Partial Replication in the Database State Machine

António Sousa
als@di.uminho.pt

Orientador: Rui Oliveira
Universidade do Minho
Departamento de Informática
rco@di.uminho.pt

Início: Outubro 2000
Fim: Outubro 2003, ?? 2006

1 Resumo

1.1 Área de Investigação e Desenvolvimento

Distributed Systems.

Replicated Databases.

Fault Tolerance.

1.2 Resumo

Data replication has been one of the biggest trends in software technologies. Data is replicated at multiple networks nodes to increase availability and reliability.

Ideally, the system would provide the whole data exactly synchronized at all nodes without access contention or performance degradation. While for small groups of replicas on local area networks such a system is feasible, it does not scale up. The growth of the number of nodes and corresponding workload easily leads to an unbearable increase of synchronization problems. Moreover, this tight synchronization makes large geographical distribution impractical due to the unavoidable increase of operations latency.

Practical large-scale replicated systems find their way exploiting application semantics in order to relax replica consistency, and typically aim at performance sacrificing fault tolerance. Indeed, there are a number of information processing applications whose requirements are currently satisfied by on-line transaction processing on a central server or a set of geographically distributed independent servers, which may or may not periodically exchange information. By taking advantage of application semantics, it is possible to postpone synchronization to off-line batch processing, enhancing on-line performance as well as scalability. Reliability is addressed by the possibility of undoing updates of uncommitted transactions, restoring atomicity after system faults.

Nonetheless, there are signs that many institutions using these applications could benefit from improved consistency and reliability, both to address a changing environment and as an increasing competitive advantage. Services in popular domains, such as banking, electronic commerce, telecommunications, and public administration are currently facing compelling reasons to adjust their replication strategies to accommodate end-user real-time interactions.

This work addresses those problems using strong consistency replication strategies in existing large-scale systems by means of partial data replication. The intuition is simple: to overcome scalability problems, data that needs to be exactly synchronized cannot be replicated at all nodes but requires a judicious distribution. Besides performance, fault tolerance issues are also addresses by the proposed solutions.

1.3 Objectivos Estratégicos

Identify applications or application classes that may benefit from partial replication.

Compare total and partial replication requirements, identifying the conditions that allow both kinds of replication to use the same protocol.

Development of efficient protocols for full and partial replication.

Some of the results obtained by this work in the “ESCADA - fault tolerant scalable distributed databases” project have been used in the proposal for the "GORDA - Open Replication of Databases" project.

2 Contribuições

2.1 Principais contribuições técnico-científicas

- An optimistic total order protocol for wide-area networks.
- A protocol suitable for partial replication.
- Specification of the conditions that would allow the use of full replication protocols in partial replication scenarios.
- The use of the centralized simulation model, to evaluate the total order and replication protocols.

2.2 Publicações

2.2.1 Partial Replication in the Database State Machine [7]

This paper investigates the use of partial replication in the Database State Machine approach introduced earlier for fully replicated databases. It builds on the order and atomicity properties of group communication primitives to achieve strong consistency and proposes two new abstractions: Resilient Atomic Commit and Fast Atomic Broadcast.

Even with atomic broadcast, partial replication requires a termination protocol such as atomic commit to ensure transaction atomicity. With Resilient Atomic Commit our termination protocol allows the commit of a transaction despite the failure of some of the participants. Preliminary performance studies suggest that the additional cost of supporting partial replication can be mitigated through the use of Fast Atomic Broadcast.

2.2.2 Optimistic Total Order in Wide Area Networks[8]

Total order multicast greatly simplifies the implementation of fault-tolerant services using the replicated state machine approach. The additional latency of total ordering can be masked by taking advantage of spontaneous ordering observed in LANs: A tentative delivery allows the application to proceed in parallel with the ordering protocol. The effectiveness of the technique rests on the optimistic assumption that a large share of correctly ordered tentative deliveries offsets the cost of undoing the effect of mistakes. This paper proposes a simple technique which enables the usage of optimistic delivery also in WANs with much larger transmission delays where the optimistic assumption does not normally hold. Our proposal exploits local clocks and the stability of network delays to reduce the mistakes in the ordering of tentative deliveries. An experimental evaluation of a modified sequencer-based protocol is presented, illustrating the usefulness of the approach in fault-tolerant database management.

