Big Data @ Microsoft

A View from CISL

Raghu Ramakrishnan

Outline

- Big Data
  - The Digital Shoebox
- Selected topics:
  - Tiered Storage
  - Compute Fabric
    - REEF and YARN

Cloud Information Services Lab (CISL)

- Applied research for Cloud and Enterprise (CE)
- Focus areas:
  - Cloud data platforms and data-driven next-gen enterprise solutions
- Modus innovatii:
  - Embedded with the product team
  - Vehicle to engage deeply with MSR
  - Work with partners to apply ML/Cloud technology

Big Data

What's the big deal?

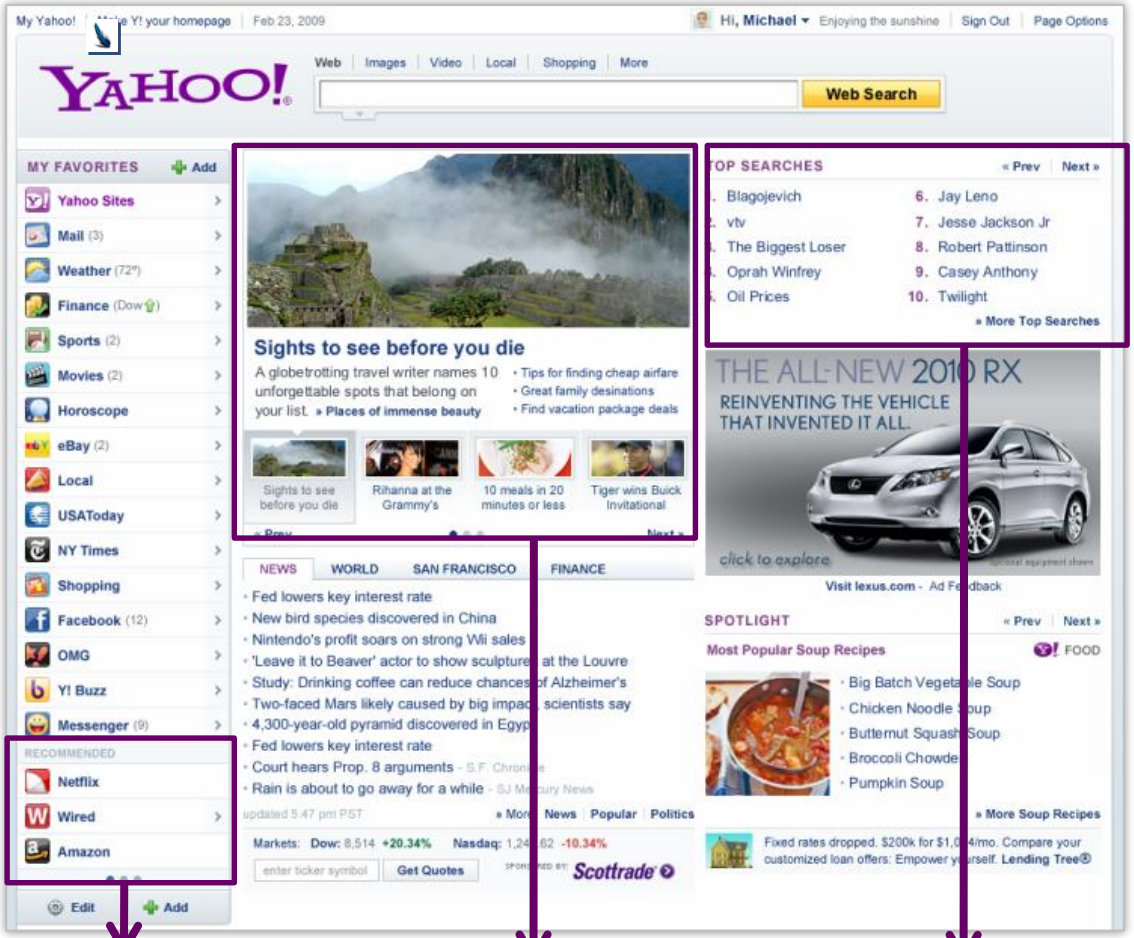
What's New?

