# IDEA: An Infrastructure for Detection-based Adaptive Consistency Control in Replicated Services

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

## Consistency control is important in replicated services

- In large-scale systems, such as the Grid, replcating data and services can provide continuous service and prevent data loss.

- Poor consistency results in poor QoS or even monetary loss (in e-business applications).

## Why do we need adaptive consistency control?

- The state-of-the-art

Current research focuses on striking a tradeoff between consistency guarantee and system's scalability by enforcing a certain level of, instead of pefect, consistency.

- Adaptability is needed because:

(1) Multiple applications with different consistency requirements can run simultaneously on a modern distributed computer system.

(2) Even one application's consistency requirement can change from time to time.

## A detection-based consistency control

- The inconsistencies among participants are detected by a timely detection module.

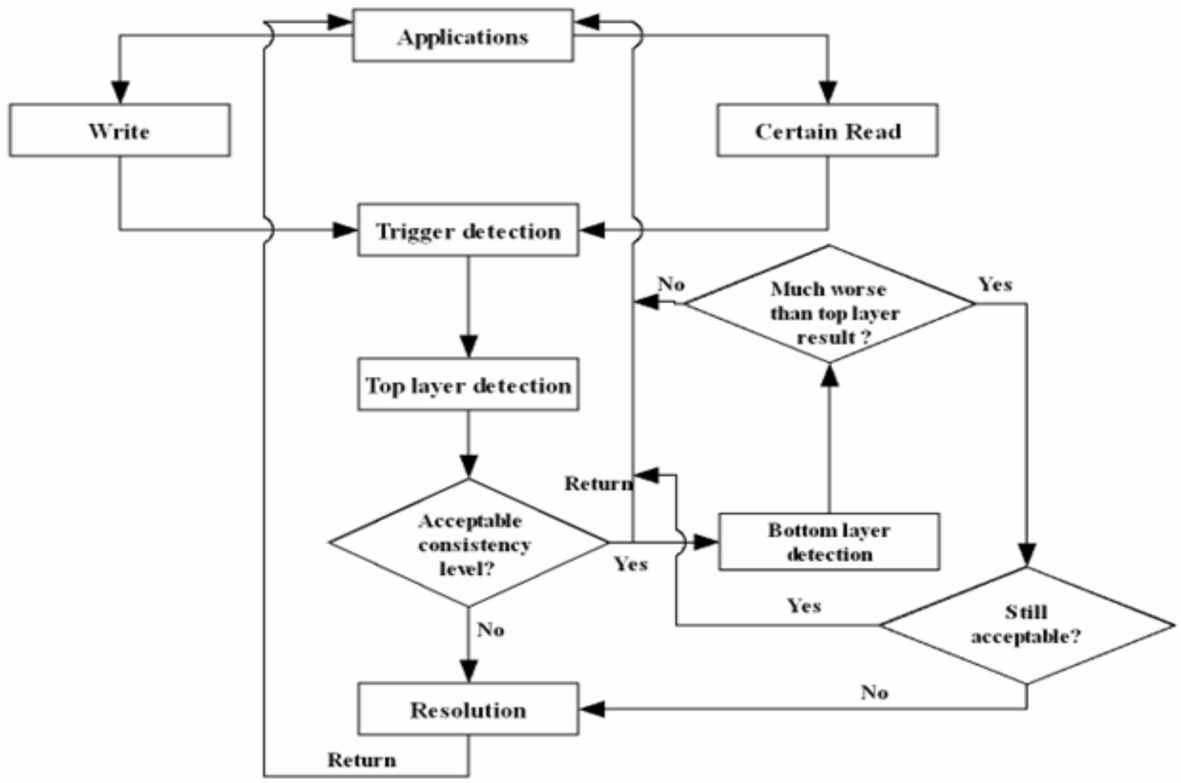
- Upon detection, the system consults with an inconsistency level monitor.

- Based on applications' senatics, if the inconsistency is tolerable or even preferable, the system does not react.

- Otherwise, the system informs the inconsistency resolution module for inconsistency resolution.

## The IDEA Protocol

### Workflow of IDEA




## A scenario of using IDEA

- A user predefines or hints an acceptable consistency level

- With certain operations, IDEA retrieves a copy and checks the copy's consistency level

- IDEA will keep the consistency level above user's predefined levels all the time to avoid annoying user

- User can communicate with IDEA about when and why a consistency level is not sufficient. IDEA will then adjust the user's acceptable consistency level accordingly.

## Adaptive consistency control

- Three possible application types that can benifit from IDEA.

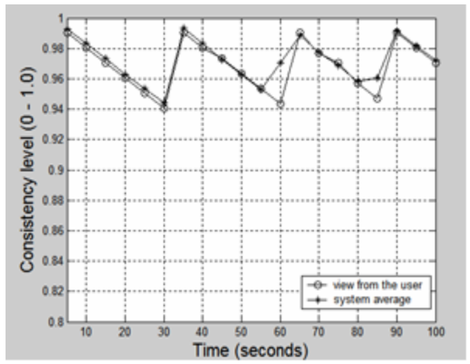
(a) **On-demand**: users explicitly requrest inconsistency resolution when needed.

(b) **Hint-based**: users need to give a hint level about acceptable consistency. IDEA maintains consistency above the hint level.

(c) **Fully automatic**: IDEA improves consistency with best effort.

## Evaluation

- Evaluate the adaptability
(1) Emulate a distributed white board on Planet-Lab.
(2) Use hint-based apprach and set the hint level as 95%.
(3) As shown in the following figure, the consistency level is improved right after IDEA kicks in.




- Speed of inconsistency resolution
(1) less than one second with up to ten simultaneous writers
(2) this speed can be further improved with simple parallesim in the resolution process.

- Communication cost
(1) incurs minimal communication cost
(2) can be supported by dial-up connections

## Discussion

- Other possible adaptive control, in addition to the three presented here, schemes?
- Ways to improve the speed of inconsistency resolution?

## Future work

- Investigate the implications of IDEA by deploying it to other distributed applications

## References

[1] Lu, Y., and Jiang, H. A framework for efficient inconsistency detection in a grid and Internet-scale distributed environment, In *Proc. of HPDC-14*. Research Triangle Park, NC, July 2005, pp. 318-319.
[2] Lu, Y., Jiang, H., and Feng, D. An efficient, low-cost inconsistency detection framework for data and service sharing in an Internet-scale system. In *Proc. of IEEE ICEBE 2005*, Beijing, China, Oct. 2005. pp. 373-380.
[3] Lu, Y., Lu, Y., and Jiang, H. IDEA: An infrastructure for detection-based adaptive consistency control in replicated services, *Technical Report TR-UNL-CSE-2007-0001*, University of Nebraska-Lincoln, Jan. 2007.