Distributed Model Verification Using Map-Reduce

Presented By
Fathiyeh Faghih

March 2010
Project Objective

- Using cloud computing in software verification
- Saving time, especially for big software!
### Table of Content

- Background of Certification
- Parallel Certification
- Implementation Details
Table of Content

- Background of Certification
- Parallel Certification
- Implementation Details
Model Checking

- Background of Certification
- Parallel Certification
- Implementation Details

Ref: CS745 Course Notes, Nancy Day
Component-based Software Engineering

- Assembly of pre-existing components

- Components’ correctness

- Background of Certification
- Parallel Certification
- Implementation Details
Component-based Software Engineering

- Component producer
  - Non-trusted
    • Does it work as advertised?
  - Trusted
    • Assessing additional properties
Search Carrying Code (SCC)

- Help the code consumer to do model checking
- Recording the search path in search scripts

70% saving in the model checking time
Search Carrying Code (SCC)

- Trustful certification
  - Checking additional properties

- Tamper proof certification
  - Correctness of the search script
Table of Content

- Background of Certification
- Parallel Certification
- Implementation Details
Search Carrying Code

- Parallel model checking
  - Huge number of states
- Known reachability graph
- Statically partitioning a search script
Parallel Certification

- Trustful
- Tamper proof
Parallel Certification

- Trustful
- Tamper proof
Parallel Trustful Certification

- Background of Certification
- Parallel Certification
- Implementation Details

Input

Map

Map

Map

Map

Map

Reduce

Reduce

Output
Parallel Trustful Certification

- Background of Certification
- Parallel Certification
- Implementation Details

Distributed Model Verification Using Map-Reduce

**Input**

```
Map
Map
Map
Map
Map
```

**Output**

```
Reduce
Reduce
```
- Partitions of the search script

- Forcing the Map-Reduce not to partition the input files
Parallel Trustful Certification

• Background of Certification
• Parallel Certification
• Implementation Details

Input

Map

Map

Map

Map

Reduce

Reduce

Output
- Model checking of the partition

- Java Pathfinder
Reducer Function

Input

Map

Map

Map

Map

Reduce

Reduce

Output
Reducer

- Checking whether the program is certified or not
- Producing counterexamples
Parallel Certification

- Trustful
- Tamper proof
Parallel Tamper Proof Certification

- Background of Certification
  - Parallel Certification
  - Implementation Details

![Diagram](image)

1. **Partition**
2. **Mapper**
3. **Reducer**
4. **<stateID, FP>**
5. **<state1, FP1>**
6. **<state1, FP2>**
7. **<state1, FP3>**

Flow: Partition → Mapper → Reducer → <stateID, FP> → FP1=FP2=FP3?
Table of Content

- Background of Certification
- Parallel Certification
- Implementation Details
Implementation Details

- Karmasphere Studio
  - Plugin to NetBeans IDE
  - *Develop and debug* MapReduce jobs on your desktop
  - *Deploy* jobs to remote Hadoop clusters

- Trustful certification
  - Testing the deployment on the desktop
Future Steps

- Deploying the application on Amazon
  - Creating images with JPF installed and configured

- Using the JPF suitable for tamper proof
  - Modifying the reducer
Evaluation

- The time needed to do certification vs. the time needed for model checking
  - Trustful
  - Tamper proof

- Experimenting on different numbers of partitions
  - Optimal number for each number of states?
  - Independent of the shape of the graph?