

Pepper: An Elastic Web Server Farm for Cloud based on Hadoop

Subramaniam Krishnan, Jean Christophe Counio

Yahoo! Inc.

MAPRED

1st December 2010

Agenda

- Motivation
- Design
- Features
- Applications
- Evaluation
- Conclusion
- Future Work

Motivation



What's in a Name

Wave 1: Grid-ification

Crunch 10-100s of GBs of data in hours

Large data like wikipedia

Hosted, multi-tenant platform

Grid workflow management system (PacMan)



Wave 2: Content Freshness

Process 100s of feeds/sec, size in KBs in seconds

Web feeds like breaking news, tweets, finance quotes

Scalable, high throughput & low latency platform

Pepper - elastic web server farm on grid

Requirements

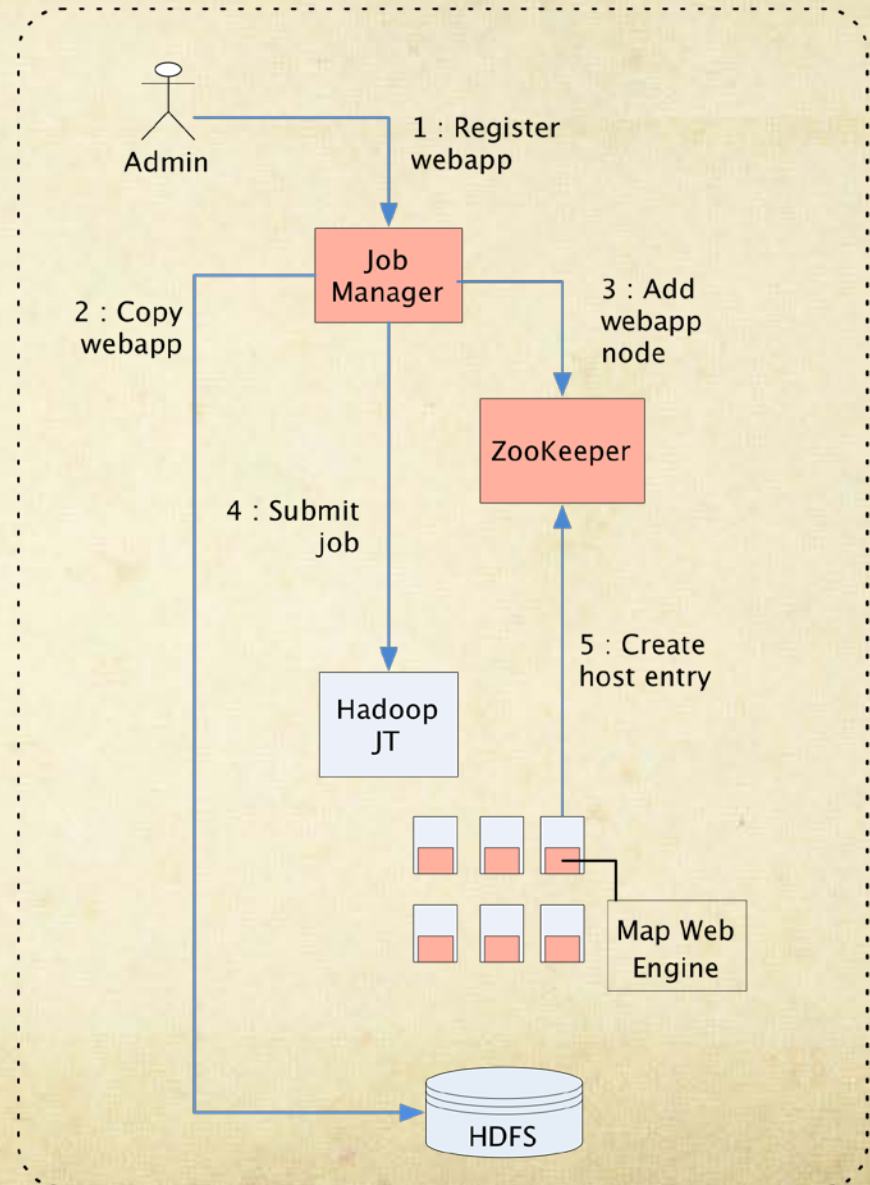
- Elastic: handle intra/inter application load variance
- Multi-tenant: provide process/memory isolation
- Sub-second platform overhead
- Simple API
- Execute user code in platform context
- Reliability: transparent fault tolerance

