BFT Selection

Ali Shoker and Jean-Paul Bahsoun

University of Toulouse III, IRIT, Toulouse, France
{ali.shoker, jean-paul.bahsoun}@irit.fr

Abstract. This paper presents the first BFT selection model and algorithm that can be used to choose the most convenient protocol according to the BFT user (i.e., an enterprise) preferences. The selection algorithm applies some mathematical formulas to make the selection process easy and automatic. The algorithm operates in three modes: Static, Dynamic, and Heuristic. The Static mode addresses the cases where a single protocol is needed; the Dynamic mode assumes that the system conditions are quite fluctuating and thus requires runtime decisions, and the Heuristic mode uses additional heuristics to improve user choices. To the best of our knowledge, this is the first work that addresses selection in BFT.

Keywords: Byzantine fault tolerance, BFT selection, dynamic switching.

1 Introduction

Byzantine fault tolerance [1] (BFT) is a replication-based approach used to maintain the resiliency of services, often state-machines, against *Byzantine* (i.e., arbitrary) faults in a partially synchronous [1] environments. Many BFT protocols have been introduced so far to maintain safety and liveness in such systems; however, no one-size-fits-all protocol was proposed. A vast discrepancy can be noticed among these protocols which governs their characteristics and performance. This can bring some confusion to BFT users [1] to choose the protocol that is most convenient to their services according to their own demands. Choosing a convenient protocol can be hard when the candidate protocols and their characteristics are numerous. Guerraoui et al. [2] proposed an abortable framework to launch alternating BFT protocols on the same service based on the changes in the underlying system conditions; however, this approach did not introduce any switching policy to run the candidate protocols efficiently and dynamically.

In this paper, we introduce the first BFT selection model and algorithm that automates the selection process of the 'preferred' BFT protocol among a set of candidate ones. The 'preferred protocol' is the one that matches user preferences the most. An evaluation process is in charge of matching the user preferences against the profiles of the nominated BFT protocols considering both: reliability

[1] A BFT user in our context is any enterprise that is choosing a BFT protocol to deploy on its services.

and performance. The selected protocol is the one that achieves the highest evaluation score. The mechanism is automated via mathematical matrices, and produces selections that are reasonable and close to reality. We explore in this paper the selection model and algorithm. The selection algorithm operates in three modes: Static, Dynamic, and Heuristic. We focus on discussing the Static mode, and we describe the Dynamic and Heuristic modes in [3, 4].

Though our model is generic (it may cover any functional and non-functional protocol), we introduce it in the context of BFT for two main reasons. First, to make a first steps towards dynamic switching between existing BFT protocols at runtime, and second, to make it easier for enterprises to select their preferred BFT protocol when BFT is provided as a service, e.g., in clouds.

The rest of the paper is organized as follows. We introduce the selection model and the selection algorithm in Sections 2 and 3, respectively. We address the evaluation in Section 4, and we conclude our paper in Section 5.

2 BFT Selection Model

2.1 Notations and Terms

We define two types of indicators: Key Characteristic Indicators (KCI) and Key Performance Indicators (KPI). KCIs are those properties (with boolean values) of a protocol that indicate its properties, and requirements, e.g., 'the minimum number of replicas needed'. The KCI can strictly decide whether an evaluated protocol could be selected or not. The KPIs are the properties that evaluate the performance of the protocol like throughput, and latency. These values are usually real numbers. KPIs are used to recommend a protocol over the others but, in general, it could not rule out a protocol. In addition, we define the system state by $S = \{s_i = (f_1, f_2, ..., f_j, ..., f_m)\}$ where f_j represents the j^{th} impact factor of the system state and m is the number of considered impact factors. 'Number of clients' and 'message size' are examples of impact factors.

2.2 Selection Model

Consider a service provider (e.g., a cloud vendor) that offers n different BFT protocols along with its provided services (e.g., signed in SLA contract). We define the set of BFT protocols $\psi = \{p_i;\ where\ 1 \leq i \leq n\}$. On the other hand, consider a selection model represented by:$\Sigma = \{$Protocol,User,Mode$\}$. Protocol represents the profile of a BFT protocol, User represents the preferences of the user (i.e., the enterprise), and Mode represents the selection mode of the system. Selection occurs through matching the Protocol profile with the User preferences according to the mapping: $f : \Sigma \longmapsto \psi$; this yields the 'preferred' protocol among all competing protocols. Here we define the 'preferred' protocol:

