiliza um conjunto diverso de ferramentas que permite e auxilia a criação de todo o tipo de *clouds*, sejam estas privadas, públicas ou híbridas, fornecendo sempre os recursos computacionais

[68] http://www.citrix.com/products/cloudplatform/overview.html
[69] http://www.vmware.com/products/datacenter-virtualization/vcloud-suite/overview.html
[70] http://www.microsoft.com/en-us/server-cloud/cloud-computing/default.aspx
[71] Tabela construída com base em (Rimal & Lumb, 2009) e (Heckel, 2010)

sobre a forma de IaaS. Este sistema caracteriza-se por ser maioritariamente utilizado na gestão de todo um *datacenter*, e por oferecer suporte para extender um ambiente privado a uma *cloud* híbrida, de forma a interligar infraestruturas locais e externas. Para isto, inclui funcionalidades para a integração, gestão, segurança e monitorização de toda a infraestrutura e recursos associados. O OpenNebula é o resultado de um projeto de investigação iniciado em 2005 por Ignacio M. Llorente e Rubén S. Montero. A sua primeira versão foi lançada em Março de 2008, estando atualmente na versão 4.0. Além de ser um projeto *open-source*, possui também uma versão comercial que, apesar de possuir as mesmas funcionalidades tecnológicas que a versão gratuita, possui algumas facilidades extra, como por exemplo suporte especializado, orientado especificamente para os seus clientes. Hoje em dia, o OpenNebula é utilizado e aprovado por grandes clientes como a RIM, Telefonica, IBM, SAP, DELL, KPMG, etc. Juntamente com estas referências, a opção pelo OpenNebula deveu-se, sobretudo, ao facto de:

- ser uma solução *open-source*;
- possibilitar a criação de *clouds* privadas;
- possuir boa documentação *online* para ajuda e suporte;
- permitir administrar a *cloud* através de uma interface gráfica;
- ter um repositório - *OpenNebula Marketplace* – no qual é possível descarregar imagens, por exemplo de máquinas virtuais, pré-configuradas e prontas a correr numa *cloud* criada a partir do OpenNebula.

A versão utilizada neste projeto foi a 3.6, com o código de referência Lagoon, já que se tratava da versão mais atual e estável no início da realização deste trabalho. Informações adicionais sobre o projeto OpenNebula podem ser encontradas na *web* em http://opennebula.org/about:about e http://opennebula.org/ about:why.

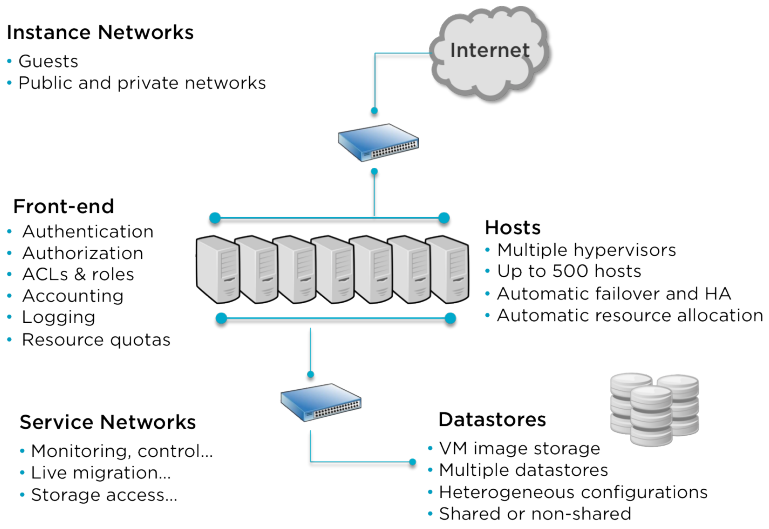


Figura 4.3 Características e serviços gerais do OpenNebula[72]

4.2.2 Instalação e configuração do sistema OpenNebula

O OpenNebula utiliza uma arquitetura clássica e típica de um *cluster* para instanciar uma *cloud*. Esta arquitetura caracteriza-se pela existência de um nodo *front-end* e um conjunto de *hosts* onde serão executadas as máquinas virtuais. No mínimo, existe ainda uma rede física que interliga todos os nodos da rede, ou seja, que permite o acesso do *front-end* a cada um dos *hosts*. No total, identificam-se pelo menos quatro componentes principais num sistema OpenNebula:

- O *front-end*, que é essencialmente uma máquina na qual está localizada a instalação do OpenNebula e que tem como missão gerir e administrar toda a infraestrutura que suporta a *cloud*. Para isso, necessita de ter acesso a todo o armazenamento, quer seja diretamente ou através de uma rede, e a todos os nodos da *cloud*.

- Os *hosts*, que são as máquinas físicas utilizadas para fornecer os recursos computacionais à *cloud* e que, na prática, executarão as máquinas virtuais instanciadas. O OpenNebula por si só não instala qualquer componente nestas máquinas. Porém, estas têm de ter uma configuração base que no mínimo possibilite a execução de ambientes virtualizados.

[72] Imagem extraída de (OpenNebula, 2013b)

– As *datastores*, que são meros repositórios que contêm as imagens dos discos das máquinas virtuais e que estão acessíveis diretamente pelos nodos ou através de uma SAN. Sempre que uma máquina virtual é instanciada, estas imagens são transferidas para os *hosts*.

– Por fim, a *Service Network*, que é uma rede que interliga os *hosts* ao *front-end* e que suporta a execução de serviços do OpenNebula entre estas duas partes.

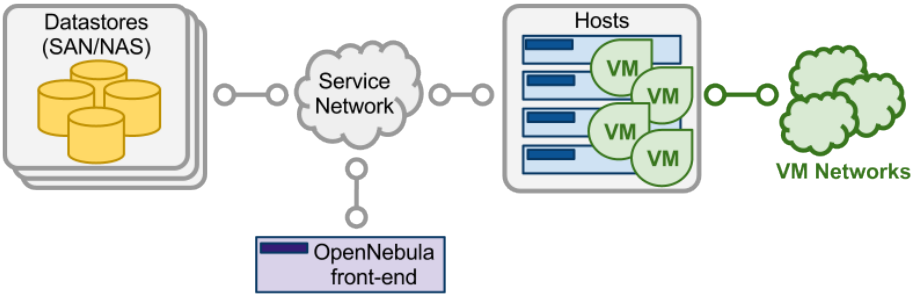


Figura 4.4 Arquitetura típica de um sistema OpenNebula[73]

É importante conhecer a arquitetura esperada e utilizada pelo OpenNebula pois, antes de procedermos à instalação do sistema, foi necessário preparar toda a infraestrutura necessária que iria suportar a *cloud*. Para o efeito, utilizamos uma máquina, designada por "Fratelo", com as seguintes características essenciais:

– CPU: Intel(R) Pentium(R) 4 CPU 3.20GHz.

– Memória RAM: 4.0GB SDRAM.

– Disco rígido: 500GB.

– Sistema Operativo: Ubuntu 12.10 – Quantal Quetzal, 64-bit.

– Endereço IP público: 193.136.19.78.

Esta máquina foi utilizada exclusivamente para o desenvolvimento e teste do sistema desenvolvido. Todavia, num ambiente de produção recomenda-se a utilização de mais máquinas para suportar a

[73] Imagem extraída de (OpenNebula, 2013a)

cloud, de forma a que a quantidade de requisitos computacionais disponíveis a partir da nuvem seja maior. Dada a utilização de uma só máquina para este projeto, na instalação do OpenNebula, esta foi utilizada como sendo simultaneamente o *front-end* e um *host* da *cloud*.

Após ter toda a infraestrutura física disponível, para iniciar o processo de instalação foi necessário atender e resolver algumas dependências que afetam um sistema OpenNebula. Assim, o primeiro passo foi dado com a instalação das librarias do Ruby e de outros pacotes requeridos por alguns dos seus componentes:

```
> sudo apt-get install libsqlite3-dev libxmlrpc-c3-dev g++ ruby libopenssl-ruby libssl-dev ruby-dev libxml2-dev libmysqlclient-dev libmysql++-dev libsqlite3-ruby libexpat1-dev rake rubygems libxml-parser-ruby1.8 libxslt1-dev genisoimage scons nokogiri rake xmlparser
```

A máquina *front-end* utiliza uma rede para poder aceder aos diversos *hosts* envolvidos, monitorizar e gerir os gestores de virtualização ou até para fazer a transferência de ficheiros entre os diversos nodos da rede. De forma a oferecer conectividade às máquinas virtuais instanciadas, interligou-se a sua interface de rede através de uma *bridge* ao respetivo *host*. Sendo assim, foi necessário criar a respetiva bridge na máquina do projeto. Para isso, instalou-se o pacote "bridge-utils", para definir uma bridge e colocou-se no ficheiro /etc/network/interfaces a respetiva configuração. Veja-se de seguida o seu conteúdo:

```
> sudo apt-get install bridge-utils
### conteúdo do ficheiro /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 193.136.19.78
netmask 255.255.255.0
network 193.136.19.0
broadcast 193.136.19.255
gateway 193.136.19.254
```

```
dns-nameservers 193.136.19.1
dns-search di.uminho.pt
dns-domain di.uminho.pt
```

