Challenges in Long-Term Logging and Tracing

Ian F. Adams and Ethan L. Miller University of California, Santa Cruz

The Big Picture

- We're good at capturing data on system activities
- We're not so good at maintenance of this data over the long-term

So? Who Cares?

- Some behaviors aren't apparent in the short term
- Long-term data can be massive in volume and challenging to work with
- Loss or corruption of long-term data can be much more difficult to deal with compared to the shorter term

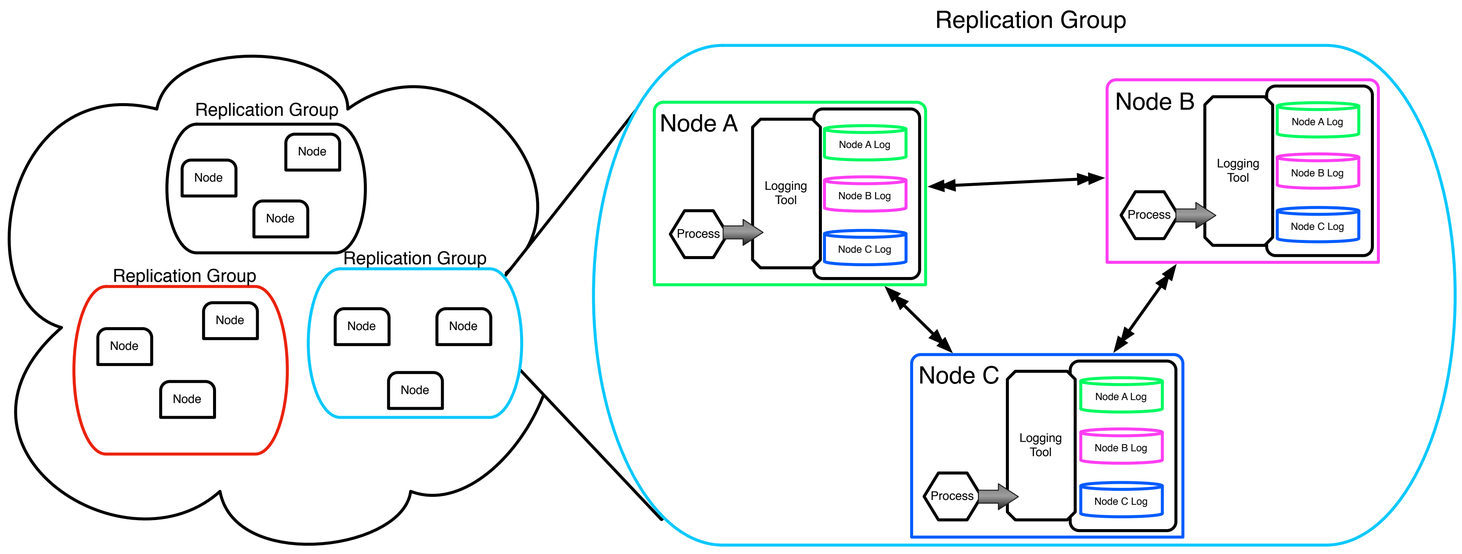
What Do We Do?

★ We use our experiences in long-term log analysis to identify several critical problem areas, and propose ways to address them within a logger framework

Reliability

"Backups? Oops...."

- With multi-year timespans, redundancy and consistency checking is a must



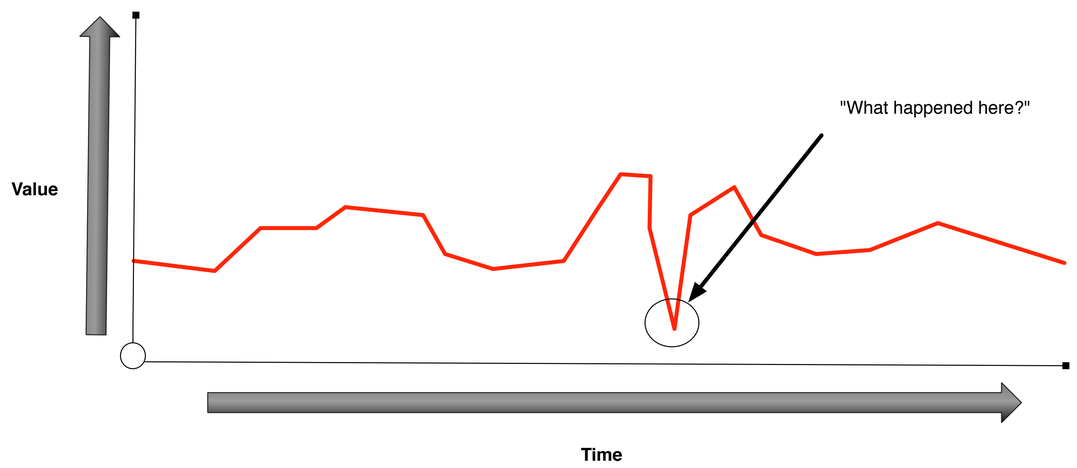
★ Maintain reliability with distributed replicas in *replication groups*