- **What we're doing with it!**
  - The tech is best thought of in terms of what it enables
- Why is this more than tech evolution?
  - Cloud services + advances in analytics + HW trends = Ability to cost-effectively do things we couldn't dream of before
  - Uncomfortably fast evolution = revolution

Content Optimization

Agrawal et al., CACM 56(6):92-101 (2013)

Content Recommendation on Web Portals



Recommended links **News Interests** **Top Searches**

Key Features

Package Ranker (CORE)

Ranks packages by expected CTR based on data collected every 5 minutes

Dashboard (CORE)

Provides real-time insights into performance by package, segment, and property

Mix Management (Property)

Ensures editorial voice is maintained and user gets a variety of content

Package rotation (Property)

Tracks which stories a user has seen and rotates them after user has seen them for a certain period of time

Key Performance Indicators

Lifts in quantitative metrics

Editorial Voice Preserved

CORE Dashboard:
Segment Heat Map

Package	male	female	OMG	BUAuto	BUEnt	BU Fin	Health	BUSport	NBA	BUTrav	ALL
	408,260 18,440 0.0452 8.477	390,404 14,449 0.037 -11.113	270,039 16,940 0.0627 50.661	121,080 7,389 0.061 46.564	270,038 16,940 0.0627 50.661	325,873 20,012 0.0614 47.488	195,796 12,763 0.0652 56.553	350,152 21,454 0.0613 47.152	132,916 9,457 0.0712 70.879	123,388 7,896 0.064 53.691	923,611 38,457 0.0416 0
	1 8,067 852 0.1056 153.654	1 7,657 674 0.088 111.405	1 5,125 720 0.1405 237.406	1 2,382 286 0.1201 188.362	1 5,125 720 0.1405 237.406	1 6,415 858 0.1337 221.221	1 3,769 532 0.1412 239	1 6,750 917 0.1359 226.272	1 2,585 385 0.1489 257.696	1 2,490 330 0.1325 218.294	1 18,131 1,738 0.0958 130.143
	5 9,968 644 0.0646 55.164	3 12,847 777 0.0605 45.256	2 8,569 885 0.1033 148.043	4 3,529 326 0.0924 121.86	2 8,569 885 0.1033 148.043	3 9,744 922 0.0946 127.252	3 6,067 643 0.106 154.537	2 10,187 1,004 0.0986 136.702	5 3,820 420 0.1099 164.058	2 4,037 433 0.1073 157.598	4 25,744 1,595 0.062 48.798
	2 3,326 249 0.0749 79.8	5 3,954 212 0.0536 28.769	5 2,521 231 0.0916 120.066	2 1,004 102 0.1016 143.995	5 2,521 231 0.0916 120.066	5 3,016 276 0.0915 119.782	5 1,860 186 0.1 140.167	3 3,291 310 0.0942 126.229	3 1,141 136 0.1192 186.264	3 1,039 100 0.0962 131.152	3 8,500 541 0.0636 52.859