Para o *back-end*, o OpenNebula permite que seja utilizada uma base de dados MySQL ou SQLite. Devido à familiaridade com MySQL, optamos pela instalação deste sistema de gestão de bases de dados. Para efeitos de testes, criamos uma configuração simples para acesso à base de dados. A saber:

```
> sudo apt-get install mysql-server


> mysql -u root -p
> CREATE USER 'oneadmin'@'localhost' IDENTIFIED BY 'oneadmin';
> CREATE DATABASE opennebula;
> GRANT ALL PRIVILEGES ON opennebula.* TO 'oneadmin' IDENTIFIED BY 'oneadmin';
```

Por omissão, o OpenNebula utiliza algumas pastas tradicionais de um sistema Unix, /usr, /etc e /var, para guardar alguns dos ficheiros resultantes do processo de instalação. De modo a se ter uma estrutura de pastas menos dispersa, criamos a diretoria /srv/cloud/ para alojar todos os ficheiros necessários. De seguida, procedemos à criação de um grupo de utilizadores e de um utilizador no sistema operativo. Esse utilizador ficou associado à instalação do OpenNebula, sendo o responsável pela sua administração. Criamos então o utilizador "oneadmin" e definimos a sua pasta *home* como sendo a diretoria: /srv/cloud/one.

```
> groupadd cloud
> useradd -d /srv/cloud/one -g cloud -m oneadmin
> sudo passwd oneadmin
```

Até ao passo anterior fizemos as modificações necessárias para se poder instalar o OpenNebula. Na altura, apesar de existir uma versão do OpenNebula nos repositórios do Ubuntu, esta não era a mais atual. Como tal, transferimos manualmente do sítio do OpenNebula a versão mais atual e

estável até à altura - OpenNebula 3.6 – Lagoon- e executamos o seu instalador, associando o utilizador, grupo e pasta "home" que foram criados no passo anterior.

```
> ./install.sh -u oneadmin -g cloud -d /srv/cloud/one
```

Com o processo de instalação terminado, apenas foi necessário realizar algumas configurações essenciais para o funcionamento do OpenNebula. Primeiro, na pasta de instalação, foi necessário criar um ficheiro com as credenciais criadas para o utilizador que iria gerir a *cloud* e a sua infraestrutura. Neste caso, criamos um ficheiro designado por "one_auth" na diretoria /srv/cloud/one/.one/, com o seguinte conteúdo "oneadmin:oneadmin" (utilizador:password). Após inicialização, o OpenNebula e alguns dos seus componentes precisam de ler este ficheiro para terem um funcionamento correto. Segundo, relembrando o facto de termos modificado a estrutura de pastas utilizada por omissão pelo OpenNebula, foi necessário definir o valor das variáveis de ambiente utilizadas nos diversos serviços inerentes. Para isso, modificamos o ficheiro ".profile" na pasta home do utilizador "oneadmin" (conteúdo abaixo) e carregamos as novas variáveis no sistema:

```
> Export ONE_LOCATION=/srv/cloud/one
> Export ONE_AUTH=$ONE_LOCATION/.one/one_auth
> Export ONE_XMLRPC=http://localhost:2633/RPC2
> Export PATH=$ONE_LOCATION/bin:/usr/local/bin:/var/lib/gems/1.8/bin/:$PATH
```

Como o processador da máquina utilizada não suportava virtualização, não foi possível utilizar o *software* kvm como gestor de virtualização. Por omissão, o OpenNebula está configurado para utilizar esse produto. Sendo assim, a opção recaiu sobre xen tendo sido necessário instalá-lo e ativá-lo na máquina referida. De seguida foi necessário indicar ao OpenNebula que seria utilizado o xen na instanciação das máquinas. Para isso, acedi ao ficheiro /srv/cloud/one/etc/oned.conf para colocar o seguinte conteúdo:

```
> IM_MAD = [
      name       = "im_xen",
      executable = "one_im_ssh",
```

```
    arguments  = "xen" ]

VM_MAD = [
    name       = "vmm_xen",
    executable = "one_vmm_sh",
    arguments  = "xen",
    default    = "vmm_sh/vmm_sh_xen.conf",
    type       = "xen" ]
```

Por fim, para conseguir inicializar o OpenNebula, foi necessário indicar os dados de acesso à base de dados MySQL criada. Acedemos novamente ao ficheiro /srv/cloud/one/etc/oned.conf para comentar a utilização de SQLite e atualizar os dados para a conexão ao MySQL.

```
> #DB = [ backend = "sqlite" ]
DB = [ backend = "mysql",
       server = "localhost",
       port = 0,
       user = "oneadmin",
       passwd = "oneadmin",
       db_name = "opennebula" ]
```

Realizados todos os passos acima, já era possível iniciar e utilizar o OpenNebula. Contudo, como pretendíamos utilizar uma interface gráfica – Sunstone - para administrar a *cloud*, realizamos mais alguns passos com vista a instalar esse componente. Assim, resultante da instalação do OpenNebula, executamos um script que permite instalar o Sunstone e, de seguida, instalamos o cliente noVNC para conseguir estabelecer ligações VNC com as máquinas virtuais instanciadas:

```
> sudo /srv/cloud/one/share/install_gems sunstone
> sudo /srv/cloud/one/share/install_novnc.sh
```

Os passos descritos nesta secção permitiram-nos obter uma instalação base e suficiente do OpenNebula para conseguir dar "forma" à *cloud* privada pretendida. Como se trata de uma solução

versátil e poderosa, o OpenNebula apresenta uma panóplia de configurações e personalizações avançadas que podem ser consultadas diretamente na documentação disponível no seu sítio online - http://opennebula.org/documentation:archives:rel3.6.

4.2.3 Administração geral da *cloud*

O OpenNebula permite administrar e gerir todos os recursos de uma *cloud* através da linha de comandos (CLI – *command-line interface*) ou através de uma interface Web (*Sunstone*). Por ser uma interface gráfica e por permitir uma gestão simplificada da *cloud*, optamos pela utilização do OpenNebula Sunstone. Basicamente, este é um painel de controlo que permite a gestão simplificada de infraestruturas privadas e híbridas numa *cloud*, que facilita a gestão dos recursos físicos e virtuais através de uma interface gráfica, bastante apelativa, da mesma forma que seria possível fazer através da linha de comandos. Pode ainda ser configurado e adaptado para diferentes perfis de utilização, restringindo assim opções de acordo com as permissões de cada utilizador. O painel de controlo do OpenNebula Sunstone é acedido através de um *browser*, por omissão no endereço localhost:9869, podendo os dados de acesso serem consultados no ficheiro sunstone-server.conf. Este ficheiro resulta da instalação do Sunstone. Na instalação realizada ficou localizado em /srv/cloud/one/etc/sunstone-server.conf.

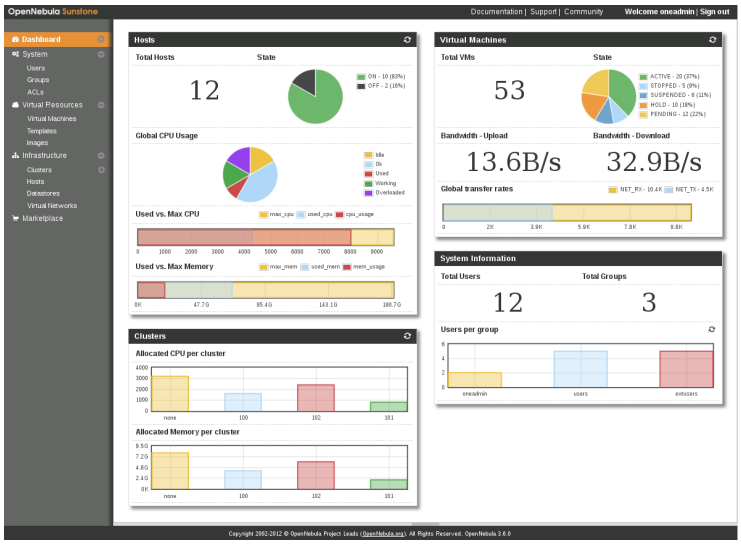


Figura 4.5 A interface do OpenNebula Sunstone

Logo após a instalação do OpenNebula e do Sunstone, foi então possível começar a instanciar as máquinas virtuais pretendidas. Para isso, tivemos de indicar quais os *hosts* que iriam suportar a

cloud, criar as respectivas redes, *datastores*, imagens e os *templates* que descrevem a configuração das máquinas virtuais. No Sunstone começamos assim por indicar que a própria máquina do sistema - a "Fratelo" - iria ser também um *host* para a *cloud*. Posteriormente, qualquer *host* que se pretendesse adicionar teria que estar acessível pela máquina *front-end*, através de uma rede. Para adicionar um *host*, bastou apenas indicar o seu nome ou endereço IP e escolher o respectivo gestor de virtualização que se pretendia utilizar.