2.2.3 Testing the Dependability and Performance of Group Communication Based Database Replication Protocols[5]

Database replication based on group communication systems has recently been proposed as an efficient and resilient solution for large-scale data management. However, its evaluation has been conducted either on simplistic simulation models, which fail to assess concrete implementations, or on complete system implementations which are costly to test with realistic large-scale scenarios.

This paper presents a tool that combines implementations of replication and communication protocols under study with simulated network, database engine, and traffic generator models. Replication components can therefore be subjected to realistic large scale loads in a variety of scenarios, including fault-injection, while at the same time providing global observation and control. The paper shows first how the model is configured and validated to closely reproduce the behavior of a real system, and then how it is applied, allowing us to derive interesting conclusions both on replication and communication protocols and on their implementations.

2.2.4 Group-Based Replication of On-Line Transaction Processing Servers[3]

Several techniques for database replication using group communication have recently been proposed, namely, the Database State Machine, Postgres-R, and the NODO protocol. Although all rely on a totally ordered multicast for consistency, they differ substantially on how multicast is used. This results in different performance trade-offs which are hard to compare as each protocol is presented using a different load scenario and evaluation method.

In this paper we evaluate the suitability of such protocols for replication of On-Line Transaction Processing (OLTP) applications in clusters of servers and over wide area networks. This is achieved by implementing them using a common infra-structure and by using a standard workload. The results allows us to select the best protocol regarding performance and scalability in a demanding but realistic usage scenario.

2.2.5 Evaluating database replication in ESCADA (Position Paper)[6]

This paper reports our experience on the development and evaluation of group communication based database replication protocols in the ESCADA project, and points out several open issues of current research.

2.2.6 Revisiting Epsilon Serializabilty to improve the Database State Machine (Extended Abstract)[2]

In this paper, we investigate how to relax the consistency criteria of DBSM in a controlled manner according to the Epsilon Serializability (ESR) concepts and evaluate the direct benefits in terms of performance.

2.2.7 Evaluating Certification Protocols in the Partial Database State Machine[4]

Partial replication is an alluring technique to ensure the reliability of very large and geographically distributed databases while, at the same time, offering good performance. By correctly exploiting access locality most transactions become confined to a small subset of the database replicas thus reducing processing, storage access and communication overhead associated with replication.

The advantages of partial replication have however to be weighted against the added complexity that is required to manage it. In fact, if the chosen replica configuration prevents the local execution of transactions or if the overhead of consistency protocols offsets the savings of locality, potential gains cannot be realized. These issues are heavily dependent on the application used for evaluation and render simplistic benchmarks useless.

In this paper, we present a detailed analysis of Partial Database State Machine (PDBSM) replication by comparing alternative partial replication protocols with full replication. This is done using a realistic scenario based on a detailed network simulator and access patterns from an industry standard database benchmark. The results obtained allow us to identify the best configuration for typical on-line transaction processing applications.

2.2.8 Avaliação de um SGBD replicado usando simulação de redes[1]

A replicação de sistemas de gestão de bases de dados (SGBD) é um mecanismo fundamental para a fiabilidade de sistemas de informação. Em sistemas geograficamente distribuídos é ainda fundamental para recuperação de desastres e disponibilidade ubíqua de dados. Uma técnica de replicação recentemente proposta é a Database State Machine (DBSM), que promete aliar fiabilidade a elevado desempenho tirando partido de sistemas de comunicação em grupo. A avaliação do desempenho desta técnica tem no entanto sido efectuada com redes de comunicação demasiado simples ou irrealistas e com uma carga não representativa. Este artigo propõe uma avaliação rigorosa de uma concretização desta técnica de replicação, aliando um modelo de simulação realista de redes de comunicação com uma geração de carga efectuada de acordo com os padrões das medidas de desempenho elaboradas pelo Transaction Processing Council (TPC). Os resultados obtidos confirmam o interesse desta técnica em redes locais mas mostram que o seu desempenho é condicionado pelas características da rede e da carga.

[1] A. Correia Jr., A. Sousa, L. Soares, F. Moura, and R. Oliveira. Avaliação de um SGBD replicado usando simulaç ão de redes. In *in Proc. of 6a Conferência de Redes e Computadores (2003)*, October 2003.

[2] A. Correia Jr., A. Sousa, L. Soares, F. Moura, and R. Oliveira. Revisiting epsilon serializabilty to improve the database state machine (extended abstract). In *in Proc. of the Workshop on Dependable Distributed Data Management, SRDS (2004)*, October 2004.