	11 2,562 133 0.0519 24.677	13 2,004 81 0.0404 -2.926	3 1,250 122 0.0976 134.403	6 629 51 0.0811 94.73	3 1,250 122 0.0976 134.403	4 1,608 151 0.0939 125.53	2 919 103 0.1121 169.175	4 1,669 154 0.0923 121.604	4 655 74 0.113 171.334	4 591 55 0.0931 123.506	10 5,342 252 0.0472 13.295
	3 2,881 206 0.0715 71.727	2 3,242 230 0.0709 70.384	4 2,071 196 0.0946 127.295	3 949 95 0.1001 140.42	4 2,071 196 0.0946 127.295	2 2,614 254 0.0972 133.368	4 1,605 165 0.1028 146.901	5 2,740 239 0.0872 109.489	10 1,036 94 0.0907 117.912	9 958 78 0.0814 95.543	2 7,043 493 0.07 68.114
	6 10,785 649 0.0602 44.523	4 12,768 742 0.0581 39.571	7 8,580 694 0.0809 94.261	7 3,511 283 0.0806 93.584	7 8,580 694 0.0809 94.261	6 9,725 795 0.0817 96.332	6 6,138 550 0.0896 115.204	6 10,670 866 0.0812 94.525	11 3,669 321 0.0875 110.122	5 3,785 339 0.0896 115.104	5 27,331 1,641 0.06 44.2
	10 22,202 1,212 0.0546 31.106	7 23,328 1,200 0.0514 23.543	6 15,593 1,289 0.0827 98.535	5 6,552 533 0.0813 95.374	6 15,593 1,289 0.0827 98.535	7 17,652 1,376 0.078 87.214	8 10,797 915 0.0847 103.532	7 19,050 1,522 0.0799 91.882	9 6,639 604 0.091 118.498	7 6,435 552 0.0858 106.018	6 52,978 2,786 0.0526 26.299
	22 26,685 1,160 0.0435 4.401	10 35,405 1,530 0.0432 3.786	8 19,832 1,572 0.0793 90.371	9 7,844 552 0.0704 69.011	8 19,832 1,572 0.0793 90.371	8 21,743 1,641 0.0755 81.26	7 13,721 1,167 0.0851 104.267	8 22,168 1,743 0.0786 88.836	8 8,249 788 0.0955 129.424	8 8,327 689 0.0827 98.721	18 74,559 3,167 0.0425 2.014
	4 7,745 518 0.0669 60.628	26 7,202 185 0.0257 -38.306	13 4,898 322 0.0657 57.889	15 2,308 148 0.0641 54.007	13 4,898 322 0.0657 57.889	11 6,051 423 0.0699 67.891	19 3,652 235 0.0643 54.544	9 6,436 506 0.0786 88.82	2 2,562 308 0.1202 188.726	12 2,359 169 0.0716 72.057	7 17,235 834 0.0484 16.217
	7 7,699 460 0.0597 43.495	29 7,201 169 0.0235 -43.635	11 4,809 340 0.0707 69.8	10 2,269 158 0.0696 67.239	11 4,809 340 0.0707 69.8	9 6,004 433 0.0721 73.205	14 3,544 243 0.0686 64.674	10 6,247 475 0.076 82.615	6 2,482 257 0.1035 148.682	11 2,329 167 0.0717 72.211	12 17,169 783 0.0456 9.525
	7,688 393	7,229 336	4,785 363	2,280 139	4,785 363	6,037 403	3,501 245	6,319 430	2,397 182	2,312 152	17,275 833