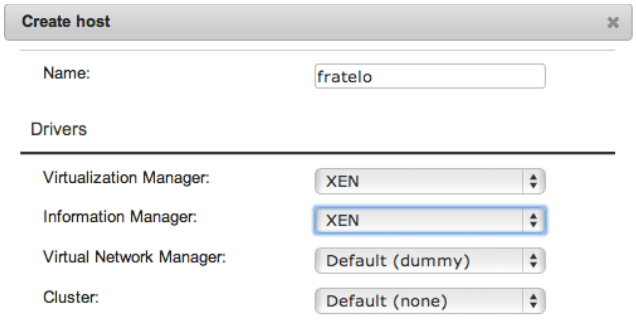


Figura 4.6 Definição da associação de um *host* à *cloud*

Se tudo correr como o previsto, em poucos segundos o *host* é associado e o seu estado definido como ligado – ON. Se ocorrer algum erro, é possível consultar o ficheiro "oned.log" para se obter mais detalhes sobre a operação.

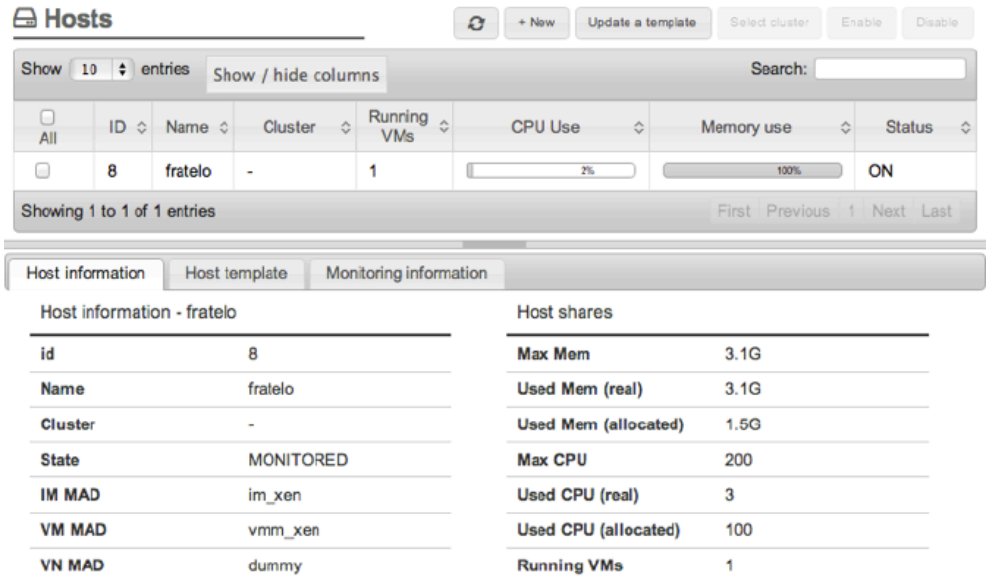


Figura 4.7 Uma listagem dos *hosts* definidos sobre a *cloud*

Para se criar a rede virtual privada que suporta as máquinas virtuais foi necessário associar uma *bridge* e definir um conjunto de endereços IP disponíveis para utilização. A *bridge* associada tem a designação de "virbr0", tendo sido criada para o efeito aquando da preparação da instalação do OpenNebula. Na Figura 4.8 apresenta-se a configuração que foi escolhida para a rede no momento da sua criação.

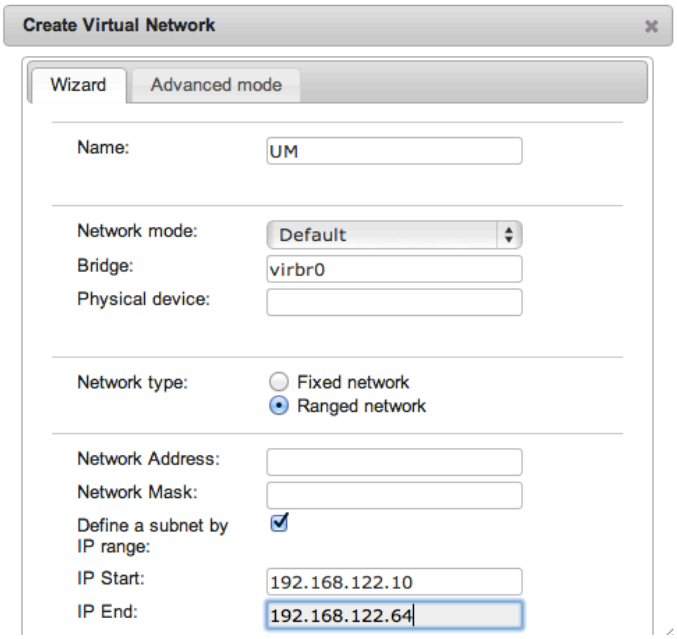


Figura 4.8 Criação de uma rede virtual privada

As *datastores* são repositórios utilizados para armazenar os discos das máquinas virtuais instanciadas. Como vimos na Figura 4.4, as *datastores* podem estar localizadas em máquinas distintas dos hosts, estando ligadas a estes por intermédio de uma rede de serviço. Neste caso, ao iniciar uma máquina virtual, o seu disco correspondente é transferido de uma *datastore* para o *host* que suporta a sua execução. Neste projeto, por falta de infraestrutura, optamos pela criação de uma única *datastore* localizada no próprio host, não havendo assim necessidade de realizar esta transferência pois havia acesso direto aos discos armazenados. Por este motivo, as opções apresentadas no menu de criação de uma *datastore* eram pouco significativas, pelo que foram utilizadas as definidas por omissão.

Figura 4.9 Criação de uma *datastore*

Finalizada a criação de uma *datastore*, obtemos então um repositório ao qual é possível associar as imagens utilizadas nas máquinas virtuais. Estas imagens podem representar sistemas operativos ou simplesmente dados, e podem ser utilizadas simultaneamente por várias máquinas virtuais. Neste projeto procedemos à criação de duas imagens: uma representativa de um sistema de ficheiros e outra que continha a imagem de instalação do Ubuntu 12.04, que foi o sistema operativo escolhido para instalar na máquina virtual que se pretendia instanciar. Neste segundo caso, apenas foi necessário indicar um URL no qual se poderia fazer a transferência da imagem. De seguida, o OpenNebula tratou desse processo automaticamente. Os parâmetros definidos para cada uma das imagens têm em consideração o gestor de virtualização escolhido (xen).

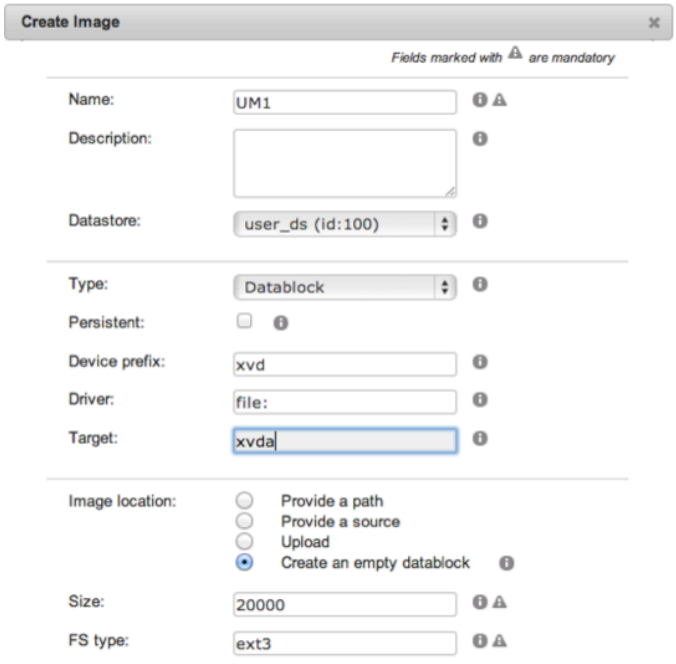


Figura 4.10 Definição da imagem representativa do sistema de ficheiros

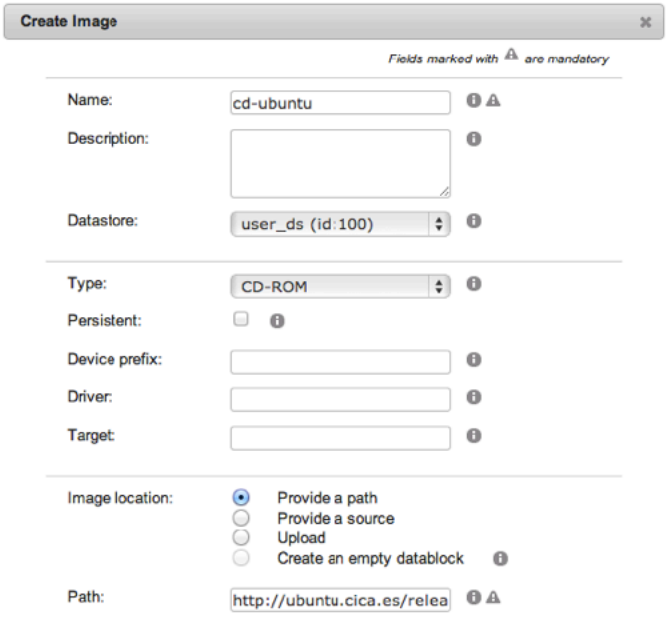


Figura 4.11 Criação da imagem representativa do CD de instalação

Normalmente, quando se monta uma infraestrutura deste tipo, é comum proceder-se à instanciação de múltiplas máquinas virtuais com as mesmas características e recursos. Para facilitar este processo, o OpenNebula possibilita a criação de *templates* que definem toda a configuração necessária para instanciar uma dada máquina virtual. Existem diversas opções que se podem especificar num *template*, umas mais genéricas, como a quantidade de recursos computacionais a utilizar em termos de memória ou capacidade de processamento, e outras mais específicas, que são disponibilizadas consoante o gestor de virtualização escolhido. É também no *template* que associamos as duas imagens criadas e que seriam utilizadas no arranque da máquina virtual, bem como a rede virtual privada também anteriormente criada. A Figura 4.12 apresenta uma visão detalhada do *template* criado para este projeto.