[3] A. Correia Jr., A. Sousa, L. Soares, J. Pereira, F. Moura, and R.Oliveira. Group-based replication of on-line transaction processing servers. In Carlos Alberto Maziero, João Gabriel Silva, Aline Maria Santos Andrade, and Flávio Morais de Assis Silva, editors, *Dependable Computing: Second Latin-American Symposium, LADC 2005*, volume 3747, pages 245–260. Springer-Verlag GmbH, October 2005.

[4] A. Sousa, A. Correia Jr., F. Moura, J. Pereira, and R. Oliveira. Evaluating certification protocols in the partial database state machine. In *The First International Conference on Availability, Reliability and Security ("ARES 2006 – The International Dependability Conference")*. IEEE CS, to-appear 2006.

[5] A. Sousa, J. Pereira, L. Soares, A. Correia Jr., L. Rocha, R. Oliveira, and F. Moura. Testing the dependability and performance of group communication based database replication protocols. In *IEEE Intl. Conf. on Dependable*

Systems and Networks - Performance and Dependability Symposium (DSN-PDS'2005), 2005.

[6] A. Sousa, L. Soares, A. Correia Jr., R. Oliveira, and F. Moura. Evaluating database replication in escada (position paper). In *in Proc. of the Workshop on Dependable Distributed Data Management, SRDS (2004)*, October 2004.

[7] António Sousa, Fernando Pedone, Francisco Moura, and Rui Oliveira. Partial replication in the database state machine. In *n Proc. of the IEEE International Symposium on Network Computing and Applications (NCA 2001)*, pages 298–309. IEEE CS, October 2001.

[8] António Sousa, José Pereira, Francisco Moura, and Rui Oliveira. Optimistic total order in wide area networks. In *Proc. 21st IEEE Symposium on Reliable Distributed Systems*, pages 190–199. IEEE CS, October 2002.

3 Enquadramento

3.1 Enquadramento Científico

Num sistema distribuído a replicação assume um papel fundamental. Quer se trate de replicação de entidades passivas (dados) quer dinâmicas (processos, transacções, objectos) vários motivos a justificam: disponibilidade, eficiência e tolerância a faltas.

Embora intuitivamente simples de perceber e justificar a replicação exige no entanto um conjunto de serviços subjacentes que tornam o sistema complexo e que, não raramente, introduzem compromissos indesejáveis nos próprios objectivos do sistema. A complexidade do sistema provém imediatamente do conjunto de mecanismos e serviços necessários ao controlo de acesso, gestão, actualização e sincronização do conjunto de réplicas.

Idealmente um sistema replicado deve funcionar sem que os utilizadores se apercebam da replicação [8], isto é, o seu comportamento deveria ser equivalente ao de um sistema não replicado.

Em [18] os autores defendem que a manutenção de réplicas de bases de dados completamente sincronizadas é conseguida compromentendo a disponibilidade do sistema. No entanto, em [35, 34, 23] os autores mostram que em sistemas transacionais é possível manter réplicas de bases de dados completamente sincronizadas sem incorrer nesses custos. Estes trabalhos assumem um pequeno número de réplicas e exploram propriedades que se podem verificar em redes locais, mas que dificilmente se verificam noutro tipo de redes. O aumento do número de réplicas e a sua dispersão geográfica, introduzem dificuldades adicionais na aplicabilidade destes resultados, assim como novos desafios a quem trabalha nesta área.

A replicação completa de todos os dados em todas as réplicas nem sempre é possível e muitas vezes até é indesejado. A escolha criteriosa de quais os dados que devem ser replicados por cada réplica, i.e. a replicação parcial dos dados, de forma a estes serem mais facilmente acedidos, pode ser um caminho a seguir na procura de mecanismos de replicação que mantenham a coerência das réplicas do sistema sem comprometerem a sua disponibilidade. Este tipo de replicação, é uma área ainda pouco explorada que introduz novos desafios na forma como são actualizadas as diferentes réplicas dos dados.

Estes são alguns dos pressupostos na base do projecto ESCADA, que foram posteriormente incorporados no project GORDA.

3.2 Motivação

A maioria dos trabalhos de investigação em replicação de bases de dados centra-se na replicação total das bases de dados. No entanto quer pelo seu tamanho, quer pela dispersão geográfica das réplicas pode não ser viável nem desejável a replicação total dos dados.