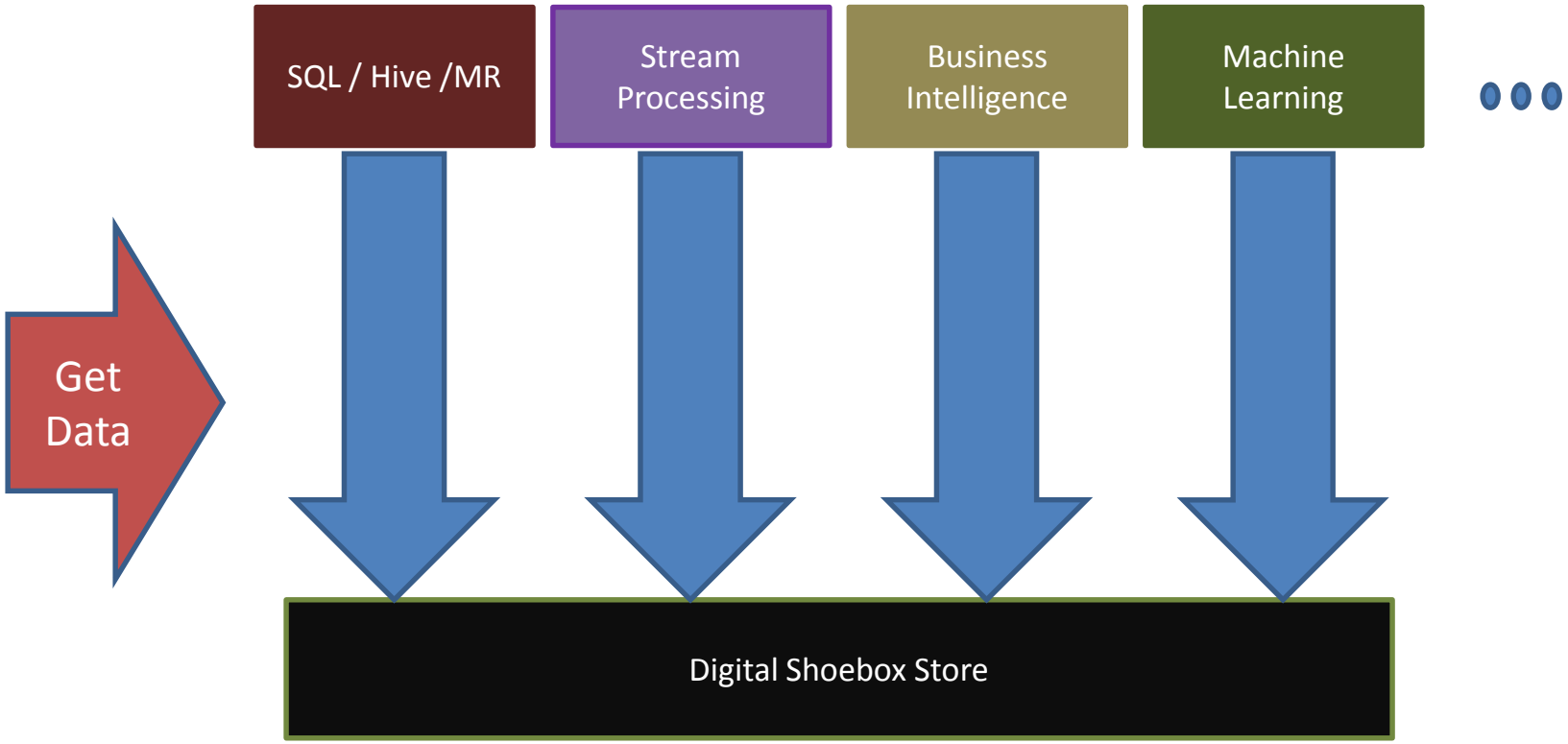
Telemetry

- Data: Time series of logs from user activity and system probes (live and historical archives)
  - CTP (Customer Touch Points) data
  - STP (System Touch Points) data
- Goal:
  - Determine possible causes of outages, particularly the long ones
  - Predictive and forensic
- Planned steps:
  - Identification of the team that can resolve an outage
  - Visualization of time series to understand long outages that are difficult to resolve
  - Discover and learn patterns associated with outage trends and use them to predict outages

Big Data

So, what should we build?