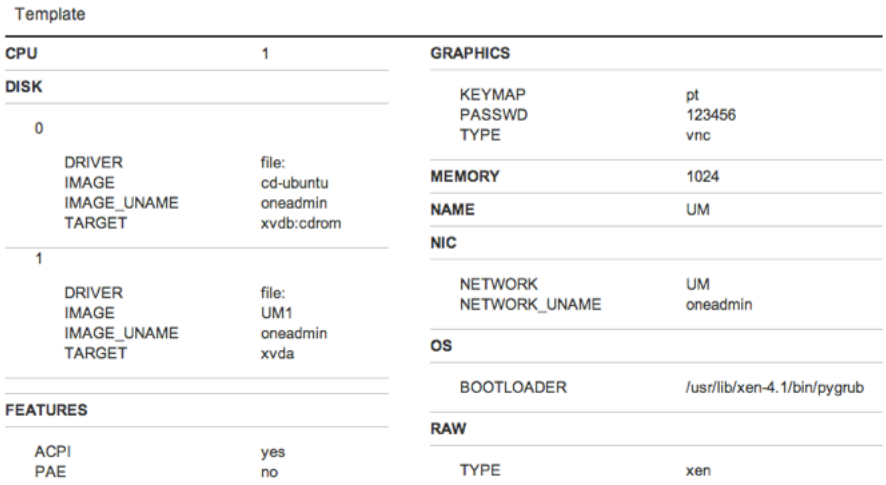


Figura 4.12 Especificação do *template* para a instanciação de máquinas virtuais

Após a criação do *template*, basta selecioná-lo e instanciar uma máquina virtual a partir do mesmo. Se tudo correr como previsto, após alguns segundos, a máquina virtual estará a ser executada normalmente. Se ocorrer algum erro, a partir do Sunstone pode ser consultado o log de operações relacionadas com a máquina virtual em causa. Após a máquina virtual estar a correr, é possível visualizar alguma informação sobre si, como por exemplo o endereço IP que lhe foi atribuído (Figura 4.13).

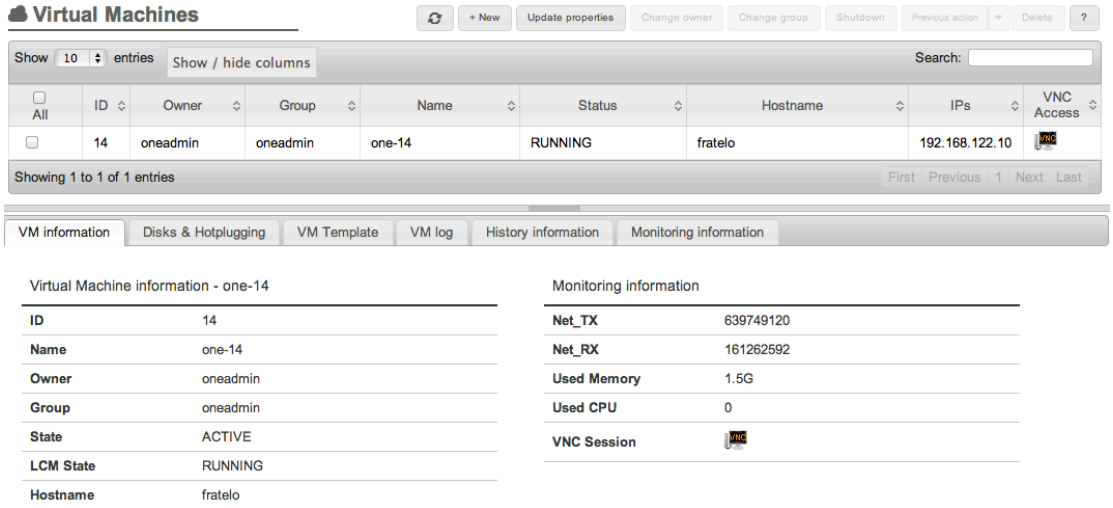


Figura 4.13 Detalhes da máquina virtual instanciada

4.2.4 O papel da *cloud* no sistema AC2DC

Como já foi dito anteriormente na descrição do funcionamento geral do sistema AC2DC (secção 4.1.3), a *cloud* seria criada para suportar um SDW que, por sua vez, providenciaria os dados de negócio necessários para alimentar os painéis de exploração analítica de dados – os *dashboards*. Assim como foi dado foco ao longo desta dissertação, pretendia-se também que o DW do sistema ficasse assente num sistema de gestão de bases de dados relacional. Para este efeito, através de uma conexão VNC, instalamos um sistema PostgreSQL na máquina virtual que instanciamos através dos passos descritos na secção 4.2.3. Nele, para efeitos de testes neste projeto, optamos pela utilização da base de dados "Foodmart" como fonte de "inspiração" e de alimentação para o DW que seria implantado na c*loud*. A "Foodmart" é um exemplo de uma base de dados que é utilizada com muita frequência para suporte a processos de aprendizagem na área dos sistemas de dados. Além disso, o facto de termos alguma familiaridade com o seu esquema e dados, tornaram a sua utilização mais fácil bem como a sua adaptação ao caso particular do sistema de dados que foi desenvolvido para o sistema AC2DC. A título de curiosidade, o esquema da "Foodmart" retrata um cenário tradicional de venda de produtos alimentares duma empresa fictícia designada por "Foodmart". Entre outros, esta base de dados contém informação sobre clientes, produtos, empregados, armazéns e, claro, vendas. No âmbito deste projeto, a análise detalhada deste esquema não era um objetivo, até porque apenas foram utilizadas algumas das suas dimensões na exploração dos dados – apenas as consideradas suficientes para trabalhar sobre um cubo OLAP de vendas, referentes ao ano de 1997, sendo que as restantes não foram enviadas para a *cloud*, permanecendo apenas nas fontes de informação, ou seja, nas instâncias do DW corporativo. Em todo o caso, a figura abaixo apresenta um esquema simplificado para se poder ter uma noção geral do cenário aplicacional retratado.

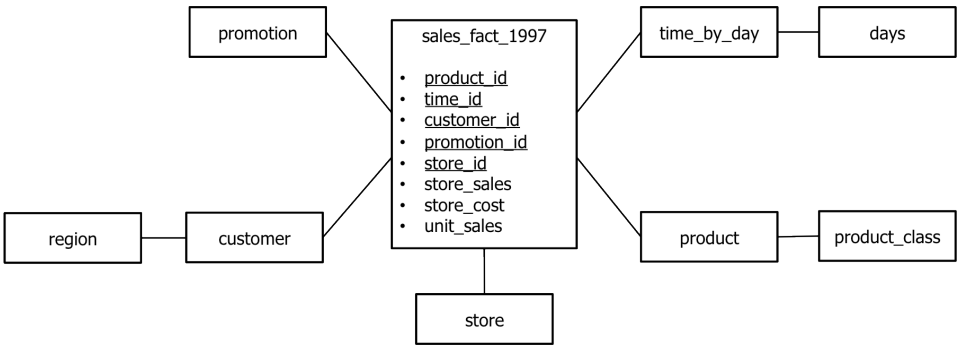


Figura 4.14 Um esquema simplificado referente a uma parte do *Data Mart* de vendas mantido na *cloud*

Como seria de esperar, no início apenas se utilizou uma máquina virtual para suportar todo DW, tendo esta sido instanciada com os recursos necessários para executar este sistema de acordo com o volume de dados armazenado e os requisitos do protótipo desenvolvido. Claro está que, à medida que os requisitos fossem mudando, quer fosse em termos de espaço de armazenamento, disponibilidade ou desempenho, seria necessário ajustar os recursos disponibilizados para o funcionamento do DW. Porém, como o foco do projeto assentava na auto-adaptabilidade e reestruturação automática de painéis de *dashboarding*, as propriedades como a escalabilidade e elasticidade da *cloud* foram pouco exploradas. Contudo, devido aos diversos serviços e possibilidades de gestão de infraestrutura que apresenta, o OpenNebula é uma solução capaz de lidar com este tipo de problemas e alcançar assim as propriedades desejadas. Para isto, o primeiro passo passaria então pela criação de um *cluster* de base de dados, optando por uma arquitetura SN como foi considerada, nesta dissertação, como sendo a mais adequada para a implantação de DWs num ambiente em *cloud*, e, de seguida, pela utilização de diversas funcionalidades do OpenNebula de modo a atingir-se o desempenho esperado para o sistema. Por exemplo, com estas poder-se-ia replicar ou particionar os dados do DW armazenado no disco da máquina virtual, realizando assim um dimensionamento horizontal do sistema, ou, em alternativa, redimensionar essa mesma máquina virtual de forma a aumentar os seus recursos computacionais, podendo para isto ser necessário recorrer à utilização das funcionalidades de migração existentes, com o objetivo de mover a instância virtual para um *host* mais poderoso.