Numa situação em que nem todas as réplicas replicam os mesmos dados será que os protocolos utilizados para replicação total continuam a ser válidos?

Os protocolos propostos para replicação em redes locais, utilizam protocolos de difusão optimistas numa tentativa de minimizar as latências dos protocolos de replicação. Será que os protocolos de difusão optimistas podem ser utilizados em redes de larga escala e com os mesmos resultados das redes locais? será possível desenvolver protocolos optimistas para estes cenários?

Uma das tarefas mais complicadas num sistema distribuído é a experimentação, pois dificilmente se consegue realiza-la num ambiente controlado sem interferências externas que a podem condicionar em mesmo inviabilizar. O recurso a simulação é muitas vezes utilizado para obter o ambiente controlado, mas por outro lado não permite avaliar o real impacto dos protocolos desenvolvidos. De forma a obter o melhor dos dois mundos utilizou-se o modelo de simulação centralizada [3], que permite combinar componentes reais com hardware e outros compomentes simulados. Esta técnica permite uma observação global e controlo do sistema dificil de obter num sistema real, permitindo ainda outras interacções com o sistema, como sejam a injecção de faltas.

3.3 Objectivos Detalhados

Identificar uma aplicação ou aplicações que possam beneficiar do uso de replicação parcial e utiliza-la como caso de estudo para o trabalho a desenvolver.

Na área das bases de dados existem um conjunto de benchmarks que caracterizam um conjunto padrão de aplicações e a carga a que as mesmas devem ser sujeitas. De

entre os benchmarks, o TPC-C [45] por ser aquele que explora de uma forma mais uniforme as várias operações das bases de dados, i.e. não priveligia as operações de leitura em relação às de escrita. Adicionalmente, o TPC-C permite explicitamente fragmentar a base de dados em bases de dados mais pequenas o que nos permitirá verificar até que ponto podemos beneficiar com o uso de replicação parcial.

Dos protocolos de replicação baseados em comunicação em grupo verificar a sua adequação a replicação parcial e caso seja necessário desenvolver um novo protocolo que permita colmatar esta lacuna dos protocolos de replicação de bases de dados.

O protocolo de difusão é uma parte essencial dos protocolos de replicação baseados em comunicação em grupo. O esforço feito na optimização destes protocolos em LANs permitiu obter melhorias no desempenho dos protocolo de replicação. No entanto as permissas em que estes se baseiam são as de uma LAN que são bastante diferentes das de uma WAN. Assim sendo é necessário verificar se os resultados obtidos são válidos em redes WAN. Caso estes resultados não sejam válidos será que é possivel efectuar algum tipo de optimização neste tipo de redes?

Uma dificuldade nos sistemas distribuídos é a experimentação. Assim sendo pretende-se desenvolver uma plataforma de teste que permita desenvolver a experimentação num ambiente controlado recorrendo a simulação, mas que simultaneamente permita avaliar o impacto do trabalho a desenvolver, utilizando código real.

3.4 Trabalhos Alternativos

A replication protocol suited to transactions encapsulated in stored procedures has been presented in [24]. It assumes data is a priori partitioned in conflict classes and a transaction accesses only one of such conflict classes. This requirement has been released in [32], allowing transactions to access a set of conflict classes named compound conflict class. Transactions accessing the same conflict class are supposed to have high conflict probability, while transactions in different conflict classes do not conflict and can be executed concurrently.

The Postgres-R [23], implements a group based replication protocol on Postgres 6.4.2 [37], a database using two phase locking for concurrency control. It uses atomic broadcast for communication requiring two messages per transaction, one to disseminate the transaction's write sets and another from the executing replica to commit or abort the transaction. The Postgres-R(SI) [49] implements the replication protocol on Postgres 7.2, a multiversion database engine. It provides snapshot-isolation as its consistency criteria [7], resulting in executions that may not always be serializable.

The clustered JDBC(C-JDBC) [11], is a flexible database clustering middleware. It addresses the scalability of database clusters by dispersing the load of the database by several back-ends using RAIDb [10].

The performance evaluation of the group communication based replication protocols has always been present [20, 23, 22, 5, 4, 49, 11]. The first proposed protocols using group communication primitives have been evaluated using simulation in [20].