The Digital Shoebox

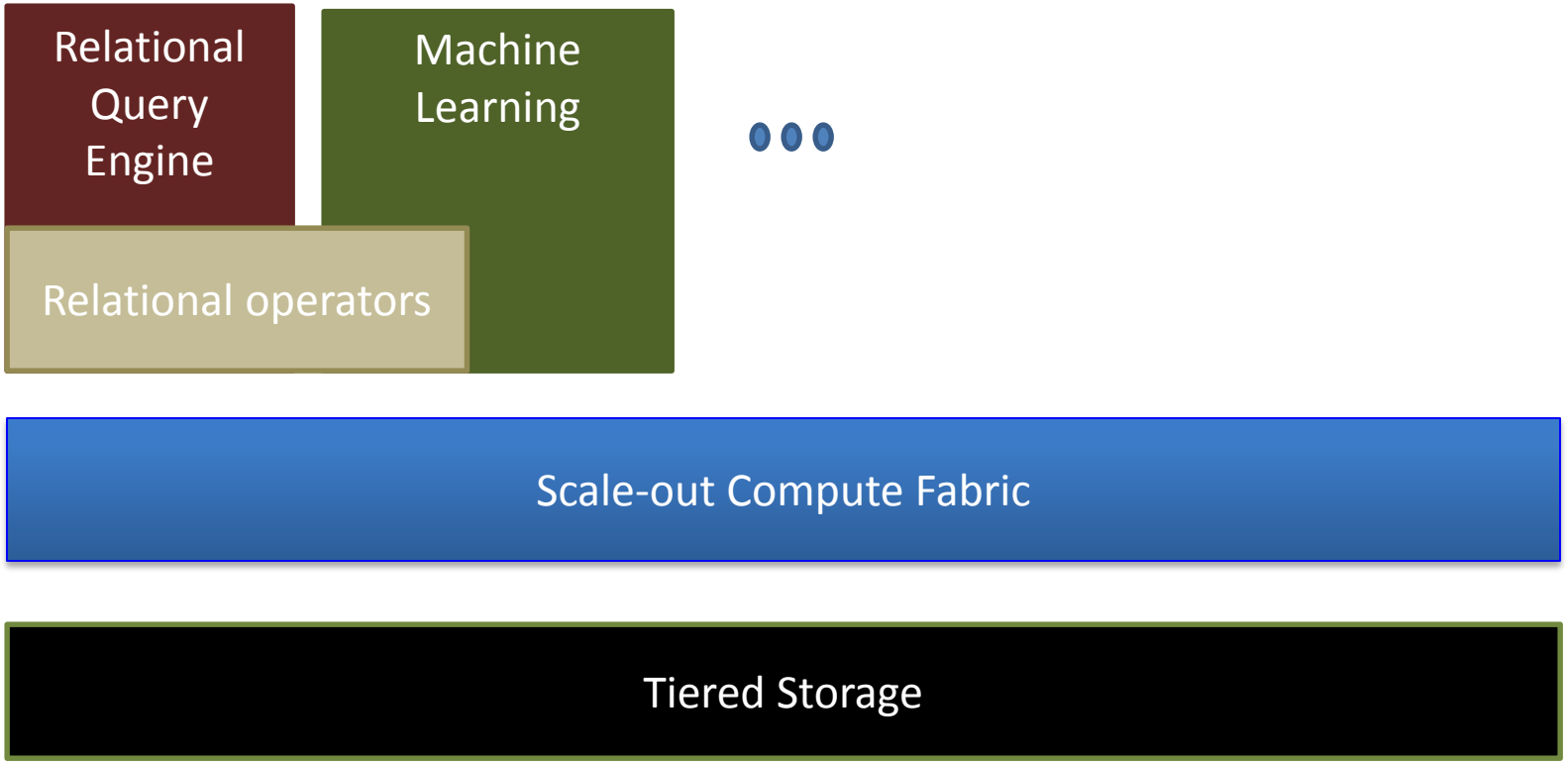


Capture any data, react instantaneously, store for later

Use any analysis tool (anywhere, in any combination, interactively)

Collaborate/Share selectively

Building a Digital Shoebox



Questions

- What is the right balance between common building blocks and custom analytic engines?
- What is the right layering to achieve this?
  - Storage vs. compute tiers; networking trends
  - Scheduling of shared resources
  - Multi-tenanted services
  - Security
  - Evaluation: What is good enough?

Challenges

- Volume
  - Elastic scale-out
- Variety
  - Trade-off: Shared building blocks vs. custom engines
  - Metadata management
    - Many catalogs at many layers (files, tables, docs)
    - Many owners (federation, integration, access control)
- Velocity
  - Real-time and OLTP, interactive, batch

Challenges

- Multi-tenanted services
  - HA, rolling upgrades
  - Security (Authentication, isolation, intrusion, DOS)
- CRUD workloads
  - How closely can we couple these with analytics?
- Federated access
  - Bring external data in for analysis
  - Apply analysis in-situ to data elsewhere

Tiered Storage

Dave Campbell, Sriram Rao, XCG

How Far Away is Data?

- GFS and Map-Reduce:
  - Schedule computation “near” data
  - i.e., on machines that have data on their disks
- But
  - Windows Azure Storage
    - And slower tiers such as tape storage …
  - Main memory growth
    - And flash, SSDs, NVRAM etc. …
- Must play two games simultaneously:
  - Cache data across tiers, anticipating workloads
  - Schedule compute near cached data

Scale-Out Compute Fabric
YARN and REEF

YARN

- Resource manager for Hadoop2.x
- Allocates compute containers to competing jobs
  - Not necessarily MR jobs!
- Other RMs include Corona, Mesos, Omega

REEF

- Relies on YARN resource manager
  - Can re-target to other RMs
- Evaluator: YARN container with REEF services
  - Capability-awareness, Storage support, Fault-handling support, Communications, Job/task tracking, scheduling hooks
- Activity: User Code to be executed in an Evaluator
  - Monitored, preemptable, re-started as needed
  - Unique id over lifetime of job
  - Executes in an Evaluator, which can be re-used

The Team










What have we built on top of REEF?