4.3 O ciclo desde a angariação ao provisionamento dos dados

Tal como foi referido na apresentação geral do projeto AC2DC, os diversos processos existentes que permitem fluir os dados por todo o sistema, desde as fontes de informação até aos painéis de exploração, são suportados por um conjunto de agentes computacionais. Cada um destes agentes é autónomo mas, como está associado a algum componente do sistema, tem o seu ciclo de vida bem definido, sendo regulado justamente consoante o início e o fim do componente respetivo. Contudo, existem situações em que o seu término não é previsível, como é o caso dos agentes conciliador e provedor que, estando associados à *cloud*, não têm um fim agendado, uma vez que este ambiente tem de estar sempre disponível e em constante funcionamento. Apesar da sua autonomia e independência, o comportamento de cada agente é gerido por um conjunto de instruções previamente definidas e guardadas na sua agenda de trabalho.

A montante, o ciclo de vida do sistema AC2DC inicia-se com a ativação de um ou mais agentes angariadores, que começarão a recolher a informação que pretendem a partir das fontes de dados à sua disposição com base naquilo que têm definido nas suas agendas de trabalho. Ao mesmo tempo, estes agentes cooperam e realizam algumas tarefas juntamente com um ou mais agentes conciliadores. De seguida, estes últimos prepararão os dados recolhidos, enviando-os posteriormente para a *cloud*. No lado jusante do sistema, encontram-se os provedores como sendo os consumidores diretos dos dados da *cloud*. Estes coletam toda a informação necessária para satisfazer as interrogações multidimensionais lançadas pelos agentes de decisão.

4.3.1 Os processos de angariação

Os agentes angariadores são os responsáveis por iniciar o processo que permite que os dados necessários à criação e apresentação dos painéis de exploração estejam disponíveis na *cloud*. Para tal, cada um destes agentes está associado a uma fonte de dados específica, que lhe foi atribuída no início da sua atividade e da qual irá obter a informação que posteriormente será enviada para o conciliador. A fonte de informação permanece associada ao agente angariador até ao momento em que este termine as suas ações. Durante a sua inicialização, o agente angariador recebe ainda a identificação do conciliador com quem irá comunicar e um ficheiro de configuração que contém

informações cruciais para o seu funcionamento. Nele estão presentes a localização da sua agenda de trabalho bem como as credenciais necessárias para acesso à fonte de dados.

```
<tasks>
  <task>
    <queries>
      <query tableName="account_dsa"> SELECT * FROM account </query>
      <query tableName="agg_c_special_sales_fact_1997_dsa"> SELECT * FROM agg_c_special_sales_fact_1997 </query>
      <query tableName="agg_lc_100_sales_fact_1997_dsa"> SELECT * FROM agg_lc_100_sales_fact_1997 </query>
      <query tableName="category_dsa"> SELECT * FROM category </query>
      <query tableName="currency_dsa"> SELECT * FROM currency </query>
      <query tableName="customer_dsa"> SELECT * FROM customer </query>
      <query tableName="days_dsa"> SELECT * FROM days </query>
      <query tableName="department_dsa"> SELECT * FROM department </query>
      <query tableName="position_dsa"> SELECT * FROM position </query>
      <query tableName="product_dsa"> SELECT * FROM product </query>
      <query tableName="product_class_dsa"> SELECT * FROM product_class </query>
      <query tableName="promotion_dsa"> SELECT * FROM promotion </query>
      <query tableName="region_dsa"> SELECT * FROM region </query>
      <query tableName="reserve_employee_dsa"> SELECT * FROM reserve_employee </query>
      <query tableName="sales_fact_1997_dsa"> SELECT * FROM sales_fact_1997 </query>
      <query tableName="store_dsa"> SELECT * FROM store </query>
      <query tableName="store_ragged_dsa"> SELECT * FROM store_ragged </query>
      <query tableName="time_by_day_dsa"> SELECT * FROM time_by_day </query>
      <query tableName="warehouse_dsa"> SELECT * FROM warehouse </query>
      <query tableName="warehouse_class_dsa"> SELECT * FROM warehouse_class </query>
    </queries>
    <periodicity> 500000 </periodicity>
  </task>
</tasks>
```

Figura 4.15 Exemplo da agenda de trabalho de um agente angariador

A agenda de trabalho de um angariador contém a lista de interrogações, que serão realizadas sobre a fonte de dados associada, e uma periodicidade, que representa a frequência, em milissegundos, com que a tarefa de extração será executada. Em cada interrogação está definido um atributo denominado por "tableName", colocado para que o conciliador consiga saber que dados está a receber. Este atributo é especialmente importante num caso em que uma interrogação apresente uma junção sobre várias tabelas, no qual não é possível utilizar um nome de uma tabela fonte. Existe ainda uma outra pequena particularidade em relação às interrogações, uma vez que cada uma destas deve possuir uma cláusula "*WHERE*". Nessa cláusula uma das parcelas é composta apenas por um *wildcard* "?". Todavia, pode possuir tantas parcelas quanto aquelas que forem necessárias, desde que sejam válidas em SQL. Aquando da extração dos dados da fonte, o *wildcard* referido será substituído por um intervalo de tempo que permite filtrar os dados que já foram extraídos previamente, de modo a não reenviar dados repetidos. Para que seja possível essa filtragem dos dados, é necessário que cada registo possua uma etiqueta temporal que indique a data da sua criação e, ainda, que exista uma tabela auxiliar na qual esteja registado

para cada tabela a extrair, qual a data de criação do último registo recolhido. Deste modo, o intervalo de tempo em questão é facilmente definido entre a data indicada na tabela auxiliar referida e a data atual da tarefa de extração, sendo que, no final da transferência de dados, é necessário atualizar a tabela auxiliar. De referir ainda que as transferências de dados para o conciliador são realizadas à medida que os dados são extraídos da fonte. Os dados são extraídos em segmentos pré-definidos - por exemplo de 10.000 (dez mil) registos - de forma a evitar transferências muito demoradas. Após extração, e antes de serem enviados para o conciliador, os dados são convertidos para XML com recurso a um *codec* fornecido pela plataforma de agentes JADE. A extração é concluída assim que os registos resultantes das interrogações definidas na agenda forem enviados com sucesso.

	id integer	tableName character varying	last_transfer_date timestamp without time zone
1	353	agg_lc_100_sales_fact_1997_dsa	2013-04-14 21:06:11
2	354	category_dsa	2013-04-14 21:06:11
3	355	currency_dsa	2013-04-14 21:06:11
4	356	customer_dsa	2013-04-14 21:06:11
5	357	days_dsa	2013-04-14 21:06:11
6	358	department_dsa	2013-04-14 21:06:11
7	359	position_dsa	2013-04-14 21:06:11
8	360	product_dsa	2013-04-14 21:06:11
9	361	product_class_dsa	2013-04-14 21:06:11
10	362	promotion_dsa	2013-04-14 21:06:11
11	363	region_dsa	2013-04-14 21:06:11
12	364	reserve_employee_dsa	2013-04-14 21:06:11
13	365	sales_fact_1997_dsa	2013-04-14 21:06:11

Figura 4.16 *Snapshot* de uma tabela de controlo auxiliar ao processo de angariação de dados

4.3.2 A conciliação do *Data Warehouse* corporativo na *cloud*

Os agentes conciliadores trabalham diretamente com a *cloud*, alimentando o DW que nela está implantado com os dados provenientes dos angariadores. Posto isto, como são os angariadores que iniciam os processos de comunicação com os conciliadores, estes últimos não precisam de conhecer outros agentes existentes para realizarem uma dada tarefa. Contudo, no momento da sua inicialização, têm de se identificar no sistema como provedores de serviço de operações de conciliação de dados, para que os angariadores consigam assim ter acesso a agentes habilitados e disponíveis para receber os seus dados provenientes das fontes de informação. Ainda durante o seu processo de inicialização, os conciliadores recebem também um ficheiro de configuração que indica a localização da sua agenda de trabalho, as credenciais necessárias para acesso à *cloud* e

para acesso à sua base de dados local. Esta base local funciona como uma área de estágio utilizada para suportar habituais processos de transformação sobre os dados, como limpeza, uniformização ou conciliação de registos.