Design



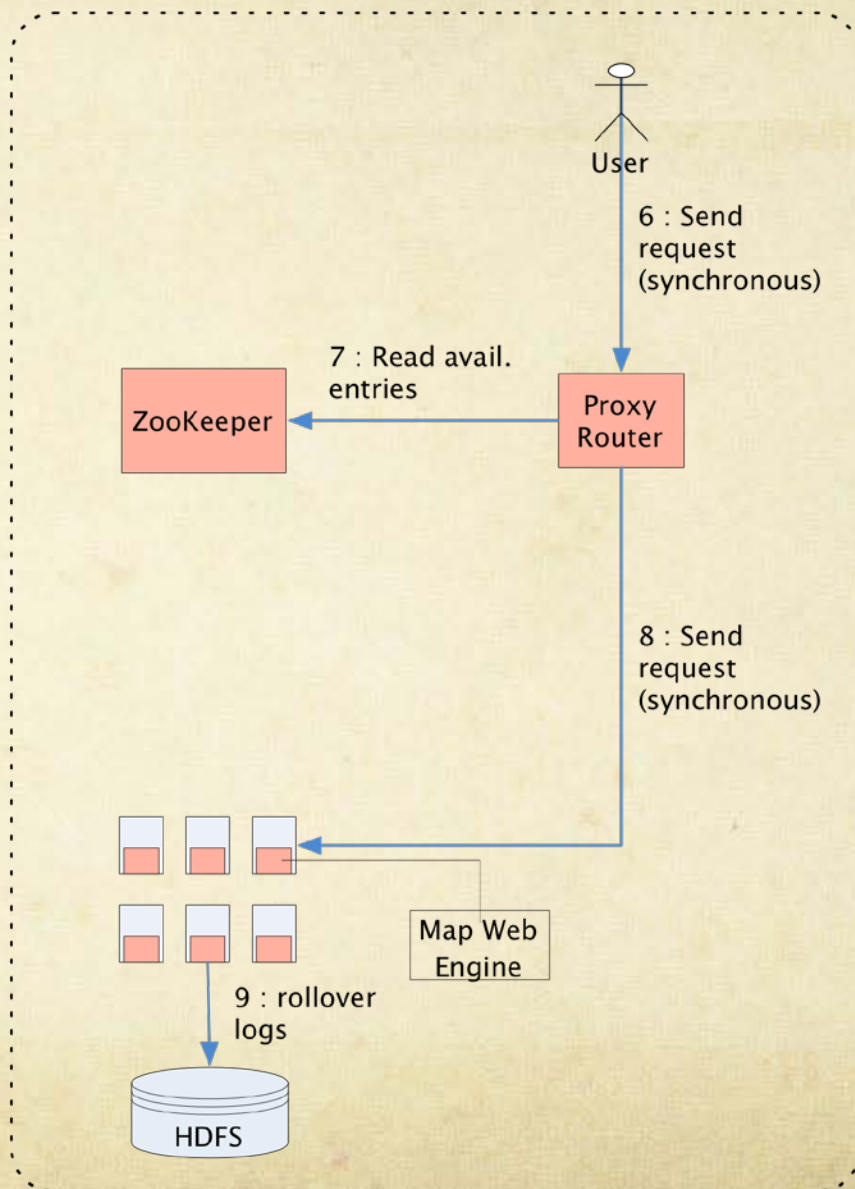
Deployment Flow

- Web application deployed as WAR onto HDFS - Job Manager
- Embedded Jetty server runs in Map task, registers with ZooKeeper
- 1 Hadoop job = 1 Map task = 1 Web Server = 1 Web application

