Distributed Systems Debugging

Jiří Patera

University of West Bohemia in Pilsen
Czech Republic

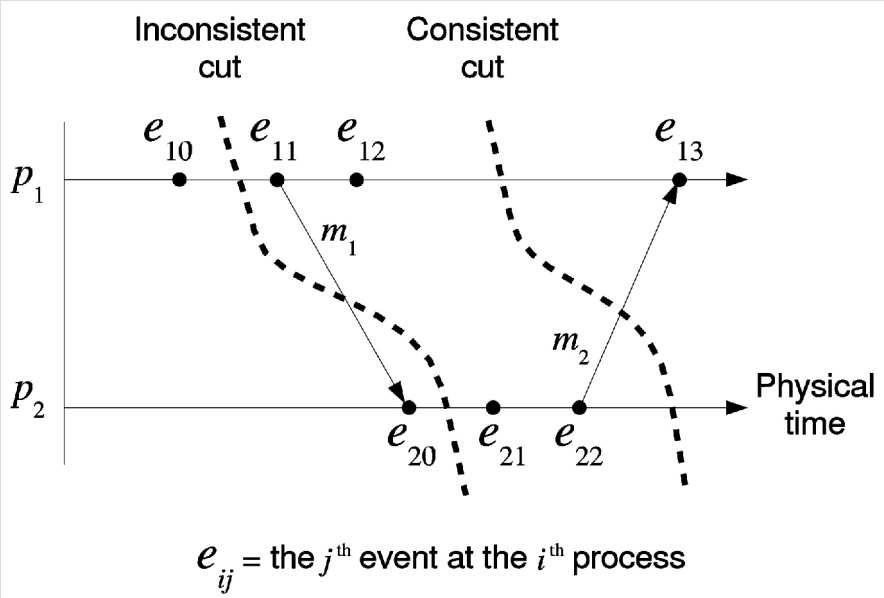
Overview

- Introduction
- Model of a distributed computation
- Two main approaches to distributed systems debugging:
  - Model checking
  - Record/Replay
- Distributed Java
- Further Work

Introduction

- What is a ***distributed system***?
  - A system with components that must co-operate on some task and which are distributed over a private or public network.
- Drawbacks of such a system are:
  - It does not have a ***global time***
  - It does not have a ***global state***

Consistency



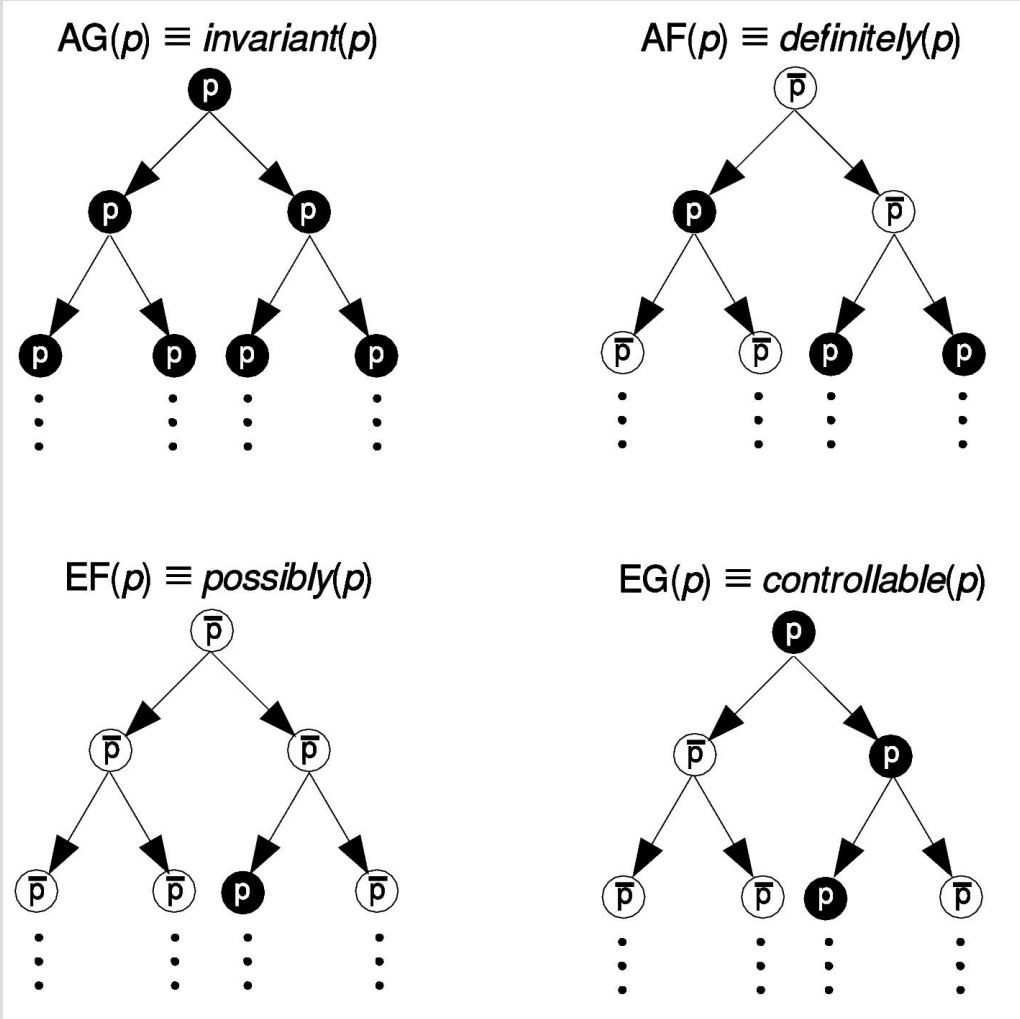
- Consistent cut
- Inconsistent cut
- Orphan messages
- Consistent state