Sempre que uma comunicação é estabelecida, o conciliador começa por converter os dados recebidos do formato XML para estruturas em JAVA, de forma a conseguir manipular os mesmos, guardando-os de seguida nas tabelas temporárias existentes na área de estágio, numa zona de pré-processamento. Toda a informação necessária para este armazenamento encontra-se descrita na sua agenda de trabalho do conciliador. Esta está dividida em duas grandes partes: a primeira é constituída por interrogações SQL que representam todo o processo de conciliação, desde inserções em tabelas temporárias, transformações sobre os dados, etc.; e a segunda parte representa o processo de transferência dos dados já conciliados para a *cloud*. Na agenda, as tarefas são executadas pela ordem que estão definidas no ficheiro, sendo que nela pode estar descrita qualquer ação passível de ser representada em SQL, como junção de tabelas, limpezas de dados, eliminação de valores repetidos, etc.. Somente quando todos os dados estiverem processados é que se inicia a sua transferência para a *cloud*, minimizando-se assim o tempo de interação com este ambiente. Para isto, na agenda estão descritas as instruções de seleção que serão executadas sobre uma zona de pós-processamento onde estão os dados preparados. As instruções SQL de inserção na *cloud* são geradas no momento, sendo que, para isso, a agenda do conciliador associa, para cada instrução de seleção, o nome da tabela de destino da *cloud*. De referir ainda que não é necessário o uso de tabelas de controlo no processo de transferência pois, à medida que os registos são inseridos na *cloud*, são removidos da área de estágio.

4.3.3 A provisão e o refrescamento das estruturas multidimensionais

Os agentes provedores são os únicos consumidores diretos da informação presente na *cloud*. Estes têm um comportamento semelhante ao dos agentes angariadores, uma vez que recolhem os dados de uma fonte, neste caso a *cloud* privada. Porém, em vez de os enviarem para um outro agente, guardam-nos localmente, garantindo assim o seu acesso mesmo em caso de falha da *cloud*. Um provedor começa, assim, por obter apenas os dados que necessita, podendo ou não representar a totalidade dos dados presentes na *cloud*, e, de seguida, armazena-os em estruturas multidimensionais de dados – cubos OLAP – geridas pelo servidor analítico Mondrian.

Para o controlo das transferências de dados, os provedores também utilizam agendas de trabalho compostas por interrogações que definem os dados a coletar, etiquetas temporais e uma tabela auxiliar que identifica os últimos registos transferidos de cada tabela. Além disto, o seu ficheiro de configuração contém toda a informação pertinente para o seu funcionamento, como os acessos à *cloud*, as credenciais para a sua base de dados local, a localização da sua agenda e a localização de um esquema, escrito em XML, do cubo de dados a utilizar. Para que o motor OLAP – Mondrian – seja capaz de responder às interrogações MDX (MultiDimensional eXpressions) que lhe forem colocadas provenientes dos painéis de exploração, este serve-se do tal esquema XML que contém os metadados, tabela de factos, dimensões, métricas e respetivas hierarquias essenciais para a criação dos cubos.

```
<agentConfiguration type="provider">
  <agenda> /home/ac2dc/agents/agendas/providerAgenda.xml </agenda>
  <mondrianCatalog> /home/ac2dc/agents/foodmart/foodmart.xml </mondrianCatalog>
  <sourceDbConfiguration dbms="postgres">
    <username> ac2dc </username>
    <password> ac2dc </password>
    <url> //fratelo.di.uminho.pt:5432/foodmart_dw </url>
  </sourceDbConfiguration>
  <targetDbConfiguration dbms="postgres">
    <username> ac2dc </username>
    <password> ac2dc </password>
    <url> //localhost/foodmart_mondrian </url>
  </targetDbConfiguration>
</agentConfiguration>
```

Figura 4.17 Exemplo de um ficheiro de configuração de um agente provedor

De referir ainda que a agenda de trabalho dos provedores indica a frequência com que estes agentes irão questionar a *cloud* para determinar a existência de novos dados, de forma a que estes possuam sempre informação atualizada. Quanto maior for a frequência de verificação maior será o nível da atualização dos dados locais dos provedores. Se necessário, pode-se mesmo chegar ao extremo de manter uma atualização constante, em tempo real, garantindo assim que os dados estão o mais atualizados possível.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Notas finais teóricas

Ao longo desta dissertação tentamos estabelecer uma ligação entre as diversas questões existentes inerentes à implementação de SDWs com as novas possibilidades e benefícios oferecidos por este novo e aliciante ambiente de computação que é a *cloud*. Esta dissertação revela-se assim como um ponto de situação que alerta para as vantagens deste ambiente e que é útil para a análise e revisão da problemática resultante da utilização da *cloud*, com vista à gestão de dados em geral e, em particular, para a implantação de SDWs.

Hoje em dia, a *cloud* é uma palavra de ordem no sector das TI. O conceito subjacente, *cloud computing*, envolve ter acesso, em qualquer parte do mundo, a qualquer momento, aos recursos computacionais e serviços necessários para executar processos de negócio ditos convencionais. Não se trata só de questões de infraestrutura física, *hardware*, mas também da possibilidade das organizações se abstraírem de questões relacionadas com redes, armazenamento, bases de dados, segurança e poder computacional. Como sabemos, num panorama tradicional, muitos dos problemas, requisitos técnicos e até mesmo os custos envolvidos são factores inibidores que impossibilitam a implementação de SDWs ou que levam a uma instalação e utilização imperfeita e incapaz de corresponder aos pressupostos que levaram à escolha deste tipo de sistemas. Para uma organização, o insucesso ou o sucesso infrutífero deste tipo de projetos trás enormes prejuízos a nível económico e claras limitações em termos de organização e planeamento estratégico do seu

negócio. Com a chegada da *cloud*, porque é que continuamos a adquirir os recursos que precisamos e, na incerteza, aqueles que podemos vir a precisar quando existe a possibilidade de alugar, a pedido e consoante as necessidades do momento, toda a infraestrutura de suporte necessária? Porque é que continuam a aparecer tantas questões relacionadas com *hardware e software* a fazer parte dos principais encargos e preocupações em projetos de *data warehousing*, quando no final do dia o que realmente importa são os dados? Graças às vantagens derivadas da utilização da *cloud*, consegue-se facilitar o acesso a este tipo de sistemas e, ao mesmo tempo, minimizar o risco presente na sua implementação.

A *cloud* é hoje uma realidade e uma mais valia para todos os *stakeholders* associados, independentemente de serem provedores ou consumidores. Contudo, apesar de não ser um conceito recente (secção 2.1), a sua aplicação e concretização são. Não se tratando, pois, de uma solução perfeita, há riscos que têm de ser controlados e ponderados antes da sua adoção. Como foi exposto nesta dissertação, no âmbito dos DWs foram surgindo diversos aspetos não funcionais (Timmermans et al., 2010) e tecnológicos que têm dificultado a sua aceitação neste novo ambiente. Entre outros, a perda de controlo resultante da mudança para a *cloud* é o que mais preocupa os possíveis utilizadores, pois esta leva à origem de preocupações relacionadas com a privacidade e segurança dos dados, ou até relacionadas com questões legais (secção 3.4.2). Isto acontece porque a *cloud* é um ambiente imprevisível e diferente do ambiente tradicional de um DW, no qual, como foi referido, toda a infraestrutura está mantida localmente dentro de uma dada organização e é dedicada exclusivamente à execução do sistema. O problema é que os DWs são sistemas bastante exigentes que fazem com que a fasquia suba em todos os sentidos (desempenho, robustez, elasticidade, etc.) quando comparados com outro tipo de aplicações e sistemas. Por exemplo, muitos dizem que a concretização da *cloud* só foi possível graças a diversos avanços ocorridos no mundo das TI, como é o caso dos verificados no domínio da banda larga. Mesmo assim, como a *cloud* pode ser um ambiente instável e imprevisível, será difícil de garantir que as transmissões de enormes volumes de dados – por vezes na ordem dos *terabytes* - entre grandes distâncias geográficas sejam eficientes, não sendo possível assegurar o desempenho dos SDWs mesmo perante as capacidades tecnológicas existentes hoje em dia. Porém, a verdade é que, mesmo face a tais incertezas, aliada aos custos reduzidos de utilização, a capacidade de provisionar poder computacional sob a forma de um serviço é um aspeto muito aliciante que impõe uma mudança nas TI direcionada para as *clouds*. Assim como qualquer nova solução que aparece, é normal que existam problemas numa primeira instância e, como tal, é também expectável que

venham a ser resolvidos num futuro próximo. Inclusive, foi com naturalidade que começaram a surgir soluções de *data warehousing* na *cloud*, algumas delas optando pela utilização de *clouds* privadas para que fosse possível minimizar alguns dos problemas e desafios identificados na implantação deste tipo de sistemas neste ambiente. À medida que os serviços de *cloud computing* forem amadurecendo, os DWs serão uns dos principais favorecidos.