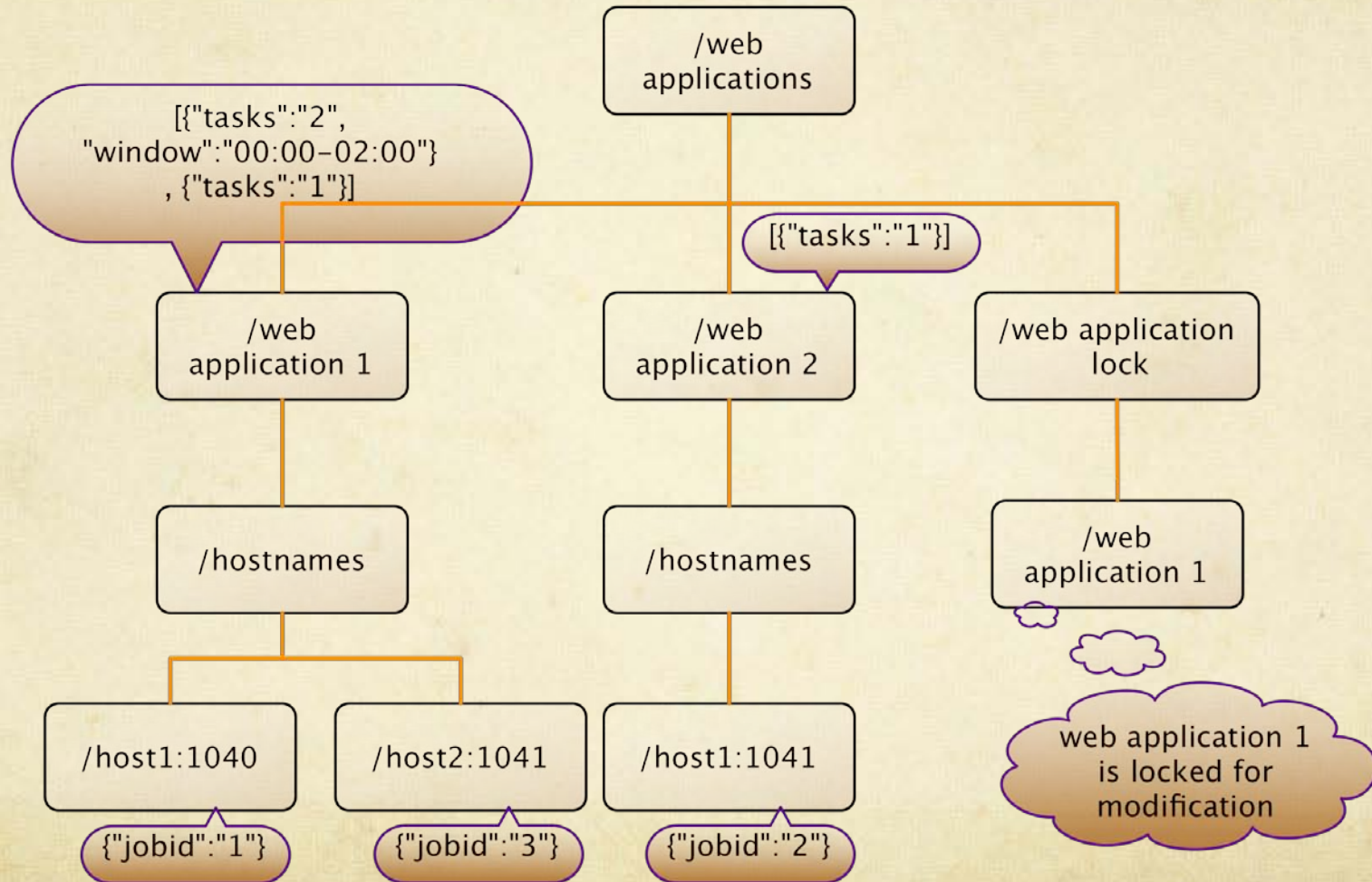


Processing Flow

- Proxy Router receives incoming requests, looks up ZooKeeper & redirects to appropriate Web Server



ZooKeeper Hierarchy



Features



Features

- **Scalability:** Web application can scale by configuring more instances (Elasticity), system can scale with addition of Hadoop nodes
- **Performance:** High throughput by ensuring that all the heavy lifting is done during deployment
- **High Availability/Self-healing:** Redundant server instances. Health check piggybacked on TaskTracker heartbeat
- **Isolation:** Hadoop map provides process isolation
- **Ease of Development:** Standard Servlet API & WAR packaging
- **Reuse of Grid Infrastructure:** The system runs on a Grid that can be shared across several applications

Applications



Applications

- **Web Feeds Processing:** Configure workflow orchestration engine to run in-memory, 1 workflow = 1 web-application.

Benefits:

- Scalability
 - Isolation
 - Avoids Hadoop job bootstrap latency and HDFS small files bottleneck.
- **Online Clustering:** Extracts features and assigns incoming feeds to clusters predetermined by offline clustering. Performed online for Yahoo! News to identify hot news clusters during ingestion of articles.