The other results have been obtained in several scenarios and using distinct applications having in common the fact of using real implementations of the group communication and replication protocols. The comparison has been with the performance of a non-replicated database [49, 11], with implementations of distributed locking [23, 22] or with two phase commit protocols [5, 4]. Additionally in [4] the protocols were evaluated in wide area networks. In all evaluations of group based replication protocols, improvements in overall system performance have been obtained indicating that it seems an effective way of improving systems performance without sacrificing its correctness.

3.5 Bibliografia Principal

[1] D. Agrawal, G. Alonso, A. El Abbadi, and I. Stanoi. Exploiting atomic broadcast in replicated databases. In *Proceedings of EuroPar (EuroPar'97)*, Passau (Germany), 1997.

[2] Mustaque Ahamad, Mostafa H. Ammar, and Shun Yan Cheung. Multidimensional voting. *ACM Transactions on Computer Systems*, 9(4):399–431, 1991.

[3] G. Alvarez and F. Cristian. Applying simulation to the design and performance evaluation of fault-tolerant systems. In *Symp. Reliable Distributed Systems*, 1997.

[4] Y. Amir, C. Danilov, M. Miskin-Amir, J. Stanton, and C. Tutu. Practical wide-area database replication. Technical Report CNDS-2002-1, Johns Hopkins University, 2002.

[5] Y. Amir and C. Tutu. From total order to database replication. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, July 2002. IEEE.

[6] Daniel Barbara, Hector Garcia-Molina, and Annemarie Spauster. Increasing availability under mutual exclusion constraints with dynamic vote reassignment. *ACM Trans. Comput. Syst.*, 7(4):394–426, 1989.

[7] Hal Berenson, Philip A. Bernstein, Jim Gray, Jim Melton, Elizabeth J. O'Neil, and Patrick E. O'Neil. A critique of ansi sql isolation levels. In Michael J.

Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995*, pages 1–10. ACM Press, 1995.

[8] P. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[9] N. Budhiraja, K. Marzullo, F. Schneider, and S. Toueg. The primary-backup approach. In S. Mullender, editor, *Distributed Systems*, chapter 8, pages 199–216. Addison-Wesley, second edition, 1993.

[10] Emmanuel Cecchet, Julie Marguerite, and Willy Zwaenepoel. Raidb: Redundant array of inexpensive databases. Technical Report 4921, INRIA - Rhône-Alpes, September 2003.

[11] Emmanuel Cecchet, Julie Marguerite, and Willy Zwaenepoel. C-jdbc: Flexible database clustering middleware. In *USENIX Annual Technical Conference, FREENIX Track*, pages 9–18, 2004.

[12] Shun Yan Cheung, Mostafa H. Ammar, and Mustaque Ahamad. The grid protocol: A high performance scheme for maintaining replicated data. In *Proceedings of the Sixth International Conference on Data Engineering*, pages 438–445, Washington, DC, USA, 1990. IEEE Computer Society Press, Los Alamitos, Calif.

[13] X. Défago, A. Schiper, and N. Sergent. Semi-passive replication. In *Proceedings of 17th IEEE International Symposium on Reliable Distributed Systems*, pages 43–50, West lafayette, IN, USA, October 1998.

[14] Derek L. Eager and Kenneth C. Sevcik. Achieving robustness in distributed database systems. *ACM Trans. Database Syst.*, 8(3):354–381, 1983.

[15] Hector Garcia-Molina and Daniel Barbara. How to assign votes in a distributed system. *Journal of the ACM*, 32(4):841–860, 1985.

[16] David K. Gifford. Weighted voting for replicated data. In *Proceedings of the seventh symposium on Operating systems principles*, pages 150–162, 1979.

[17] Nathan Goodman, Dale Skeen, Arvola Chan, Umeshwar Dayal, Stephen Fox, and Daniel Ries. A recovery algorithm for a distributed database system. In *Proceedings of the 2nd ACM SIGACT-SIGMOD symposium on Principles of database systems*, pages 8–15. ACM Press, 1983.

[18] J. Gray, P. Helland, P. O'Neil, and D. Shasha. The dangers of replication and a solution. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 173–182, Montreal, Canada, June 1996.

[19] Maurice Herlihy. Dynamic quorum adjustment for partitioned data. *ACM Trans. Database Syst.*, 12(2):170–194, 1987.

[20] JoAnne Holliday, Diviakant Agrawall, and Amr El Abbadi. The performance of database replication with group multicast. In *Proceedings of IEEE International Symposium on Fault Tolerant Computing (FTCS29)*, pages 158–165, 1999.

