Optimal inter-object correlation when replicating for availability

Haifeng Yu · Phillip B. Gibbons

Received: 23 August 2007 / Accepted: 9 October 2008 / Published online: 11 November 2008
© Springer-Verlag 2008

Abstract Data replication is a key technique for ensuring data availability. Traditionally, researchers have focused on the availability of individual objects, even though user-level tasks (called operations) typically request multiple objects. Our recent experimental study has shown that the assignment of object replicas to machines results in subtle yet dramatic effects on the availability of these operations, even though the availability of individual objects remains the same. This paper is the first to approach the assignment problem from a theoretical perspective, and obtains a series of results regarding assignments that provide the best and the worst availability for user-level operations. We use a range of techniques to obtain our results, from standard combinatorial techniques and hill climbing methods to Janson’s inequality (a strong probabilistic tool). Some of the results demonstrate that even quite simple versions of the assignment problem can have surprising answers.

Keywords Multi-object operation · Inter-object correlation · Availability · Data replication · Object assignment

1 Introduction

Masking failures is a key goal in distributed computing, and data replication is a well-known and widely used technique to ensure data availability in the presence of failures. Traditionally, researchers typically focus on the availability of individual data objects (e.g., individual file blocks [4] or individual variable-sized objects [7, 13]). On the other hand, a user-level task often needs to request multiple data objects; we refer to this as a multi-object operation. For example, in order to compile a project, all source files need to be available. Similarly, a database query usually touches multiple database objects. Our recent experimental study [24] shows that the assignment of object replicas to machines has a subtle yet critical effect on the availability of such multi-object operations, even though the availability of individual objects remains the same.

A simple yet subtle example Consider the example in Fig. 1 with four objects: A, B, C, and D. Each object has exactly two replicas. We have four identical machines to hold these eight object replicas, and each machine holds exactly two objects. Each machine may fail (crash) independently with the same probability p, causing all its data to become unavailable. An object is unavailable if and only if both its replicas are unavailable. Clearly, there are many ways to assign the object replicas to the machines. Figure 1 gives two possible assignments.

Imagine that the four objects are source files of a project and the user is trying to compile the project. Here, if any source file is unavailable, the multi-object operation (i.e., the compilation process) will fail. Which assignments in Fig. 1

1 The same assignment problem also arises [24] when using erasure coding for the objects, but that is beyond the scope of this paper.
four machines. Each box represents a machine.

![Fig. 1 Two possible assignments of four objects, A, B, C, and D, to four machines. Each box represents a machine](image)

The failure probabilities are \( FP(\alpha) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2 \) and \( FP(\beta) = p^4 + 4p^3(1-p) + 4p^2(1-p)^2 \).

2 The failure probabilities are \( FP(\alpha) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2 \) and \( FP(\beta) = p^4 + 4p^3(1-p) + 4p^2(1-p)^2 \).

3 The failure probabilities (i.e., the probabilities that fewer than three objects are available) are \( FP(\alpha) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2 \) and \( FP(\beta) = p^4 + 4p^3(1-p) \).

Our goal in this paper is to find the best and the worst assignments, among all possible assignments, in terms of the

Previous results We recently first identified [24] the availability effects of object assignments and inter-object correlation on multi-object operations. This earlier work used simulation to compare several specific assignments, including the PTN and RAND assignments. PTN is the assignment where we partition the objects into sets and mirror each set across multiple machines (as in assignment \( \alpha \) in Fig. 1). RAND is obtained by randomly assigning object replicas to machines. The simulation results [24] show that: (1) previously proposed assignments can result in dramatically different availability; (2) if the multi-object operation cannot tolerate any missing objects, then PTN and RAND provide the best and the worst availability among the set of assignments simulated, respectively; and (3) in contrast, if the multi-object operation can tolerate a sufficient number of missing objects, then PTN and RAND provide the worst and the best availability among the set of assignments simulated, respectively. Our earlier work [24] also proposed designs to approximate PTN and RAND for dynamic contexts, and performed a thorough evaluation of their implementation on the PlanetLab. We are not aware of any other previous work on this topic.

Our results To the best of our knowledge, this paper is the first to study this problem from a theoretical perspective. Experimental methods as in [24] have the following fundamental limitations: given the exponential number of possible assignments, it is infeasible to experiment with them all. Are there better assignments that were overlooked? Also, experimental methods can cover only specific parameter values (e.g., specific \( p \) values)—will the same conclusions hold under other parameter values? The theoretical results in this paper not only provide a deep understanding of the problem, but also help to ultimately confirm what was observed experimentally.

Our goal in this paper is to find the best and the worst assignments, among all possible assignments, in terms of the

Practical importance The object assignment problem has significant practical relevance, and applies to almost all replication systems. A long list of previous replication systems and protocols (such as CAN [15], CFS [4], Chord [20], Coda [11], FARSITE [1], GFS [6], GHT [16], Glacier [7], Pastry [18], R-CHash [10], and RIO [19]) use different object assignments, which can yield dramatically different availability for multi-object operations. For example, it has been shown [24] that under practical settings, the failure probability of the TPC benchmark [22] can vary by four orders of magnitude under different assignments used in these previous systems. Thus, a thorough understanding of the subject is critical to guide system design.

Our results To the best of our knowledge, this paper is the first to study this problem from a theoretical perspective. Experimental methods as in [24] have the following fundamental limitations: given the exponential number of possible assignments, it is infeasible to experiment with them all. Are there better assignments that were overlooked? Also, experimental methods can cover only specific parameter values (e.g., specific \( p \) values)—will the same conclusions hold under other parameter values? The theoretical results in this paper not only provide a deep understanding of the problem, but also help to ultimately confirm what was observed experimentally.

Our goal in this paper is to find the best and the worst assignments, among all possible assignments, in terms of the

Practical importance The object assignment problem has significant practical relevance, and applies to almost all replication systems. A long list of previous replication systems and protocols (such as CAN [15], CFS [4], Chord [20], Coda [11], FARSITE [1], GFS [6], GHT [16], Glacier [7], Pastry [18], R-CHash [10], and RIO [19]) use different object assignments, which can yield dramatically different availability for multi-object operations. For example, it has been shown [24] that under practical settings, the failure probability of the TPC benchmark [22] can vary by four orders of magnitude under different assignments used in these previous systems. Thus, a thorough understanding of the subject is critical to guide system design.