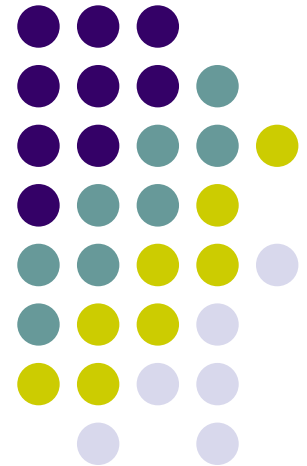


Protocols and a Framework for Intrusion-Tolerant Applications

HariGovind V. Ramasamy
Adnan Agbaria
William H. Sanders

PERFORM Research Group
<http://www.perform.csl.uiuc.edu>
University of Illinois at Urbana-Champaign, USA

Rapid-Fire Session
ITI Workshop on Dependability and Security
Dec 3, 2004





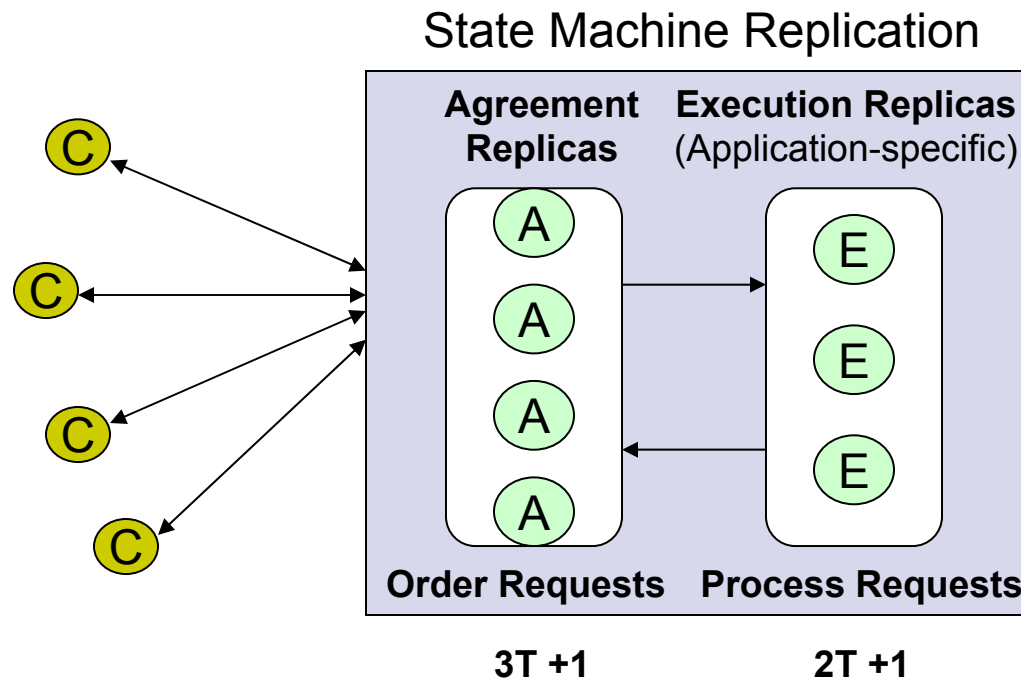
Motivation and Overview

- Intrusion Tolerance (IT)
 - providing “acceptable” service despite intrusions
- Target Applications
 - availability, security, reliability *and* operating costs are concerns
 - e.g., web services based on Application Service Provider (ASP) model
- A toolkit for IT consisting of
 - Byzantine fault-tolerant state machine replication protocols
 - Component-Based Framework for Intrusion Tolerance (CoBFIT)

Byzantine fault tolerant (BFT) State Machine Replication (SMR)



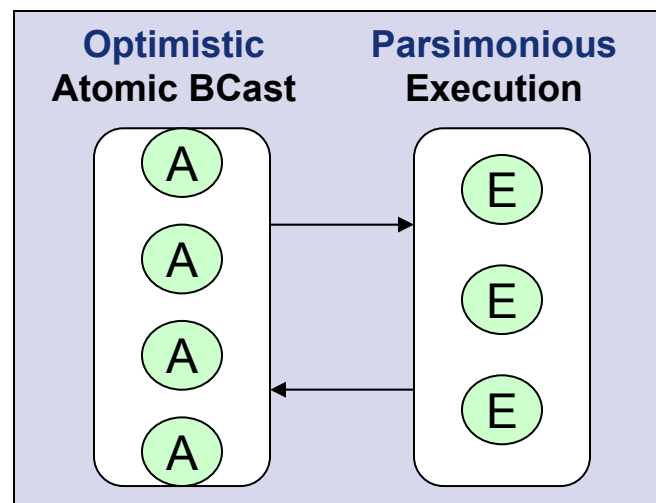
- BFT SMR for high availability, security, reliability





Our BFT SMR Protocols

- ASP business model: Usage-based pricing
 - (Resource usage in each replica) X (# replicas)
- Usual design to withstand some expected worst-case # of simultaneous corruptions, T
- $F < T$, (maybe $F=0$) for **most** of system run-time
- Our approach: optimistic & parsimonious protocols
 - minimum costs normally
 - Temporarily higher, yet acceptable overhead when new faults discovered





Our BFT SMR Protocols (contd.)

- Weak assumptions, strong adversary
 - Group Management Protocols
 - intrusion/failure detection
 - admission control
 - membership agreement
 - reconfiguration of rest of system
 - state transfer
 - without opening an avenue for DoS attacks
 - without relying on them for progress of other protocols
- (joint work
with C. Cachin,
IBM Research Labs,
Zurich)



CoBFIT Framework

- Using Byzantine fault model for **protocol design**
 - convenient for malicious attacks
 - don't have to worry about specific attack types
- Actual **protocol implementation**: different story
 - have to worry about specific attacks, intrusions
 - e.g., does the implementation tolerate buffer-overflow attacks?
- Many commonalities in support needed for robust implementation of BFT protocols
 - tight coupling in existing implementations
- CoBFIT Framework
 - Identify and isolate common support primitives, abstractions needed for BFT protocol implementations
 - Implement the support in reusable, reconfigurable, portable manner



Status

- Completed
 - Prototype CoBFIT framework
 - Parsimonious Execution Protocol
 - Reduces redundant processing upto half
 - Prototype of group management protocols
- Current work
 - Optimistic atomic broadcast protocol
 - Performance measurements