[21] S. Jajodia and David Mutchler. Dynamic voting algorithms for maintaining the consistency of a replicated database. *ACM Trans. Database Syst.*, 15(2):230–280, 1990.

[22] R. Jimenez-Peris, M. Patino-Martinez, B. Kemme, and G. Alonso. Improving the scalability of fault-tolerant database clusters. In *22nd International Conference on Distributed Computing Systems (ICDCS '02)*, pages 477–484, Washington - Brussels - Tokyo, July 2002. IEEE.

[23] B. Kemme and G. Alonso. Don't be lazy, be consistent: Postgres-r, a new way to implement database replication. In A. El Abbadi, M. Brodie, S. Chakravarthy, U. Dayal, N. Kamel, G. Schlageter, and K. Whang, editors, *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt*, pages 134–143. Morgan Kaufmann, 2000.

[24] B. Kemme, F. Pedone, G. Alonso, and A. Schiper. Processing transactions over optimistic atomic broadcast protocols. In *Proc. the Int'l Conf. on Dist. Comp. Syst.*, Austin, Texas, June 1999.

[25] Bettina Kemme and Gustavo Alonso. A new approach to developing and implementing eager database replication protocols. *ACM Transactions on Database Systems*, 25(3):333–379, September 2000.

[26] Akhil Kumar. Hierarchical quorum consensus: A new algorithm for managing replicated data. *IEEE Transactions on Computers*, 40(9):996–1004, 1991.

[27] Akhil Kumar, Michael Rabinovich, and Rakesh K. Sinha. A performance study of general grid structures for replicated data. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 178–185. IEEE Computer Society Press, Los Alamitos, Calif., 1993.

[28] Moni Naor and Avishai Wool. The load, capacity, and availability of quorum systems. *SIAM J. Comput.*, 27(2):423–447, 1998.

[29] P. Erdös and l. Lovasz. Problems and results on 3-chromatic hypergraphs and some related questions. In A. Hajnal et al., editors, *Infinite and Finite Sets*, volume 11 of *Colloq. Math. Soc. Janos Bolyai*, pages 609–627. North-Holland, 1975.

[30] Jehan-François Pâris. Voting with witnesses: A constistency scheme for replicated files. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 606–612. IEEE Computer Society Press, Los Alamitos, Calif., 1986.

[31] Jehan-François Pâris. Voting with bystanders. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 394–405. IEEE Computer Society Press, Los Alamitos, Calif., 1989.

[32] M. Patinño-Martínez, R. Jiménez-Paris, B. Kemme, and G. Alonso. Scalable replication in database clusters. In M. Herlihy, editor, *Proceedings of hte* 14^{th} *Internationnal conference on Distributed Computing (DISC 2000)*, volume 1914 of *lecture notes in computer science*, pages 315–329, Toledo, Spain, 2000. Springer Verlag, Berlin.

[33] F. Pedone, R. Guerraoui, and A. Schiper. Exploiting atomic broadcast in replicated databases. In *Proceedings of EuroPar (EuroPar'98)*, Southampton, England, September 1998.

[34] F. Pedone, R. Guerraoui, and A. Schiper. The database state machine approach. *Journal of Distributed and Parallel Databases and Technology*, 2002.

[35] F. Pedone and A. Schiper. Optimistic atomic broadcast. In *Proceedings of the 12th International Symposium on Distributed Computing (DISC'98, formerly WDAG)*, September 1998.

[36] David Peleg and Avishai Wool. Crumbling walls: a class of practical and efficient quorum systems. *Distributed Computing*, 10(2):87–97, 1997.

[37] PostgreSQL. http://www.postgresql.org.

[38] D. Powell, M. Chéréque, and D. Drackley. Fault-tolerance in delta-4*. *ACM Operating Systems Review, SIGOPS Conference*, 25(2):122–125, April 1991.

[39] Jehan-François Páris and Darrell D. E. Long. Voting with regenerable volatile witnesses. In *Proceedings of the Seventh International Conference on Data Engineering*, pages 112–119, Washington, DC, USA, 1991. IEEE Computer Society.

[40] A. Schiper and M. Raynal. From group communication to transactions in distributed systems. *Communications of the ACM*, 39:84–87, April 1996.

[41] F. Schneider. Replication management using the state-machine approach. In S. Mullender, editor, *Distributed Systems*, chapter 7. Addison Wesley, second edition, 1993.