Predicates

- Detecting a condition (***true***/***false***) in a distributed system equals to evaluating a ***global state predicate***.
- The truth value of a ***local predicate*** depends on the state of a single process.
- Four operators are used for predicate detection (possibly, definitely, controllable, and invariant).

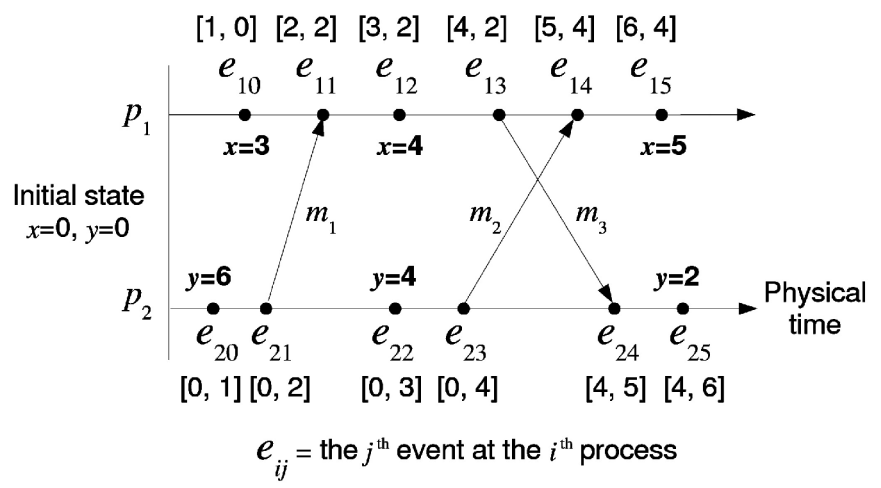
Predicates and CTL*

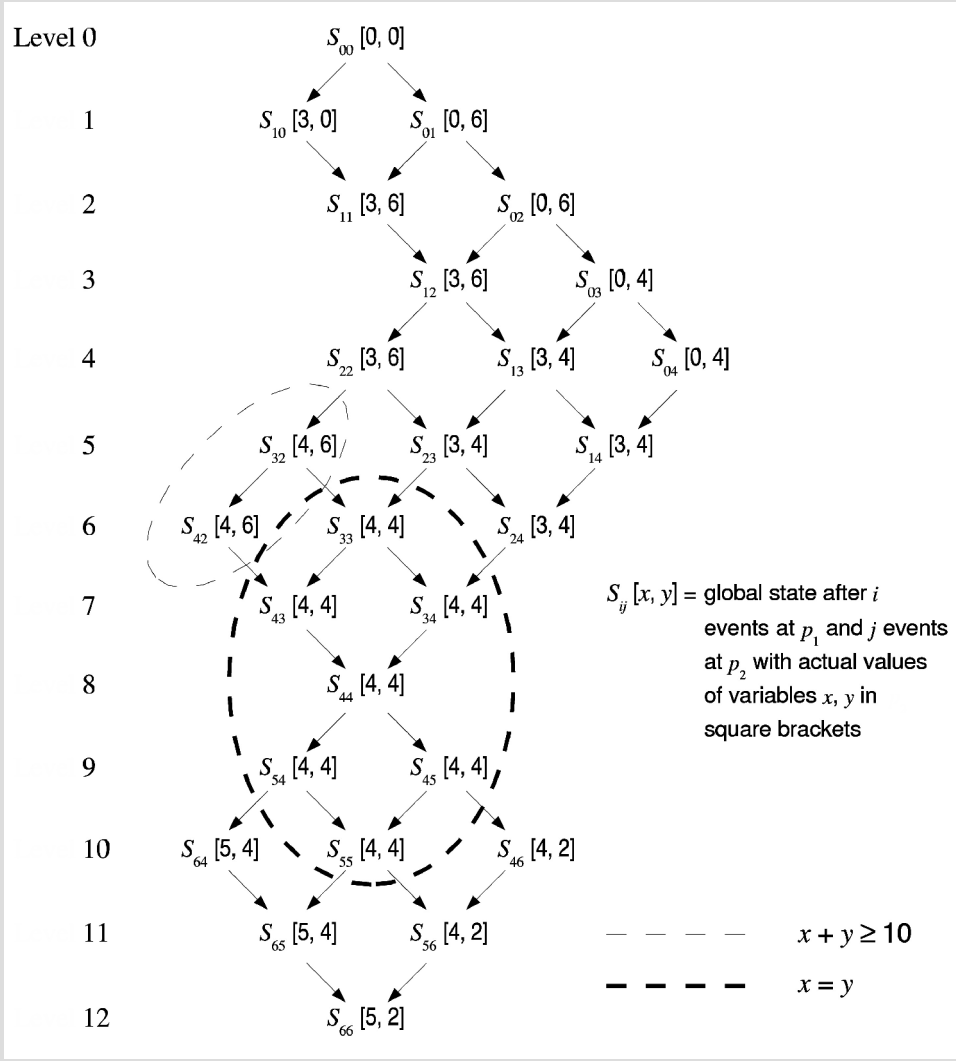
- The picture shows the connection of predicates and CTL* temporal logic operators.



Global State Lattice

- Composed of consistent global states only





Model of Observation

- The detection of a global predicate is divided among a ***checker*** and ***non-checker processes***.
  - Non-checker processes send messages to the checker. They are situated at computation nodes and monitor local variables and messages.
  - The checker process searches for a consistent cut that satisfies a predicate.

Monitoring Approaches

- Observation = Monitoring
- The three monitoring approaches:
  - Hardware
    - Expensive, large amount of data, no probe-effect
  - Software
    - Cheap, prone to probe-effect (can be eliminated)
  - Hybrid
    - Mixture of hardware and software

Model of Control

- Every process has associated with it a ***supervisory process***.
  - This process observes the underlying process and controls it by delaying, disabling, or by changing the order of incoming and outgoing messages.
- Useful for testing under some desired conditions.

Replaying a Distributed Execution

- Two possible ways of doing record/replay:
  - ***Deterministic replay***, ***incremental replay***.
- The amount of recorded information depends on how it will be used.
  - ***Browsing*** (minimal information)
  - ***Replay***
  - ***Simulation*** (maximal information)

Summary

- The formal checking approach
  - Advantage is that it checks all possible ways of execution.
  - Drawback is that its complexity grows exponentially with the number of processes.
- The record/replay approach
  - Advantage is that it is not as complex as the previous one.
  - Drawback is that it checks only one run of many possible.

Distributed Java

- Java is a programming language suitable for distributed heterogenous environments.
- However, Java debuggers do not fully support applications distributed across multiple JVMs.

Java Drawbacks

- There are two types of distributed debuggers for the Java language:
  - The first type is based on a modified JVM on each of the co-operating nodes.
  - The second one is based on compiling the source codes while adding DSM capabilities to the final application.
- Advantage: standard programming.
- Drawback: no cross-platform portability.

Further Work

- Detailed investigation and analysis of approaches to the debugging of distributed Java programs.
- Proposal of a new method for the debugging of distributed programs, a method which is based on a modification of the Java bytecode.
- Experimental implementation and testing of the proposed method.

Thank You

Thank you for your attention...

Now it is time for your questions.