A Framework for Efficient Inconsistency Detection in a Grid and Internet-Scale Distributed Environment

Yijun Lu and Hong Jiang
Department of Computer Science and Engineering
University of Nebraka-Lincoln
{yijlu, jiang}@cse.unl.edu

Abstract

In this paper, we argue that a broad range of Internet-scale distributed applications can benefit from an underlying low-cost consistency detection framework – an alternative to inconsistency avoidance that can detect inconsistency among nodes sharing data or services in a timely manner. .

After introducing a framework of inconsistency detection, this paper presents the design and evaluation of a two-layer inconsistency detection module. The proposed two-layer inconsistency detection module is evaluated by both analysis and simulations. The results show that this framework can significantly reduce the time to detect inconsistency among nodes without adding much maintenance cost.

1. Introduction

Conventionally, consistency control is designed to avoid the inconsistency up-front. Well-defined consistency protocols, such as strong consistency protocols or optimistic consistency protocols that increase the availability while tolerate relaxed inconsistency among nodes, are predefined and deployed before the system starts to run an application. In this paper, we refer to this scheme as inconsistency *avoidance*.

While inconsistency avoidance can be effective in a small-scale networked system, such as a small cluster, it has some drawbacks in an Internet-scale environment, such as Grid or large-scale distributed e-business applications. In this environment, a strong consistency protocol can be very costly to maintain because of the membership maintenance and strict protocol enforcement cost [2]. While optimistic consistency protocol relieves the costly maintenance and strict enforcement burden associated with strong consistency protocols, it also does not suit the large-scale distributed system because it is *predefined*. In an environment where many applications are deployed, providing a predefined consistency protocol can be either overkill when an application does not need that strong consistency, or insufficient when an application needs stronger ones.

This paper proposes a framework to detect inconsistency in a timely manner when it occurs instead of avoiding it in the first place. We refer this as inconsistency *detection*.

Comparing to inconsistency avoidance, several advantages can be obtained from an inconsistency detection framework. First, it removes the costly membership management requirement that is used to enforce a consistency in the first place. Instead, it detects the inconsistency when it happens. That makes the system scalable. Besides, by ensuring that the potential inconsistent behavior be detected in a *timely* manner, a system can combine the results (if the results are combinable), break the tie or alert the users so that they can resolve the conflict as soon as possible using appropriate resolution protocols.

Second, after the inconsistency is detected, the system can respond based on the application semantics. That is, it resolves the inconsistency when it is needed, while letting the detected inconsistency continue to exist when it is tolerable or even preferred.

2. Overview of the Inconsistency Detection Framework

As an alternative to inconsistency avoidance, the inconsistency detection framework detects inconsistency among nodes in a timely manner. A logical diagram of this framework is shown in Figure 1.

In this framework, multiple applications share data and services through the support of the Internet-scale middleware and the inconsistencies among them are detected by the detector. Upon detection, the detector consults with the inconsistency level monitor (step 1 and step 2) before reaction is initiated. Based on the applications' semantics, if the inconsistency is tolerable, the detector does not react; otherwise, the detector informs the inconsistency resolution model to resolve this inconsistency (step 3).

3. Timely Inconsistency Detection

The basic idea is to build an overlay on top of the underlying network based on nodes' updating history. As the top layer is based on nodes' updating history, or updating temperature, it is referred as "temperature overlay". The bottom layer of gossip-based [1] inconsistency detection is used as a backup and only triggered when the top layer does not find any inconsistency. The architecture of the framework is illustrated in Figure 2.

In the temperature overlay, each node tracks its own updating history and exchanges this information with others through the RanSub [3] protocol periodically. When a node commits an update, this update is propagated in the temperature overlay in such a way that the nodes that update this file most frequently are visited first.

4. Evaluation

Probability that the top-layer fails: By analysis, the probability that the top layer fails to detect conflicts is quite low when the file is hot (as low as 0.04%). This result fits well with our design goal, which is to minimize the delay of inconsistency detection. The proposed scheme is especially effective when there are a lot of spontaneous updates, an indication that a file is becoming hot.

Maintenance cost: The number of messages received by each node during the maintenance process, also called node stress in the literature, is used to evaluate the maintenance cost of the temperature overlay.

We run the two-layer inconsistency detection module for 800 seconds. At the end of the simulation, we collect the number of messages received by each node and the result is illustrated in Table 1.

Although the Max (which comes from the root of the tree which RanSub uses) is much higher than the mean value, it must be pointed out that it is accumulated over 800 seconds. Even if the size of a message is 1KB, the network bandwidth cost is only 6KB/s for the root. From that we can see that the maintenance cost will not overwhelm the root.

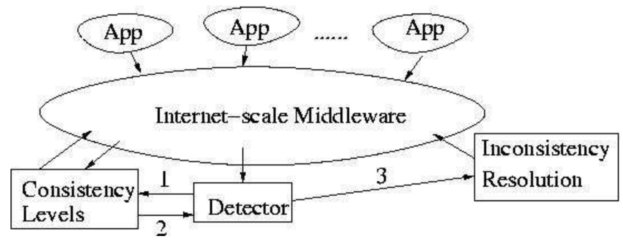

Figure1. Architecture of the Inconsistency Detection Framework

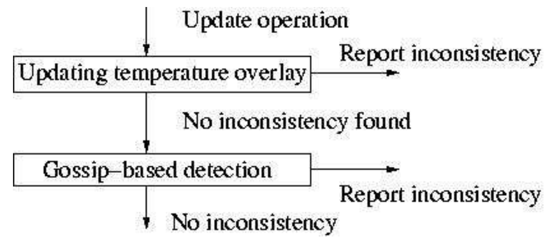

Figure2. Architecture of the two-layer Inconsistency Detection Module

Table 1. Maintenance cost

Max	Mean	Median
4680	51.9	26

6. Conclusions

In this paper, we presented an inconsistency detection framework, as an alternative to inconsistency avoidance, in Internet-scale distributed systems. The basic idea and evaluation of a two-layer inconsistency detection module was elaborated.

References

[1] P.T. Eugster, R. Guerraoui, S. B. Handurukande, A. M. Kermarrec, P. Kouznetsov. Lightweight Probabilistic Broadcast, In Proc of the International Conference on Dependable Systems and Networks (DSN 2001), July, 2001

[2] D. Dullmann, W. Hoschek, J. Jaen-Martinez, B. Segal, A. Samar, H. Stockinger, and K. Stockinger, Models for Replica Synchronization and Consistency in Data Grid, In Proc. of 10^{th} IEEE International Symposium on High Performance Distributed Computing (HPDC), Aug. 7-9, pp. 67-75, 2001

[3] D. Kostic, A. Rodriguez, J. Albrecht, A. Bhirud, and A. Vahdat. Using Random Subsets to Build Scalable Network Servioces, In Proc. of 4^{th} USENIX Symposium on Internet Technologies and Systems. March 2003