5.2 Apreciação prática e trabalho futuro

O projeto prático apresentado nesta dissertação é referente ao desenvolvimento de um sistema de *Dashboards* auto-adaptáveis, que reflete sempre que possível as preferências de utilização dos agentes de decisão. Devido à sua elasticidade e à alegada disponibilidade infinita de recursos que consegue disponibilizar, a *cloud* pareceu ser um ambiente ideal para utilizar neste projeto para sustentar todo este sistema de suporte à decisão. Assim, tratando-se de um contexto de *data warehousing*, e dada a necessidade de acesso exclusivo ou restrito aos dados, projetou-se e implementou-se uma *cloud* privada, concebida como um repositório global de dados capaz de acolher informação proveniente de várias instâncias de um DW corporativo que, como frequentemente acontece, estão naturalmente distribuídas pelas instalações de uma empresa. Com a utilização deste ambiente computacional, o objetivo era o de garantir a alocação adequada de recursos em casos em que o sistema escalasse de forma acentuada. O principal serviço disponibilizado por esta *cloud* consistia em alimentar estruturas multidimensionais de dados – cubos OLAP –, cujo papel era satisfazer as interrogações de dados realizadas pelos agentes de decisão, disponibilizando a informação requerida para constituir os *dashboards*. Como é fácil de perceber, este é um sistema no qual são executados diversos processos de migração de dados, com o propósito de fluir toda a informação pelos diversos componentes do sistema, começando nas fontes de informação, conciliação na *cloud*, passando pelas estruturas multidimensionais e culminando nos painéis de exploração. Devido aos enormes volumes de dados envolvidos neste tipo de sistemas, seguindo também a mesma linha de raciocínio que introduziu a *cloud* neste projeto, de forma a garantir a escalabilidade do sistema em qualquer situação optamos pela utilização de uma comunidade de agentes computacionais distribuídos, cada um capaz de executar e suportar, autonomamente, os diversos processos de dados existentes, sendo o seu número variável consoante a necessidade e requisitos existentes no momento. Dado o tema desta dissertação, foi dado foco aos processos e respetivos agentes de angariação e conciliação na *cloud*

dos dados provenientes das fontes de informação, e aos agentes de provisão dos dados nas estruturas multidimensionais.

A realização do projeto correu bem, sem grandes sobressaltos que se destaquem ou problemas de maior influência que tenham prejudicado a sua planificação e calendarização iniciais. Claro que como o tema desta dissertação reúne dois assuntos distintos, *data warehousing* e *cloud computing*, num dos quais, atendendo ao meu percurso académico, tinha pouca experiência e conhecimento, houve alguns problemas que naturalmente foram aparecendo e que se revelaram como desafios e obstáculos a suplantar. Tendo feito especializações a nível de engenharia de software e sistemas de suporte a decisão, era normal que no início não estivesse preparado, a nível teórico ou técnico, para enfrentar a problemática que envolve a *cloud*, por esta abranger temáticas relacionadas com sistemas distribuídos, redes, gestão de infraestruturas, etc.. Mesmo tendo optado pela utilização do OpenNebula de forma a auxiliar e facilitar a instalação da *cloud*, recordo-me de alguns entraves que surgiram exatamente durante esse processo:

- por falta de conhecimento do gestor de virtualização kvm, não sabíamos que este requeria virtualização assistida por hardware, o que fez com que, após toda a configuração do OpenNebula ter sido feito com kvm, não conseguíssemos instanciar máquinas virtuais devido à falta desse requisito;

- o ponto acima obrigou assim à mudança de gestor de virtualização, tendo a escolha recaído sobre xen. Durante a sua instalação e configuração foram surgindo alguns conflitos que obrigaram à resolução de dependências, tendo mesmo havido uma situação pontual em que não foi possível resolver a questão remotamente com uma ligação à máquina utilizada no projeto;

- outros problemas derivados da utilização do xen, por exemplo, uns erros de configuração que estavam a impossibilitar que as máquinas virtuais instanciadas identificassem os discos de instalação dos sistemas operativos que pretendia instalar;

- incapacidade de estabelecer ligações VNC às máquinas virtuais através da interface de administração do OpenNebula. Na altura este problema não ficou resolvido, mas acabamos por estabelecer este tipo de ligações através de um cliente VNC externo.

Apesar da ocorrência das situações evidenciadas acima, estas acabaram por ser ultrapassadas sem se revelarem como sendo prejuízos importantes para a realização do projeto, tendo "apenas" obrigado a maior esforço e tempo despendido com esta parte. No final do projeto, os objetivos inicialmente propostos tinham sido alcançados, tendo o protótipo especificado e desenvolvido ficado de acordo com as expectativas, o que levou a satisfação do *stakeholders*, neste caso da equipa do lado da organização que financiou o projeto e à qual foram realizadas as apresentações e demonstrações do mesmo. De realçar também o interesse e o valor verificado deste projeto, através dos pareceres favoráveis obtidos em eventos científicos e técnicos internacionais em duas publicações já alcançadas: (Belo et al., 2013a) e (Belo et al., 2013b).

Se tomarmos em consideração apenas o âmbito do projeto, posso afirmar que os resultados alcançados satisfazem claramente os objetivos e pretensões inicialmente traçadas. Contudo, como o foco do mesmo era na auto-adaptabilidade de *dashboards* mediante processos de análise de dados, este aspeto limitou um pouco a exploração de outros temas subjacentes ao projeto. Relativamente à abordagem à *cloud*, esta foi planeada meramente numa ótica de utilização, e não de exploração, prova ou teste das capacidades deste sistema. Assim, durante todo o projeto não foram sequer usados os volumes de dados ou os acessos necessários que justificassem a realização de testes de carga e esforço sobre a *cloud*, de forma a observar a sua capacidade de adaptação a exigências pontuais. Assim, como trabalho futuro, para uma utilização num possível ambiente de produção seria necessário:

- adicionar mais infraestrutura física para suporte à *cloud*, de forma a que esta conseguisse disponibilizar mais recursos sempre que fosse necessário;
- proceder a uma implementação efetiva de um *cluster* de base de dados seguindo um arquitetura *shared-nothing*, bem como à definição e planeamento de um esquema de particionamento para o mesmo, adequado ao caso prático a tratar de forma a evitar que as interrogações e transações se propaguem por diversos nodos;
- verificar e assegurar o desempenho e eficiência do sistema multiagente e dos processos de migração executados por estas unidades computacionais, quando confrontados com maiores volumes de dados.

Claro que, para concretizar tais tarefas, poderá ser necessário realizar algumas adaptações ao estado atual do projeto. Contudo, prevê-se que tais mudanças sejam mínimas, pois tanto o

OpenNebula como o sistema multiagente foram inicialmente considerados exatamente para suportar estes requisitos de escalabilidade e desempenho que são necessários ao funcionamento dum sistema deste tipo num contexto real dentro de uma organização.

Bibliografia

Hacıgümüş, H., Iyer, B. & Mehrotra, S., 2002. Providing Databases as a Service. In *Proceedings 18th International Conference on Data Engineering.*, 2002.

Willis, J., 2009. *Did Google's Eric Schmidt Coin "Cloud Computing"?* [Online] Disponível em: http://cloudcomputing.sys-con.com/node/795054 [Acedido a 6 Janeiro 2013].

Abadi, D.J., 2009. Data Management in the Cloud: Limitations and Opportunities. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 32, pp.3-12.

Abah, J. & Ogwueleka, F.N., 2013. Cloud Computing with Related Enabling Technologies. *International Journal of Cloud Computing and Services Science* , 2(1), pp.40-49.

Armbrust, M. et al., 2009. *Above the Clouds: A Berkeley View of Cloud Computing*. Berkeley: University of California at Berkeley.

Arora, I. & Gupta, D.A., 2012. Cloud Databases: A Paradigm Shift in Databases. *IJCSI International Journal of Computer Science Issues*, 9(4), pp.77-83.

Buyya, R. et al., 2009. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems*, 25(6), pp.599-616.

Barsch, P., 2012. *Data Warehouse "as a Service" – A Good Pick for Mid-Sized Companies*. [Online] Disponível em: http://smartdatacollective.com/paulbarsch/88921/data-warehouse-service-good-pick-mid-sized-companies [Acedido a 18 Junho 2013].

Belo, O., Rodrigues, P. & Barros, R., 2013a. Adaptive Dashboarding - Reflecting Usage Preferences in OLAP. In *European Conference on Data Analysis (ECDA'2013)*. Luxemburgo, 2013a.

Belo, O., Rodrigues, P. & Barros, R., 2013b. Automatic Personalization of Analytical Dashboards. In *EWG-DSS "Exploring New Directions for Decisions in the Internet Age"*. Salonica, 2013b.

Chou, Y., 2011. *An Inconvenient Truth of the NIST Definition of Cloud Computing, SP 800-145*. [Online] Disponível em: http://blogs.technet.com/b/yungchou/archive/2011/12/19/an-inconvenient-truth-of-the-nist-definition-of-cloud-computing-sp-800-145.aspx [Acedido a 1 Fevereiro 2013].

Claburn, T., 2011. *Google Introduces Cloud Database*. [Online] Disponível em: http://www.informationweek.com/cloud-computing/platform/google-introduces-cloud-database/231900352 [Acedido a 18 Junho 2013].

Cohen, R., 2012. *Interest in Cloud Computing Has Peaked*. [Online] Disponível em: http://www.forbes.com/sites/reuvencohen/2012/05/24/interest-in-cloud-computing-has-peaked/ [Acedido a 12 January 2013].

Connolly, T.M. & Begg, C.E., 2004a. Data Redundancy and Update Anomalies. In *Database Systems: A Pratical Approach to Design, Implementation and Management*. 4th ed. Addison Wesley. pp.390-92.