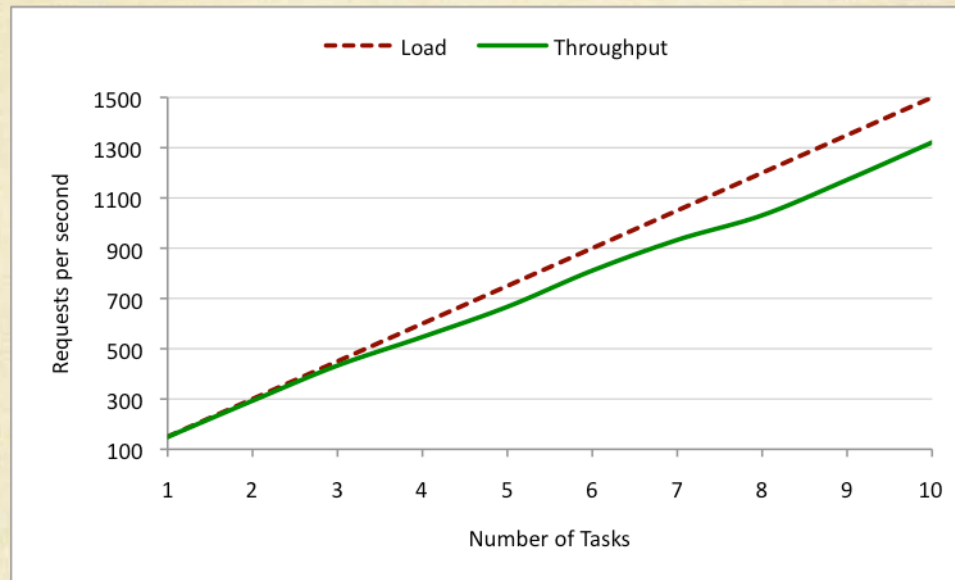
Evaluation



Setup

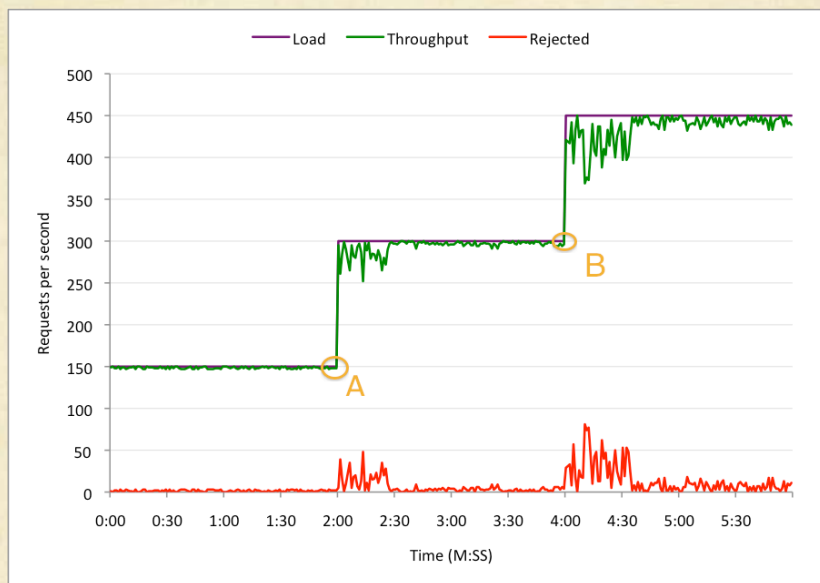
- Hardware: Intel Xeon L5420 2.50GHz with 8GB DDR2 RAM
- Software: 64-bit SUN JDK 1.6 update 18 on RHEL AS 4 U8, Linux 2.6.9-89.ELsmp x86_64
- Configuration: 8 map slots/node with 512MB heap, 25 threads/Jetty server
- Number of Computing Hadoop nodes: 3

Linear Scaling for Predefined Capacity



- Throughput: number of requests handled successfully per second for a specified number of tasks

Elastic Scaling for Dynamic Capacity



- Rejection: failure to execute within predefined timeout
- Load is increased and additional map task allocated at points A and B based on predefined schedule
- Failure rate of $< 0.001\%$ observed in Production

Pepper Performance Numbers

System	Burst Rate (request/minute)	Throughput (requests/day)	Platform Latency (Avg.)	Response Time (Avg.)
Pepper	2,000	3 million	75 ms	4s
PacMan	50	10,000	90s	120s

- Dataset is Yahoo! News feeds with sizes < 1MB
- Processing is typically computation intensive like processing and enriching web feeds that involves validation, normalization, geo tagging, persistence in service stores, etc

Conclusion



Conclusion

- Pepper marries the benefits of traditional server farms i.e. low latency and high throughput with those of cloud i.e. elasticity and isolation
- In production within Yahoo! from December 2009
- Current Y! properties - Newspaper Consortium, Finance & News. Sports & Entertainment are in pipeline
- System scales linearly with addition of more Hadoop computing nodes

Future Work



Future Work

- On demand allocation of servers
- Experimenting with async NIO between Proxy Router & Map Web Engine to increase scalability
- Improving distribution of requests across web servers
- Integrate into Hadoop (?)

References

- Hadoop, Web Page <http://hadoop.apache.org/>
- J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Cluster”, *6th Symposium on Operating Systems Design and Implementation (OSDI’04), San Francisco, CA, December 2004, pp. 137–150*
- P. Hunt, M. Konar, F.P. Junqueira, and B. Reed, “ZooKeeper: Wait-free coordination for Internet-scale systems”, *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, Boston, MA, June 2010, pp. 11- 11*
- Oozie (successor to PacMan), Web Page <http://yahoo.github.com/oozie/>, <http://www.cloudera.com/blog/2010/07/whats-new-in-cdh3-b2-oozie/>

Questions ?

