PO.DAAC DMAS—Dynamically Scalable Job Management

Michael Gangl, Nga Chung, Christian Alarcon & Thomas Huang

Jet Propulsion Laboratory, California Institute of Technology
Copyright 2012 California Institute of Technology. Government sponsorship acknowledged.

DMAS (Data Management Archive System)
- DMAS handles ingestion and cataloging of data and its metadata, facilitating search and retrieval of physical oceanography data.
  - What’s a dataset?
    - “A logically meaningful grouping or collection of similar or related data. Data having mostly similar characteristics (source or class of source, processing level and algorithms, etc.)” [1]
  - What’s a granule?
    - Data file + ancillary files (checksums, images) consisting of a range of measurement (spatially and/or temporally)
    - Ranges from 2-6 files in nominal cases

Stats and info
- 800+ operational datasets
- ~2150 granules/day for 2012
- As many as 9225 granules in a single day (Varying usage)

DMAS: Before

Manager Architecture

DMAS: Before, cont

• Nominal operations were great, but what about…
  • Reprocessing of granules
    • Data provider updates software and sends all of their data back over using the new algorithms
  • Reingestion of old data
    • Existing data needs to be reingested to capture new desired metadata
  • Ingesting legacy data
    • Thousands of granules we want to migrate from our old system to our new system
  • Manager can’t scale infinitely
    • Manager can take on new workers, but not an ‘infinite’ amount
Enter Zookeeper

• What is it
  • “ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications.” [2]
  • Apache Project

• Main features:
  • Scalable (dynamically)
  • Redundant
  • Hardened by commercial users

• Simple building blocks to create more complex structures
  • Queues, Priority Queues
  • Synchronization
  • Lockable resources


Who Uses Zookeeper?

• Netflix
• Yahoo
• Zynga games
• Rackspace
• Hadoop
Zookeeper and DMAS

- Zookeeper is the coordination layer between Manager(s) and Worker(s)
- Managers don’t know about workers, workers don’t know about managers
- A worker processes a single job, and then immediately asks for another. Jobs are not queued up for a specific worker.
- Add and remove new managers, workers dynamically
  - Manually add/remove if we know what is needed
  - Create rules that automatically scale when certain thresholds are met
    - Number of queued jobs
    - Average granule processing time
Priority Queue

- Queue system
  - FIFO structure
  - Can add or remove engines dynamically to speed up work

- Priority Queuing
  - Give preference to certain datasets (we set this explicitly, not algorithm/heuristic based)
    - Some datasets need to be made available 2 hours after they are created
    - Don’t want them to compete with lower priority jobs
  - Simple change to the Queue mechanisms allows for this prioritization
  - Multiple ways of processing the priority queue
    - One queue, all jobs assigned to it (starvation for low priority)
    - Multiple queues with explicitly assigned workers for each

Priority Queue, cont

- Workers allocated to each Queue type
  - 2 machines dedicated to processing low
  - 4 machines dedicated to processing medium
  - 8 machines dedicated to processing High
  - Workers work solely on queue to which they are assigned
When do you add more resources?

- Home brewed Monitoring tools
  - Decision making support

Lessons Learned

- Monitoring zookeeper
  - Visualizing this data isn’t trivial
    - Homebrew tools to monitor job times, throughput
  - More tools are available now (top, dashboard, latency test)

- Error scenarios become harder to conceptualize
  - Manager doesn’t care about workers, but it really, *REALLY* cares about the jobs.
    - Can we recreate the ‘job’ if it gets lost somewhere?
  - What if an engine fails as soon as it removes a job from the queue?

- Priority queue
  - One large pool of workers can cause starvation for low priority jobs
    - This might be ok for some organizations! (but not us)… multiple queues
Known Issues

- Zookeeper Rest Service
  - The version we adopted had limited support for restful services
  - Better in recent versions, but need to migrate

- Security
  - Basic security types are built into zookeeper
  - IP/Host filtering
  - No encryption of the traffic between clients and zookeeper… yet
    - [https://issues.apache.org/jira/browse/ZOOKEEPER-1000](https://issues.apache.org/jira/browse/ZOOKEEPER-1000)
    - Makes this a suitable solution for closed, cluster-based communications
    - Not suitable for WAN usage between geographically dispersed clusters

Looking forward

- Dynamically scaling based on load
- Cloud deployment testing
- Enhanced monitoring capability (remove, add, overwrite nodes)
- Reusing the ZK Architecture
  - Product Generation Pipeline
  - Configuration management/Naming service for distributed services
Questions

Backup Slides
Zookeeper Internals

• Persistent Connections
  • Zookeeper communicates with clients over a persistent connection, with two threads (event thread and IO thread)
    • Event thread handles callbacks
    • IO thread maintains connection (heart beats, send/receive)

• Callbacks/watchers
  • Watchers can be set on nodes to see when they change or are removed
    • Includes when a child is set on a parent node
  • Consistently ordered by the zookeeper servers
  • Allow developers to react on certain cues
    • Know when a job is removed from the processing queue (someone is working on it)
    • Knows when processing is finished (update to a job node)