- ▶ Leverage replicas for annotations and failure notes

Noting Absence

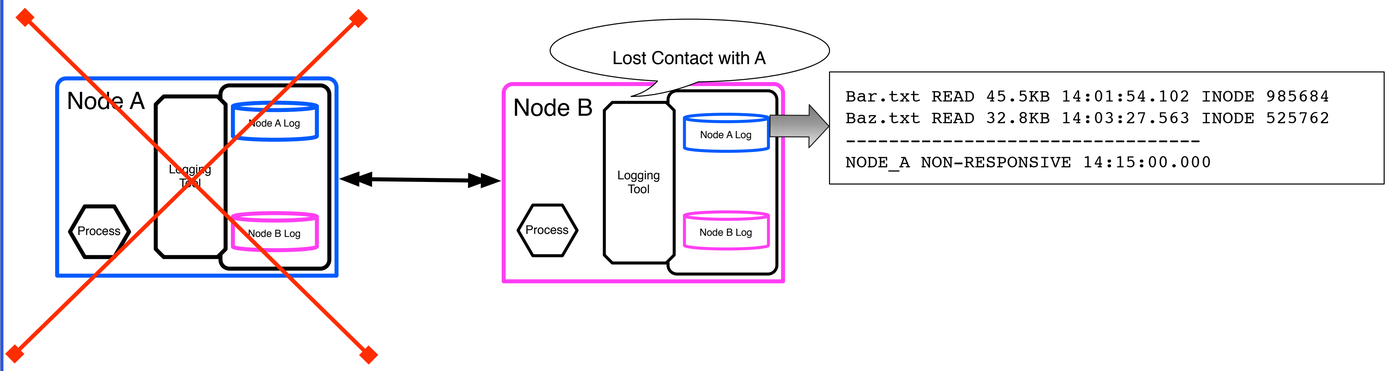
"Wait...what the heck happened here?"

- Is a reduction in logged activity *actually* due to less activity, or is a process, logger or node down?



★ When a node notices a replica or log it manages hasn't received an update or heartbeat recently from a node or process, note it in the relevant logs

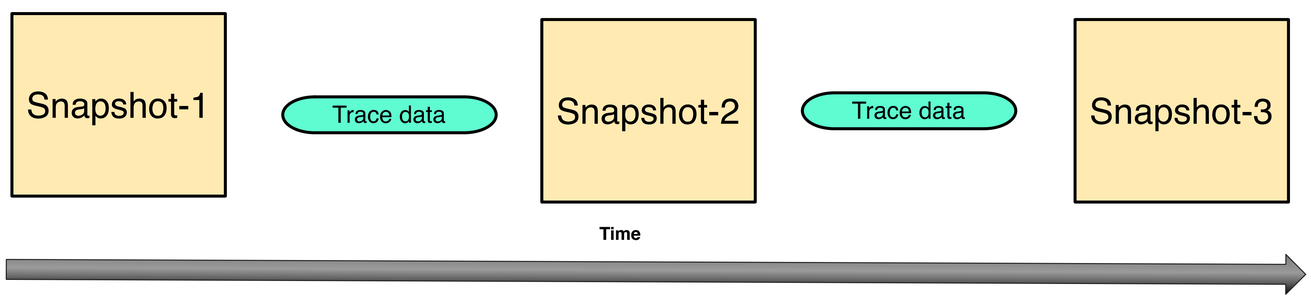
- ▶ If the other node comes up, note the return and later merge the logs
- ▶ This can aid in understanding the nature of the activity drop



System State

"I'm hope that part was optional..."

- Without periodic pictures of the system state, one's understanding of what is really occurring is degraded
  - ▶ For example, one can't tell the fraction of a directory accessed without knowing the start state of a system



★ Periodically take snapshots of total system state, in a storage system this could be a filesystem crawl.

- ▶ We make more accurate statements about the nature of activity, as well as answer questions we couldn't with only a trace or snapshot(s)
- ▶ We can also understand the 'coverage' of a trace by using a trace as a delta on top of a snapshot and comparing the result to a second snapshot

Tracing Resolution

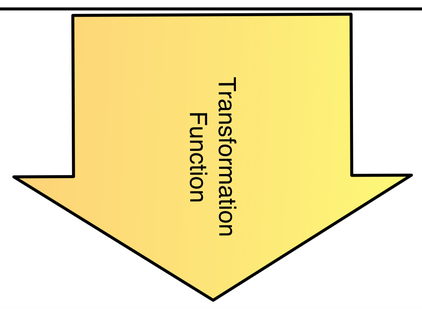
"Too. Much. Data."

- Even low rates of data growth can lead to extremely large datasets if kept for years on end
- Large amounts of data can also make it hard to work with and analyze
  - ▶ We don't all have 'Google' level resources

★ Periodically transform logged data to the desired granularity

- ▶ Leave 'interesting' events at original granularity if desired

```
Foo.txt READ 45.5KB 10:31:24.102 INODE 585684
Foo.txt READ 45.5KB 10:33:26.563 INODE 585684
Foo.txt READ 45.5KB 10:36:29.985 INODE 585684
Foo.txt READ 45.5KB 10:45:35.999 INODE 585684
Foo.txt WRTE 45.5KB 10:52.27.059 INODE 585684
```




```
10:30:00-10:59:59
Foo.txt 4 READ
Foo.txt 1 WRTE
```

Format Shifts and Logger Hiccups

"I'm sure they'll figure it out eventually..."

- A common problem is small changes in the format of logged data
  - ▶ Strange logger hiccups occur often as well...
- These issues can often be difficult to catch and diagnose
  - ▶ This can break parsers and/or silently corrupt analyses

★ Have the logger periodically check for format consistency