Connolly, T.M. & Begg, C.E., 2004b. Distributed DBMSs – Concepts and Design. In *Database Systems: A Pratical Approach to Design, Implementation and Management*. 4th ed. Addison Wesley. p.693.

ContextLogic, 2013? *Treasure Data*. [Online] Disponível em: http://www.treasure-data.com/success-stories/contextlogic [Acedido a 19 Junho 2013].

Comissão Europeia, 2010. *The Future of Cloud Computing - Opportunities for European Cloud Computing Beyond 2010*. [Online] (1.0) Disponível em: http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf [Acedido a 26 Janeiro 2013].

Comparison of the conventional software-system management model with the cloud services, 2012? *Why 2013 Could Be a Great Year for Cloud Computing*. [Online] Disponível em: http://www.wfs.org/Upload/u174/201203071101401437.jpg [Acedido a 20 Junho 2013].

EMC Consulting, 2010. *Private Cloud Means Business: Costs Down and Agility Up*. [Online] EMC Corporation Disponível em: http://www.rbiassets.com/GetFile.ashx/72411121261 [Acedido a 1 Julho 2013].

DataZion, n.d. *Evolution of Cloud Computing - 1960 to 2012*. [Online] Disponível em: http://blog.datazion.com/wp-content/uploads/2012/12/cloud-computing-evolution.jpg [Acedido a 6 Janeiro 2013].

DeWitt, D.J., Madden, S. & Stonebraker, M., 200-. *How to Build a High-Performance Data Warehouse*. [Online] Disponível em: http://db.csail.mit.edu/madden/high_perf.pdf [Acedido a 10 Julho 2013].

Delphix, 2011. *The ABC's of DaaS – Enabling Data as a Service Application Delivery, Business Intelligence, and Compliance Reporting*. [Online] Delphix Corp. Disponível em: http://www.isaca.org/Groups/Professional-English/cloud-computing/GroupDocuments/Delphix_ABCs%20of%20DaaS.pdf [Acedido a 14 Junho 2013].

Foster, I., Zhao, Y., Raicu, I. & Lu, S., 2008. Cloud Computing and Grid Computing 360-Degree Compared. In *Grid Computing Environments Workshop*. Austin, 2008.

Gartner, 2012. *Gartner Releases Their Hype Cycle for Cloud Computing, 2012*. [Online] Disponível em: http://cdn.business2community.com/wp-content/uploads/2012/08/hype-cycle-for-cloud-computing-201211.jpg [Acedido a 12 January 2013].

Gelder, K.v., 2011? *Elastic Data Warehousing in the Cloud: Is the sky really the limit?* [Online] Disponível em: http://homepages.cwi.nl/~boncz/msc/2011-KeesvanGelder.pdf [Acedido a 9 Julho 2013].

Gilbert, S. & Lynch, N., 2002. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2), pp.51-59.

IBM, 2010. *Dispelling the vapour around cloud computing - Drivers, barriers and considerations for public and private cloud adoption*. [Online] IBM Disponível em: http://www.unitiv.com/Portals/51762/docs/ibm%20cloud%20computing.pdf [Acedido a 28 Julho 2013].

Inmon, W.H., 2000. The Data Warehouse Budget.

Inmon, W.H., 2005. *Building the Data Warehouse*. 4th ed. Indianápolis, Indiana: Wiley Publishing, Inc.

Heckel, P.C., 2010. *Hybrid Clouds: A Comparison of Cloud Toolkits*. [Online] Disponível em: http://blog.philippheckel.com/2010/05/08/hybrid-clouds-comparing-cloud-toolkits/3/#toc-3-cloud-toolkits [Acedido a 17 Julho 2013].

Hogan, M., ?a. *A Primer on Database Clustering Architectures*. [Online] ScaleDB Disponível em: http://scaledb.com/pdfs/APrimerDb.pdf [Acedido a 6 Julho 2013].

Hogan, M., ?b. *Shared-Disk vs. Shared-Nothing - Comparing Architectures for Clustered Databases*. [Online] ScaleDB Disponível em: http://www.scaledb.com/pdfs/WP_SDvSN.pdf [Acedido a 8 Julho 2013].

Janssen, C., n.d. *Data as a Service*. [Online] Disponível em: http://www.techopedia.com/definition/28560/data-as-a-service-daas [Acedido a 23 Outubro 2012].

Kim, W., 2009. Cloud Computing: Today and Tomorrow. *Journal of Object Technology*, 8(1).

Kimball, R., Reeves, L., Ross, M. & Thornthwaite, W., 1998. Project Management and Requirements. In *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*. 1st ed. Wiley.

Kleinrock, L., 2003. An Internet vision: the invisible global infrastructure. *AdHoc Networks Journal*, 1(1), pp.3-11.

Klopp, R., 2012. *Cloud Computing and Data Warehousing: Part 1 – The Architectural Issues*. [Online] Disponível em: http://robklopp.wordpress.com/2012/06/04/cloud-computing-and-data-warehousing-what-are-they-thinking/ [Acedido a 14 Julho 2013].

Kossmann, D., Kraska, T. & Loesing, S., 2010. An Evaluation of Alternative Architectures for Transaction Processing in the Cloud. In *2010 ACM SIGMOD International Conference on Management of data*. New York, 2010. ACM - Association for Computing Machinery.

Lee, S., 2011. Shared-Nothing vs. Shared-Disk Cloud Database Architecture. *International Journal of Energy, Information and Communications*, 2(4), pp.211-16.

NIST, 2011. *NIST Cloud Computing Standards Roadmap*. [Online] (1.0) Disponível em: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909024 [Acedido a 26 Janeiro 2013].

NIST, 2011. *The NIST Definition of Cloud Computing*. [Online] Disponível em: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf [Acedido a 26 Janeiro 2013].

Madsen, M., 2012. *Cloud Computing Models for Data Warehousing*. Third Nature.

Mathur, A., Mathur, M. & Upadhyay, P., 2011. Cloud Based Distributed Databases: The Future Ahead. *International Journal on Computer Science and Engineering (IJCSE)* , 3(6), pp.2477-81.

McCarthy, J., 1961. Speech given at MIT to celebrate its centennial. In *Time Sharing Computer Systems.*, 1961. MIT.

MobFox, 2013? *Success at MobFox*. [Online] Disponível em: http://www.treasure-data.com/success-stories/mobfox [Acedido a 19 Junho 2013].

OpenNebula, 2013a. *Planning the Installation 3.6*. [Online] Disponível em: http://opennebula.org/documentation:archives:rel3.6:plan [Acedido a 17 Julho 2013].

OpenNebula, 2013b. *An Overview of OpenNebula 4.0*. [Online] Disponível em: http://opennebula.org/documentation:rel4.0:intro [Acedido a 17 Julho 2013].

Oracle, 2010. *Oracle Cloud Computing*. [Online] Oracle Disponível em: http://www.oracle.com/us/technologies/cloud/oracle-cloud-computing-wp-076373.pdf [Acedido a 2 Fevereiro 2013].

Plummer, D.C. et al., 2008. *Cloud Computing: Defining and Describing an Emerging Phenomenon*. [Online] Gartner Disponível em: http://www.emory.edu/BUSINESS/readings/CloudComputing/Gartner_cloud_computing_defining.pdf [Acedido a 22 Janeiro 2013].

Sadashiv, N. & Kumar, S.M.D., 2011. Cluster, Grid and Cloud Computing: A Detailed Comparison. In *The 6th International Conference on Computer Science & Education*. Singapore, 2011.

Salazar, N.Y. & Jiming, H., 2012. Confidentiality and Availability of Data Warehouses in the Cloud Computing System. *Indian Journal of Computer Science and Engineering (IJCSE)*, 3(5), pp.720-30.

Slack, E., 2011. *How do you know that "Delete" means Delete in Cloud Storage?*. [Online] Disponível em: http://www.storage-switzerland.com/Articles/Entries/2011/8/16_How_do_you_know_that_Delete_means_Delete_in_Cloud_Storage.html [Acedido a 10 Julho 2013].

Russom, P., 2011. *TDWI Checklist Report - Consolidating Data Warehousing on a Private Cloud*. [Online] TDWI - The Data Warehouse Institute Disponível em: http://i.zdnet.com/whitepapers/Oracle_DW_US_EN_WP_Checklist_2.pdf [Acedido a 11 Julho 2013].

Rahm, E. & Do, H.H., 2000. Data Cleaning: Problems and Current Approaches. *IEEE Data Engineering Bulletin*, 23.

Rimal, B.P. & Lumb, I., 2009. A Taxonomy and Survey of Cloud Computing Systems. In *Fifth International Joint Conference on INC, IMS and IDC, 2009*. Seoul, 2009.

Timmermans, J., Stahl, B.C., Ikonen, V. & Bozdag, E., 2010. The Ethics of Cloud Computing: A Conceptual Review. In *IEEE Second International Conference on Cloud Computing Technology and Science - cloudcom 2010*. Indianapolis, 2010.

Traditional Data Warehouse and Treasure Data Process Comparison, 201-. *Architecture Overview*. [Online] Disponível em: http://docs.treasure-data.com/images/td_traditional_comparison.png [Acedido a 20 Junho 2013].