OTUS: RESOURCE ATTRIBUTION IN DATA-INTENSIVE CLUSTERS

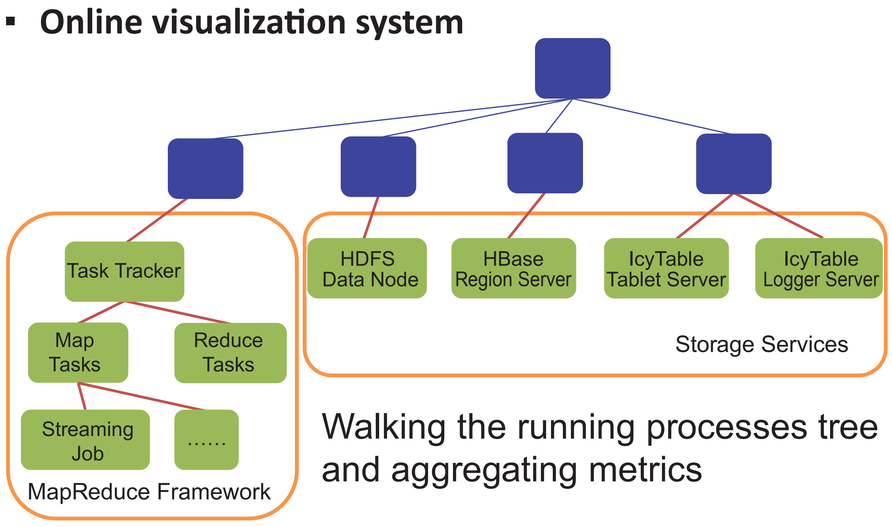
Kai Ren, Julio López, Garth Gibson (CMU)

MOTIVATION

- Debugging & tuning large distributed systems is challenging
- Limitations of current monitoring tools
  - Log-based tools (e.g. Chukwa, Mochi)
    - Lack OS level metrics
    - Do not show system resource utilization
  - OS-based tools (e.g. Ganglia)
    - Provide gross OS information, lack notion of MapReduce (MR)
    - Host-wide: they do not distinguish MR processes from others
- "Otus": a genus of owls with superior vision and hearing in the dark
- Key ideas:
  - Correlate resource metrics of specific MR jobs and services
  - Allow users to see utilization according to user concepts: my job / HDFS process, Hadoop process, etc.

APPROACH

- Collect fine-grained MR-related OS metrics from /proc
  - Walk tree of running processes looking for specific command lines
  - Aggregate stats of all children processes for MR tasks
- Current Implementation status
  - Daemon collects metrics from MR Framework and /proc file system
  - Store metrics into OpenTSDB (using HBase as storage backends)
  - Online visualization system

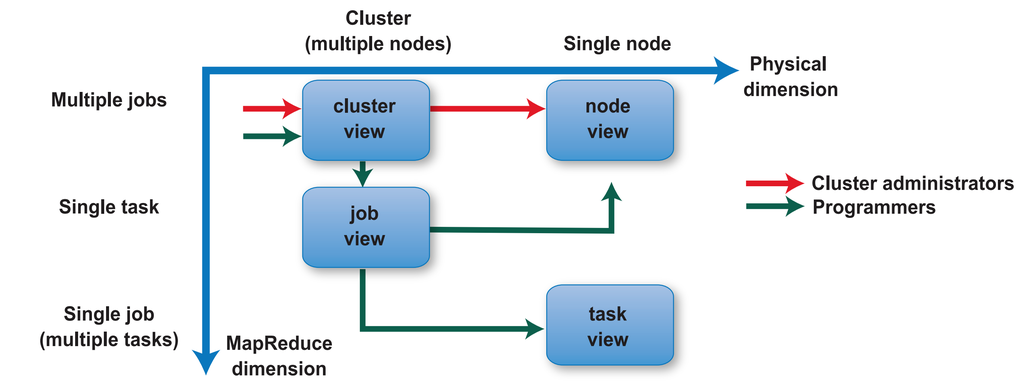


Walking the running processes tree and aggregating metrics

MULTI-VIEW VISUALIZATION

VIEW ZOOMING MODEL

- Administrators: understand resource utilization of MR jobs on a cluster/node
- Programmers: reason about their Hadoop workflow

