Challenges to Error Diagnosis in Hadoop Ecosystems

Jim Li, Siyuan He, Liming Zhu, Xiwei Xu, Min Fu, Len Bass, Anna Liu, An Binh Tran
About NICTA

National ICT Australia

- Federal and state funded research company established in 2002
- Largest ICT research resource in Australia
- National impact is an important success metric
- ~700 staff/students working in 5 labs across major capital cities
- 7 university partners
- Providing R&D services, knowledge transfer to Australian (and global) ICT industry

NICTA technology is in over 1 billion mobile phones
Problem

• Operator invokes some process in cloud (e.g. rolling upgrade or installation)
• 45 minutes or an hour later – the process fails
  – Usually with an error message
  – Possibly with a silent failure that manifests itself much later
• Operator must then diagnose failure
• Problem is most complicated when multiple components are involved.
But aren’t there tools and recipes?

• Yes – but …

• Recipes for deployment tools make assumptions about what you want.

• In many cases, these assumptions are wrong.

• In these cases, you must troubleshoot installation problems.

• Troubleshooting is based on examination of generated logs.
What are the difficulties associated with using logs?

• The system being deployed is an ecosystem with multiple independently developed systems. Each component’s logging is independently determined and not under central control.
  – Events and state deemed worthy to log may be different from different components

• Results in
  – Sequence of events leading to failure may be difficult to reproduce
  – Missing or contradictory information in combined logs
Our envisioned deployment solution

• A solution will
  – Execute the correct steps in a correct order
  – The execution of a step will result in a correct state of the environment

• Use a process model annotated with assertions to detect incorrect steps or incorrect state

• The detection of an error will trigger a look up in a repository that maps symptoms to fault trees to root causes.
Rolling upgrade process model example

- Attach assertions to process model to test state
- Use progress within process to determine which assertions to test.
- This approach restricts root cause determination to particular step in the process.
This paper

- Makes a contribution to the envisioned repository
  - Present 15 examples of systems/possible root causes for Hbase/Hadoop deployment

- Provides a classification of errors into
  - Operational
  - Configuration
  - Software
  - Resource

- Identifies specific error diagnosis challenges in multi-layer ecosystems.
What did we do?

• We manually deployed HBase/Hadoop on EC2
  – 5 NICTA people from 2 different groups
  – 10 installations in total

• We diagnosed and recorded errors we discovered
  – With help from a Citibank person
Case study

Hbase Cluster on Amazon EC2
Sample Errors - 1

• Source – HDFS
• Logging Exception: “DataNode is Shutting Down”
• Possible Causes/diagnostics
  – Instance is down/ping ssh connection
  – Access permission/check authentication keys, ssh connection
  – HDFS configuration/check “conf/slaves”
  – HDFS missing component/check data node settings and directories
Sample errors - 2

• Source: Zookeeper
• Logging exception: “java.net.UnknownHostException”
• Possible causes/diagnostics:
  – DSN/check DSN configuration
  – Network connection/check with ssh
  – Zookeeper configuration: zoo.cfg
  – Zookeeper status/processes (PID and JPS)
  – Cross-node configuration error/check consistency
Comments on Errors

• Paper has
  – 15 enumerated exceptions and potential causes
  – Discussion of classification of errors and samples

• Most useful to non-expert installers

• Information could potentially be found on
  – Stack Overflow
  – Specific source forums

• Better to have
  – Consistent form for fault trees
  – Known place to find them
  – Standard environmental description
Different types of errors

• Operational errors
  – Start up/shutdown errors
  – Artifacts not created or created incorrectly

• Configuration errors
  – Syntactic errors
  – Cross system inconsistency

• Software errors
  – Compatibility errors
  – Bugs in the software

• Resource errors
  – Resource unavailability or exhaustion
Challenges to trouble shooting from logs

• Inconsistency among logs
• Signal to noise ratio
• Uncertain correlations
Inconsistency among logs

• IP address is used as ID but IP addresses can change in the cloud. For example, if an instance is restarted.

• Inconsistent time stamps in a distributed environment due to network latency makes determination of a sequence of events difficult.
Signal to noise ratio

• Logs contain huge amount of information
• Tools exist to collect logs into a central source
  – Scribe
  – Flume
  – Logstash
  – Chukwa
• Tools that search logs need guidance to filter information
• We propose an approach that uses a process model to guide diagnosis (to be explained shortly).
Uncertain correlation

• Between exceptions
  – Connections among exceptions arising from the same cause are difficult to detect.

• Between component states
  – Dependent relations among component states not shown in log messages and are difficult to detect.

• Between events
  – Connections among distributed events are difficult to detect.

• Between states and events
  – Diagnosis depends on connecting state and events and these may not be obvious from log messages.
Summary

• Deploying or updating ecosystems is an error prone activity

• Determining root cause of an error is difficult and time consuming

• We provided a list of 15 specific errors and their potential root causes for Hbase/Hadoop deployment

• We categorized types of errors and uncertainties in error diagnosis