Definition 1. *A protocol p_i with profile $Protocol_i$ is called the 'preferred' protocol among a set of candidate protocols ψ with respect to a specific user with preferences $User_j$ if and only if according to an evaluation function $e : \Sigma \longmapsto \Re$, $e(Protocol_i, User_j, \phi)$ is maximal.*

The interpretation of Protocol, User, and Mode is as follows:

Protocol. Each protocol has a profile: Protocol=$\{A_P, A_U, B_P, B_V\}$. $A_P = (\alpha_1, \alpha_2, ..., \alpha_a)$ is a vector of a KCIs. A_U represents the vector of the default weights of these KCIs: $A_U = (u_1, u_2, ..., u_a)$. $B_P = (\beta_1, \beta_2, ..., \beta_b)$ is a vector of b KPIs and, finally, B_V represents the vector of the default weights of these KPIs: $B_V = (v_1, v_2, ..., v_b)$.

User. Each user, e.g., an enterprise, defines his preferences in User=$\{U, V, M\}$, where U (resp., V) is a vector of user defined weights corresponding to the KCIs (resp., KPIs) of the Protocol's preference A_P (resp., B_P). M defines the mode required by the user, i.e, either Dynamic, Static, or Heuristic.

Mode. The selection can occur in three different modes: Static, Dynamic, or Heuristic. In the former, the selection occurs only once, i.e., at the time the BFT user requires a service; afterwards, the user does not change his selection (i.e., the used protocol) until the system is halted/rebooted and, thus a new selection is provoked. On the other hand, the Dynamic mode makes the system react dynamically to the changes of the system state. This mode allows the system to adapt to the upcoming conditions at runtime and hence the user will be using multiple alternating protocols. The Heuristic mode uses some heuristics to adjust the preferences of the user, especially V, to improve his choices in some cases.

3 Selection Mechanism

The selection mechanism of the preferred protocol according to the user preferences is achieved through computing the evaluation scores E of the competing protocols, and then electing the protocol that corresponds to the maximum score. For any state s, and protocol $p_i \in \psi$ that has an evaluation score $E_{i,s}$; a protocol p_{pref} is chosen according to Equation 1:

$$p_{pref} = p_i, \ s.t. \ E_{i,s} = \max_{1 \leq j \leq n} E_{j,s}. \tag{1}$$

If the mode of the system is Dynamic or Heuristic, the KPIs are computed at runtime, and the system chooses the protocol that has the highest evaluation score E among all protocols to launch it in the next phase. To make computations easier, we define a new operator, i.e., the OR product $\dot{\vee}$.

Definition 2. *Consider two boolean matrices $A \in \{0,1\}^{n \times l}, B \in \{0,1\}^{l \times m}$ with entries a_{ij}, and b_{ij}, respectively. The OR product $A \dot{\vee} B$ is a matrix $C = A \dot{\vee} B \in \mathbb{N}^{n \times m}$, where its elements are defined by: $c_{ij} = \sum_{k=1}^{m} a_{ik} \vee b_{kj}$. The operator $\vee$ is the logical OR operator.*

$$\begin{cases} E = C \circ P \\ where \ C = \left\lfloor \frac{1}{a}.(A \ \dot{\vee} \ (e_n - U)) \right\rfloor \\ and \ P = B^{\pm}.(V \circ W). \end{cases} \tag{2}$$

The evaluation score E is calculated according to the formulas introduced in Equation 2. The evaluation matrix E is the Schur product of the KCI matrix C and the KPI matrix P. C represents the part of the evaluation that deals with the KCIs of the profiles of the protocols; whereas, P represents the evaluation part that deals with the KPIs. E is calculated after computing the values of C and P. If the mode of the system is Dynamic or Heuristic, then E may change at runtime as P changes.

The KCI matrix $C = \left\lfloor \frac{1}{a}.(A \dot{\vee} (e_n - U)) \right\rfloor$ matches the user preferences against the profiles of different protocols. a represents the number of KCIs considered. The operator $\lfloor\ \rfloor$ is the integer value operator (it is sometimes indicated by [ ] too). The operator $\dot{\vee}$ was defined in Definition 2. Matrix A represents the profiles of the protocols. The dimension of A is $n \times a$; where n is the number of candidate protocols and a is the number of KCIs considered in the evaluation. Each row of the matrix represents a KCI vector profile of a protocol. Matrix U represents the preferences of the user. According to this matrix, the protocols that satisfy all user requirements will be considered for selection (i.e., will continue the competition). On the contrary, the protocol that lacks a single property among those demanded by the user will be out of selection. The column matrix e_n is a unit matrix is to invert the values of the matrix U to $-U$. After defining the matrices A and U, the computation of C becomes straightforward.