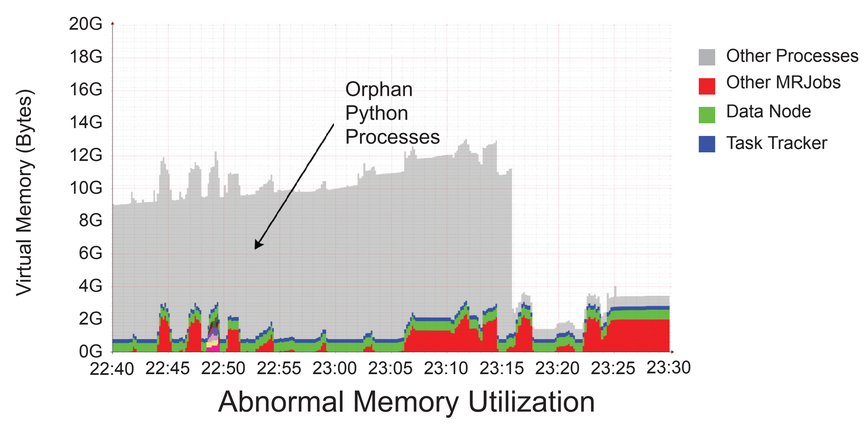


Intel Science & Technology Center for Cloud Computing

CASE STUDIES

RESOURCE EXHAUSTION IN CLUSTER HOSTS

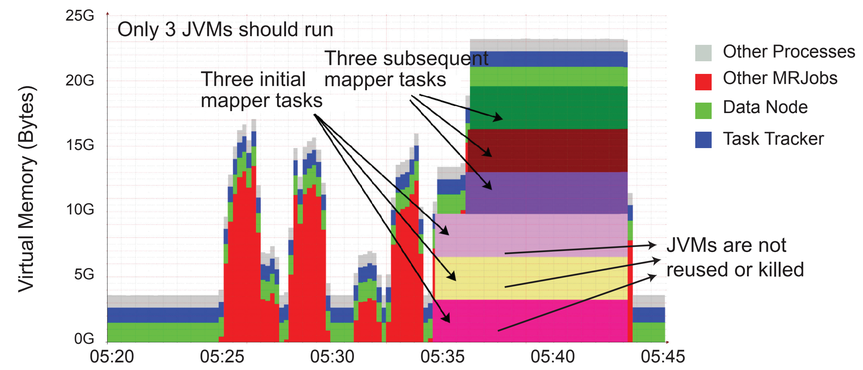
- Symptom: Hosts crashed with moderate CPU and high memory usage
- Debugging: Otus shows memory consumed by processes other than MR job or framework processes
- Root cause: Orphan Python processes spawned by Hadoop streaming jobs were not properly cleaned up



Abnormal Memory Utilization

DISCOVERING PITFALLS IN JVM REUSE

- JVM Reuse: MR framework allows JVMs from finished tasks to be used by new tasks of the same job (to shorten time for starting new JVMs)
- Pitfall: JVMs were not immediately reused nor terminated. These JVMs consumed available physical memory and triggered process failures as they could not allocate memory



Pitfall in Memory Management

FUTURE WORK

- Design automatic strategies to detect abnormal behaviors and performance problems in large clusters
- Disaggregate requests to storage services to attribute to specific MR jobs

CONCLUSIONS

OTUS

- Non-intrusive, distributed monitoring and visualization system
- Attribute resource utilization information to user jobs and cluster services
- Provides multi-view visualization to aid performance analysis
- Code Release: https://github.com/otus/otus
- Known users: CMU OpenCloud & CMU Qatar Cloud Research

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley