
# CumuloNimbo
# Final Review Meeting
# Brussels, Nov 27th, 2013

*FP7-257993*

## Filesystem Layer & DeltaOMID

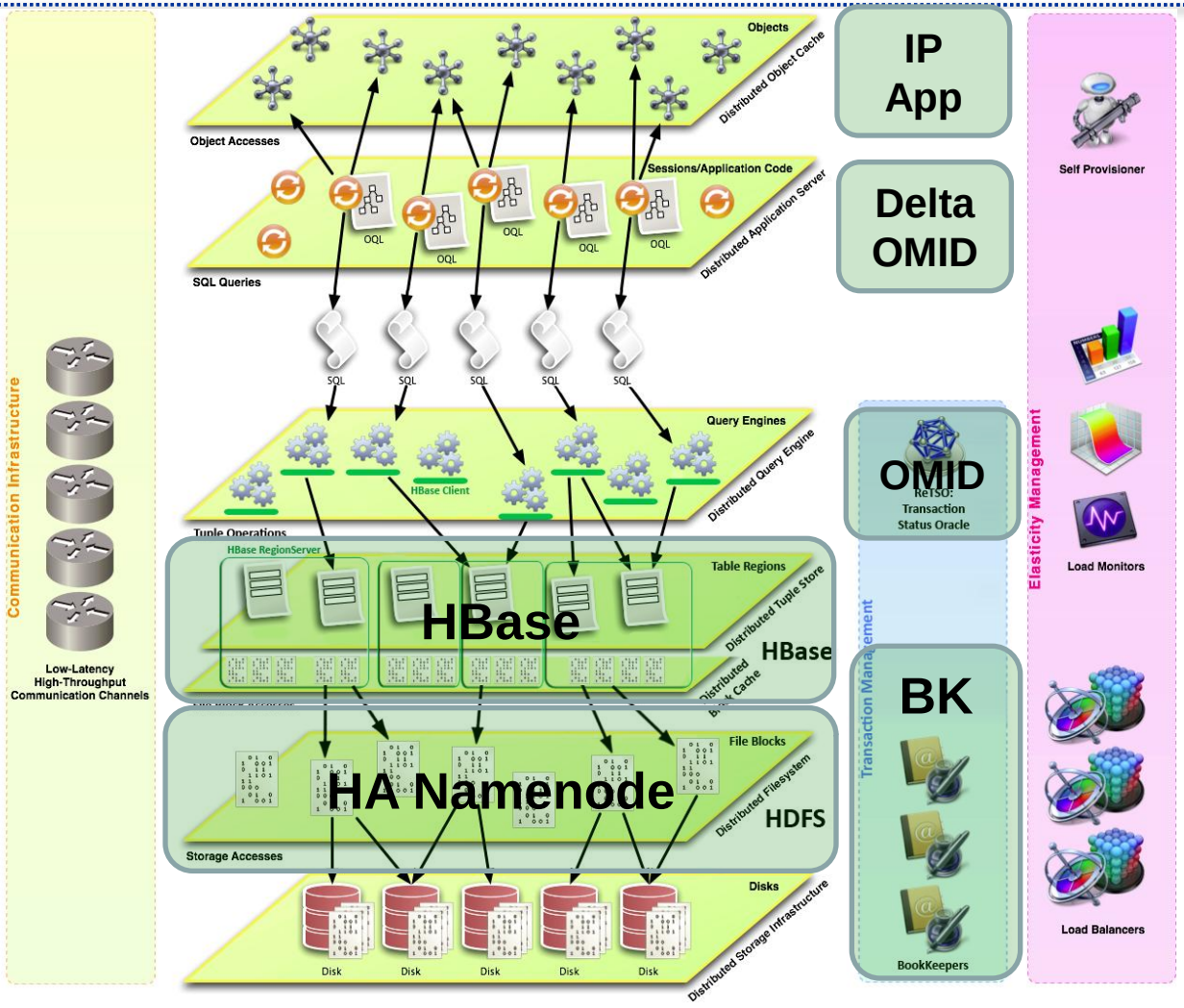
***Ivan Kelly**, Yahoo! Inc*
***Francisco Perez-Sorrosal**, Yahoo! Inc.*

SEVENTH FRAMEWORK
PROGRAMME

# Overview

- Primary contributions to WP3
  - Original goal was to build a filesystem, block cache
  - Focus shifted due to existence of HBase/HDFS
  - New focus:
    - Tackle the fault tolerance issues in HDFS & HBase
    - Provide direct Transactional access to HBase (OMID)
    - Incremental processing framework on HBase (DeltaOMID)
- Year 3:
  - HBase + BookKeeper
  - DeltaOMID

# Yahoo! Components in CN Stack



# HDFS (D3.3)

- HDFS namenode is SPOF for Hadoop
- Involved a large refactor of the namenode WAL, which was previously file targeted
  - JournalManager framework
  - Bookkeeper implementation of JournalManager
- Ships with Hadoop since 2.0.3-alpha
- In production in Huawei, under evaluation in Yahoo

# Bookkeeper (D3.1, D5.6)

- Scalable reliable distributed write ahead logging
- Started as subproject of Apache Zookeeper
- Many important features added during CumuloNimbo
  - Fencing
  - Speculative reads
  - Separate write and ack quorums
  - Auto rereplication
- Plan to graduate to Apache Top Level Project in Q1 2014
- In production in Yahoo, Twitter, Huawei, HubSpot

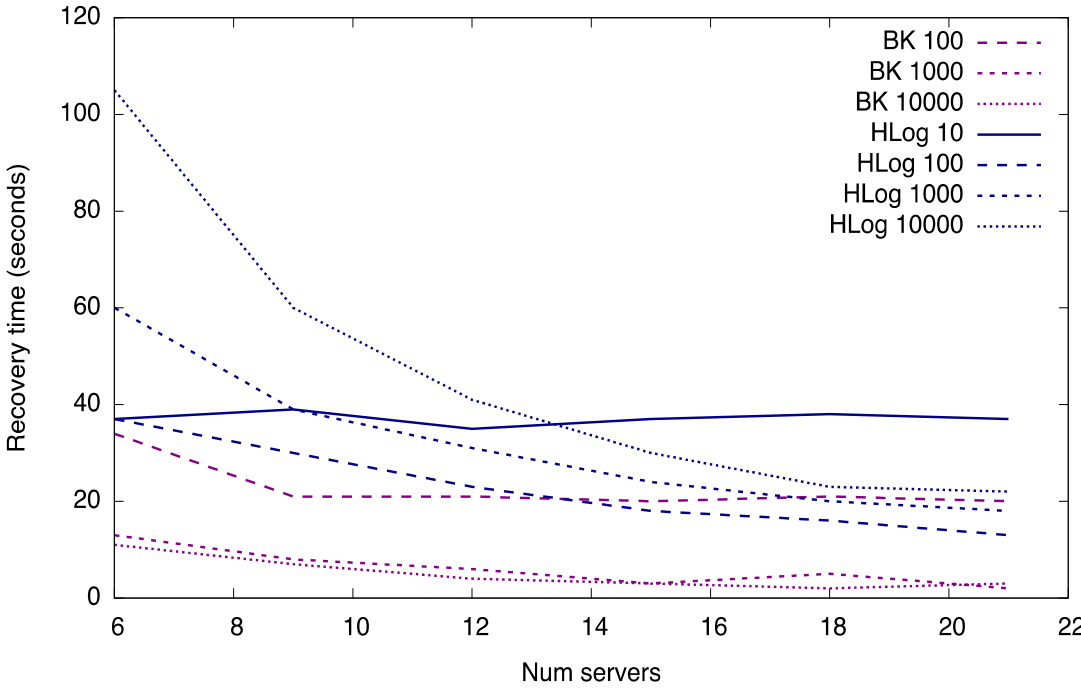
# HBase (D5.5)

- Through HDFS work, and speaking with HBase devs discovered a persistence issue with HBase:
  - WAL is only flushed out the sockets, no guarantee of hitting disk
  - In case of power loss → data loss
  - Possible Split-brain

- Our solution was to integrate BookKeeper with HBase
  - Strong persistence guarantees
  - Fencing → prevents split-brain problems

# HBase (Pre-evaluation)

- Evaluate whether Bookkeeper (BK) can give acceptable read performance
  - BK highly write optimized, reads a concern
  - BK already shown to match the write throughput required by HBase
- Benchmark simulated read pattern during node failure
- Result → Bookkeeper matched the read rate for > 100 regions per node (D5.7)
- Gave us confidence to continue implementation



# HBase (Implementation + Benchmark)

- Prototype implementation of HBase+BK
  - Functionally complete
  - Benchmarks with YCSB revealed a very low-write throughput (D5.7)

- Root cause: Synchronous architecture
  - HBase Regionserver has a number of IPC handler threads
  - IPC handlers block while handling a request
  - Higher latency due to hitting disk = less requests handled per second
  - Increasing IPC handler count had minimal effect

# HBase Evaluation + Recommendation

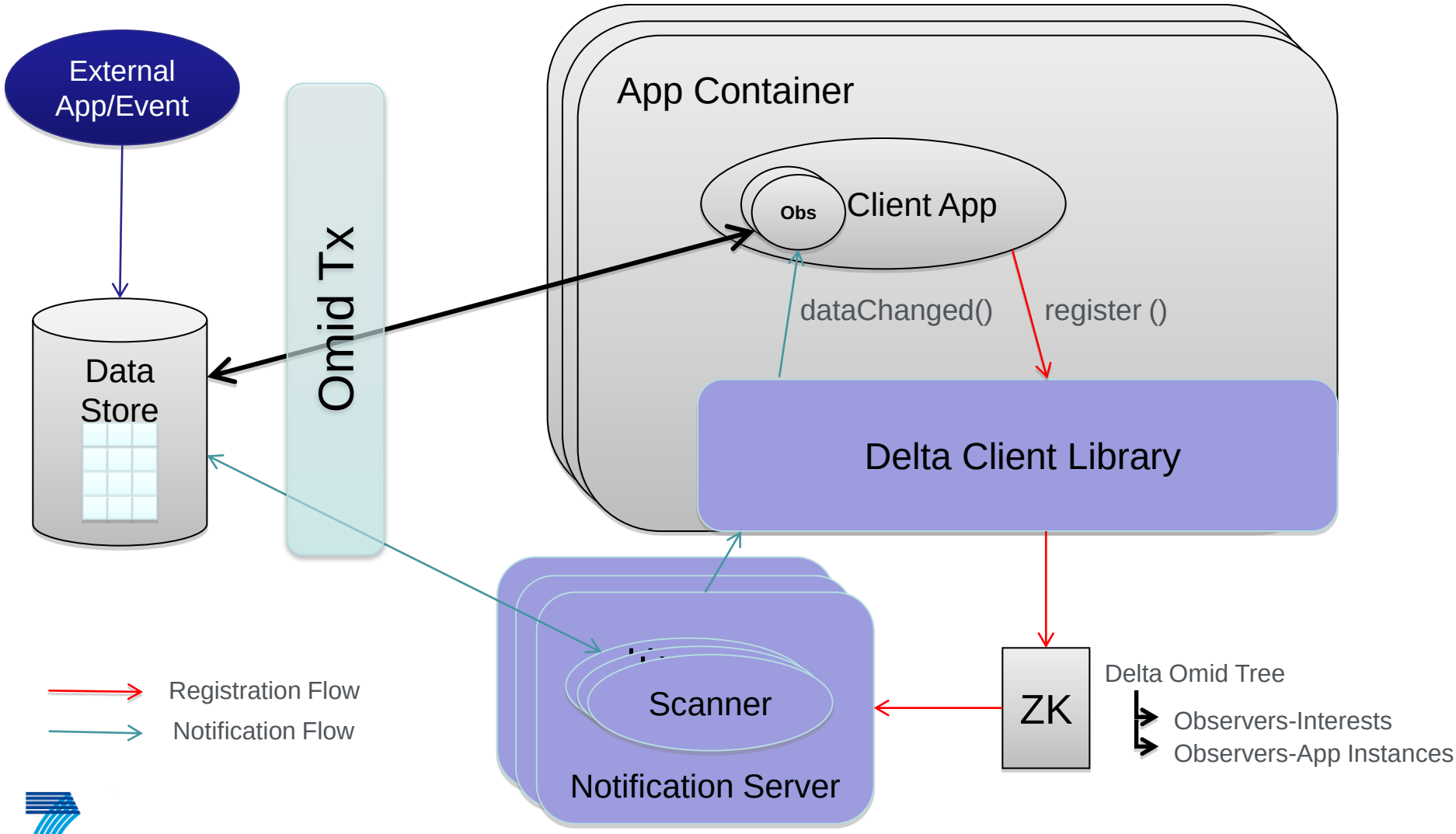
- Underlying **problem is HBase architecture**:
  - Not possible to add strong persistency guarantee w/o destroying write perf
  - Using HDFS's new fsync functionality will see same problem

- Therefore **we would recommend against using HBase on rotational storage**, as fsync latency will cripple throughput.

# DeltaOMID (D5.5)

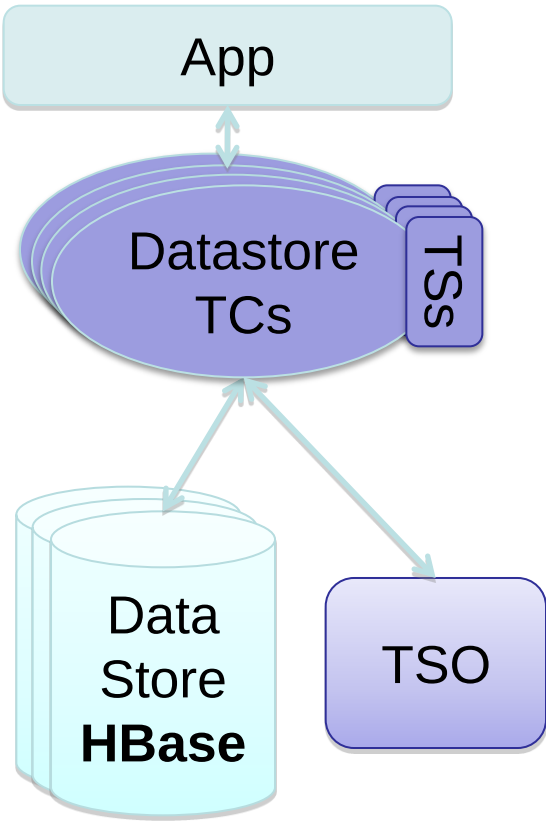
- Incremental Processing on top of HBase:
  - Big Data processing as small mutations to a large dataset triggered by events
- Benefits:
  - Low latency
  - Data consistency
- Requires:
  - **Notifications** → Allow register application interest in specific data and trigger specific actions at the application level upon updates → The Delta part
  - **Transaction management** → Provides Concurrency and Consistency of updates → The OMID part

# DeltaOMID


External App/Event
Omid Tx
App Container
Obs
Client App
dataChanged()
register ()
Data Store
Delta Client Library
Scanner
Notification Server
ZK
Delta Omid Tree
Observers-Interests
Observers-App Instances
Registration Flow
Notification Flow

# Transaction Management: OMID

- The Status Oracle (TSO)
  - A Tx Manager on top of HBase
  - Replicates Tx info (TSs) to Transactional Clients (Reduce comm overhead)
  - Uses a BookKeeper to recover from crashes
- Transactional Clients (TCs)
  - Connect to the TSO to:
    - Start Tx & Commit/Rollback Tx
  - Perform the required operations in HBase in the corresponding Tx Context

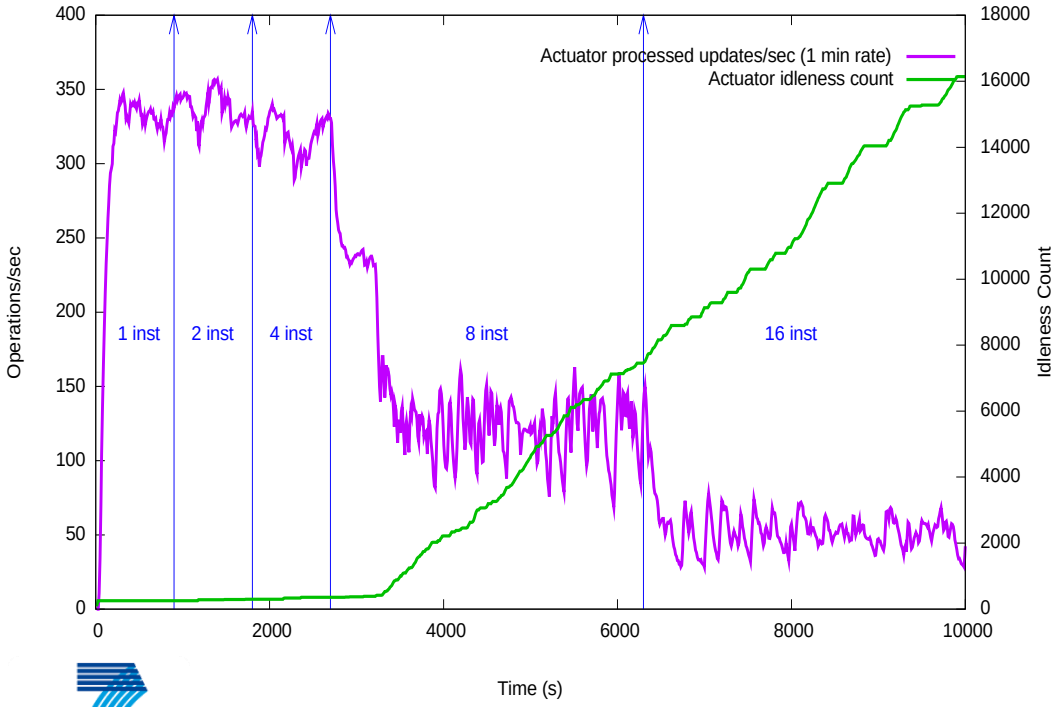


# DeltaOMID + OMID Results

# Benchmarks to test Consistent IP Fmwk (D5.10 & 11)

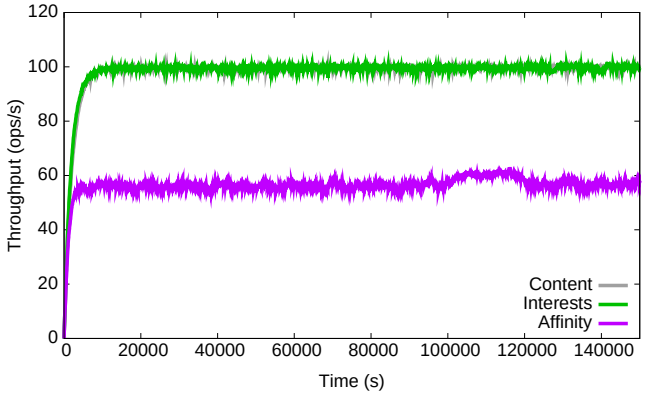
## Simple App: 1 Observer/Actuator

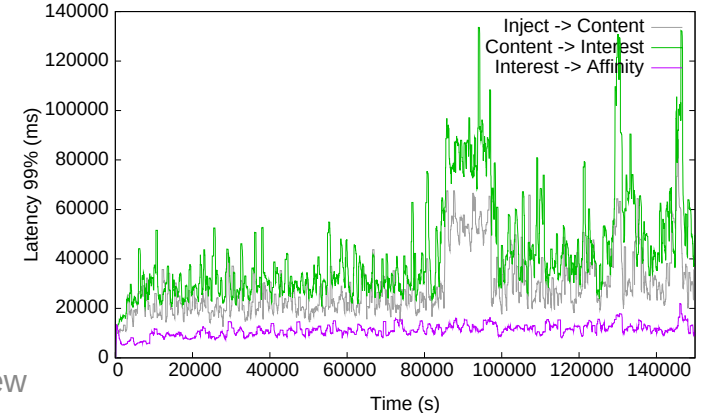
- Injection rate 1000 updates/s
- Throughput and Idleness




## Realistic App: 4 Observers/Actuators

- Injection rate 100 updates/s
- Throughput and Latency



# Y! Results

- Main Contributions
  - HDFS HA for CN storage layer (D3.3)
  - HBase unfeasible for CN platform (D5.5 & D.5.7)
  - Prove BK as a reliable and scalable WAL mechanism for CN platform
  - OMID - Transactions on key value store
  - Delta OMID Consistent Incremental Processing Platform for low-latency apps running in CN (D5.5 & D.5.7)
  - Deployed in Flexiant Cloud Solutions, Private & Public (D5.10-D5.11)
  - Use of SAP PMF Monitoring (Demo)

# Questions?


# Exploitation

*FP7-257993*

Yahoo Inc.

# Yahoo Exploitation

- Internal technology transfer

- Increase industry exposure through open source
  - Raises profile of engineering in Yahoo
  - Makes Yahoo attractive to developers

# BookKeeper

- Distributed write ahead logging
- In production during last 6 months
- Providing persistence layer in a major infrastructure service
- Apache Top Level Project in Q1 2014

# HDFS High Availability

- Currently being evaluated by Grid Services Team
- Concurrent effort to build next generation NameNode
  - Will reuse much of the work from CumuloNimbo
- Already shipping with Hadoop 2.0.3-alpha onwards

# DeltaOMID

- Omid open sourced on Github in 2011
- Pilot use-case discussed with product architects
- Will be presented at internal Yahoo conference in December 2013
- Plan to open-source in Q1 2014