Matrix P is used to complete the selection process by considering the KPIs of the protocols, seeking better performance. The KPI matrix P is defined in the formula: $P = B^{\pm}.(V \circ W)$. $B^{\pm}$ is a normalized version of another matrix B that represents the KPI profiles of each protocol. Each profile is presented in one row. B and $B^{\pm}$ have the same dimension $n \times b$ where n is the number of protocols and b is the number of KPIs considered. The entries of the matrix $B^{\pm}$ are denoted by $\beta^{\pm}$ and are calculated from the entries of B that are denoted by β. We say that a KPI has the property Tendency='high', if a higher value means better evaluation score E, e.g., throughput; this KPI is denoted by β^{+}. On the contrary, a KPI of type β^{-} has the property Tendency='low', e.g., latency, and a higher KPI value means worst evaluation score E. Suppose the number of β-KPIs is b, then the matrix B can be divided into b column matrices (i.e., vectors): $B_1, B_2, B_i, ...,$ and B_b. Let the maximum (resp., minimum) value of the entries of each vector B_i be max_i (resp., min_i). Then, the entries of the matrix $B^{\pm}$ can be calculated according to Equation 3:

$$\begin{cases} \beta^{+}_{ji} = 1 - \dfrac{max_i - \beta_{ji}}{max_i - min_i}; \\ \beta^{-}_{ji} = 1 - \dfrac{\beta_{ji} - min_i}{max_i - min_i}; \\ where\ i \leq b\ and\ j \leq n. \end{cases} \tag{3}$$

Matrix V represents the KPI user preferences used to recommend a protocol. V is a column matrix of dimension $b \times 1$, where b is the number of KPIs considered in the evaluation. The entries of this matrix follow two constraints: (1) all entries $\in$ [0,10], and (2) their sum $\sum_{i=1}^{b} v_{i1} = 10$. Matrix W is a column matrix used in

the Heuristic mode only. W is important to adjust the user preferences given in V by considering the system state to improve his choice according to predefined heuristic rules. If the mode is Static, then the entries of W are equal to 1, i.e., W=e_b.

4 Evaluation

To evaluate our approach we have considered seven existing BFT protocols by listing their different KCIs like the number of replicas needed, speculative or not, tolerate malicious clients or not, etc. Also we have considered three KPIs: throughput, latency, and capacity. The KPI values are estimated based on the message exchange patterns of the different protocols. Our mechanism gave selection results as expected according to many user preferences we have chosen. The mechanism minimizes the complexity of selection significantly. Due to lack of space, we do not reveal our examples and results in this paper, but we encourage the reader to read our extended papers in [3, 4].

5 Conclusion

We presented a BFT selection model and algorithm to automate the selection of the 'preferred' BFT protocol according the preferences defined by the BFT user, i.e., an enterprise. This is useful in large services that provide BFT as a service, and in fluctuating systems that require dynamic runtime switching of BFT protocols as the underlying system conditions change. We consider three modes: (1) Static mode: where the user chooses a protocol only once; he can only change it when the service is rebooted. (2) Dynamic mode: which allows the user to multiple protocols, where a running protocol can be stopped and another protocol is launched after performing selection process. The intuition is that the performance of protocols differ as the underlying system state changes, and thus adapting to the new state is required. (3) Heuristic mode: this mode is similar to the Dynamic mode; however, it allows to modify the weights (i.e., preferences) chosen by the user as the system state changes using some predefined heuristics. This paper focused on the Static mode, while future work addresses the other interesting modes: Dynamic and Heuristic.

References

1. Castro, M., Liskov, B.: Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. **20**(4) (2002) 398–461
2. Guerraoui, R., Knežević, N., Quéma, V., Vukolić, M.: The next 700 bft protocols. In: EuroSys '10: Proceedings of the 5th European conference on Computer systems, New York, NY, USA, ACM (2010) 363–376
3. Shoker, A., Bahsoun, J.P.: Bft selection. Technical report, IRIT (2013)
4. Shoker, A.: Byzantine fault tolerance: From static selection to dynamic switching. Technical report, IRIT (2012)