[42] I. Stanoi, D. Agrawal, and A. El Abbadi. Using broadcast primitives in replicated databases. In *Proceedings of the* 18^{th} *icdcs ICDCS'98*, pages 148–155, Amsterdam, The Netherlands, May 1998. IEEE.

[43] O. Theel and H. Pagnia-Koch. General design of grid-based data replication schemes using graphs and a few rules. In *ICDCS '95: Proceedings of the 15th International Conference on Distributed Computing Systems (ICDCS'95)*, page 395, Washington, DC, USA, 1995. IEEE Computer Society Press, Los Alamitos, Calif.

[44] R. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Transactions on Database Systems*, 4(2):180–209, jun 1979.

[45] Transaction Processing Performance Council (TPC). TPC Benchmark^{TM} C standard specification revision 5.0, February 2001.

[46] Irving L. Traiger, Jim Gray, Cesare A. Galtieri, and Bruce G. Lindsay. Transactions and consistency in distributed database systems. *ACM Transactions on Database Systems (TODS)*, 7(3):323–342, 1982.

[47] Robbert van Renesse and Andrew S. Tanenbaum. Voting with ghosts. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 456–462. IEEE Computer Society Press, Los Alamitos, Calif., 1988.

[48] Chienwen Wu and Geneva G. Belford. The triangular lattice protocol: A highly fault tolerant and highly efficient protocol for replicated data. In *IEEE International Symposium on Reliable Distributed Systems*, pages 66–73. IEEE Computer Society Press, Los Alamitos, Calif., 1992.

[49] Shuqing Wu and Bettina Kemme. Postgres-r(si): Combining replica control with concurrency control based on snapshot isolation. In *ICDE*, pages 422–433, 2005.

4 Desenvolvimento

4.1 Macro-planeamento das Actividades

Finalização da escrita da dissertação e correcções sugeridas pelo orientador.

4.2 Recursos Necessários

4.3 Recursos Disponibilizados

5 Avaliação

5.1 Análise Comparativa

As soluções propostas neste trabalho diferem das anteriormente propostas em:

- Aplicação utilizada na avaliação de desempenho. As soluções propostas eram avaliadas utilizando cargas sintécticas que se por um lado permitiam avaliar um determinado aspecto do sistema, não reflectiam a utilização no mundo real. A adopção do padrão de carga do TPC-C permitiu avaliar o comportamento dos protocolos num ambiente bem conhecido da comunidade de bases de dados.

- Metodologia utilizada na avaliação de desempenho. A avaliação de desempenho é feita combinando componentes simulados do sistema com componentes reais. Esta técnica permite avaliar e optimizar o componente em estudo sem a necessidade de operar um sistema distribuído em larga escala com toda a complexidade a ele inerente.

  Antes da utilização do sistema este é calibrado, tendo como ponto de referência a máquina em que serão executadas as experiências. Esta calibração tem como objectivo garantir que em condições semelhantes o modelo real e o simulado apresentam o mesmo comportamento.

- Nível de integração com o DBMS (motor da base de dados). Os protocolos propostos assumem uma clara integração/cooperação com o DBMS utilizado. Uma vez que a informação necessária não é disponibilizada actualmente pelos DBMS, optou-se pela utilização de um DBMS open-source que foi extendido para fornecer a informação necessária para os protocolos utilizados.

  Dos trabalhos existentes os que fazem replicação ao nível do middleware utilizam uma unidade de concorrência mais lata, por exemplo a tabela e não o tuplo, ou então têm que duplicar as funcionalidades do DBMS em middleware. Os que fazem a replicação in-core, utilizam normalmente protocolos mais pesados, isto é, com um maior número de passos de comunição, pois a certificação das transacções é feita pela réplica que executa a transacção.

5.2 Auto-avaliação do Trabalho Realizado

O trabalho realizado até ao momento atingiu os objectivos pretendidos, faltando apenas a conclusão da dissertação.

5.3 Auto-avaliação da Documentação Produzida

A documentação produzida até ao momento foi na sua maioria publicada em conferências da IEEE (NCA2001, SRDS2002, SRDS2004, DSN-PDS2005).

As publicações estão interligadas complementando-se, e algumas delas já forma citadas mais do que uma vez em trabalhos não desenvolvidos na Universidade do Minho.

6 Dificuldades

6.1 Dificuldades Técnico-Científicas

6.2 Outras Dificuldades