(so far)

MapReduce library

- Runs Hive and Pig
- Excellent starting point for M/R optimizations: Caching, Shuffle, Map-Reduce-Reduce, Sessions, …

Machine Learning algorithms

- Scalable implementations: Decision Trees, Linear Models, Soon: SVD
- Excellent starting point for: Fault awareness in ML

Scheduling in Hadoop

(Curino, Douglas, Rao)

Popular schedulers

CapacityScheduler

FairScheduler

Deadline-oriented scheduling

New idea:

Support work-preserving preemption

(via) checkpointing → more than preemption

Previous Work

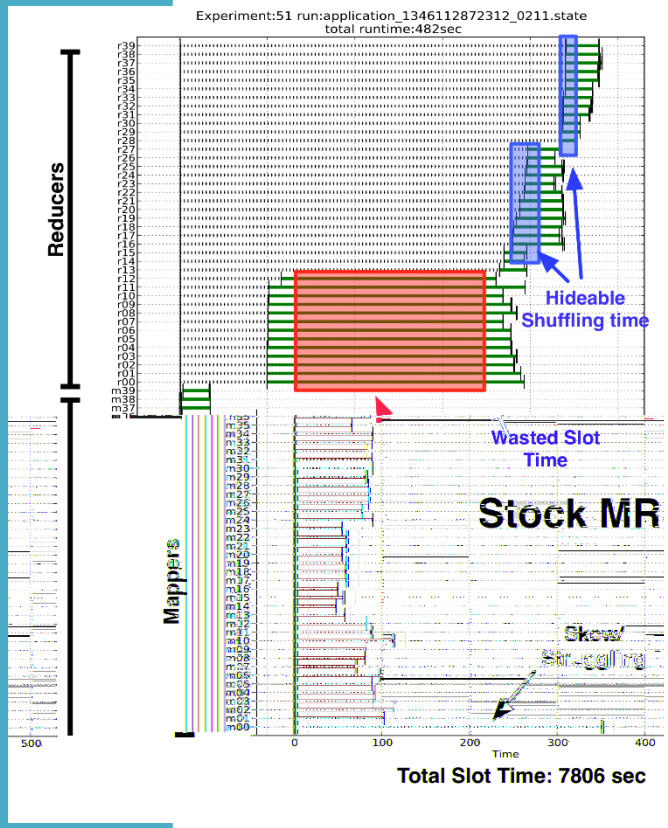
(*Amoeba*: SoCC'12)

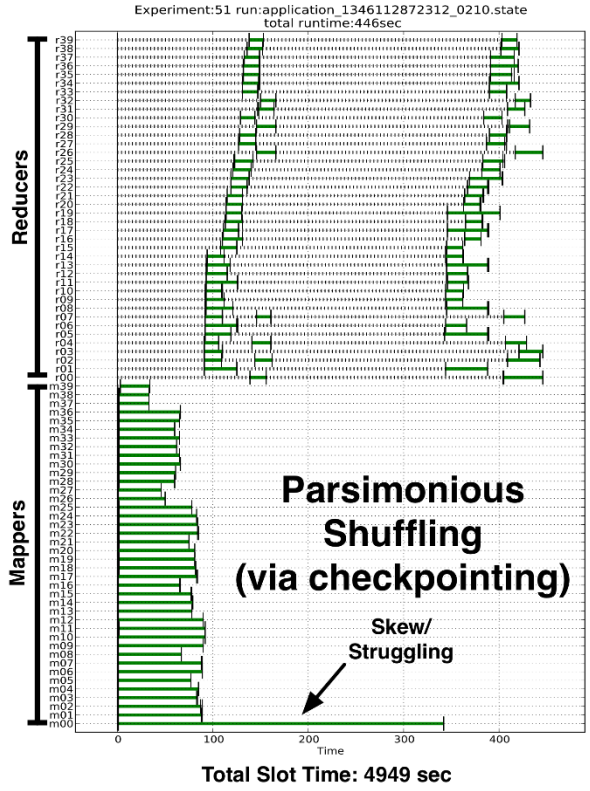
- *Amoeba*, a lightweight mechanism for enabling elasticity in data-intensive compute frameworks
  - Add work-conserving preemption via a "checkpoint/restart" mechanism that saves task output
  - Our observation:
    - A reduce task processes a group of keys, one at a time
    - A "key boundary" is a split point—where a reduce task execution can be safely terminated, and a new task can be spawned for the remaining work
- Resource consumption of jobs is elastic
  - Scale up/down usage based on cluster resource availability
- Preliminary results show that Amoeba can speed up jobs by 33%

- *Build on previous work to add preemption to YARN and focus on scheduling*

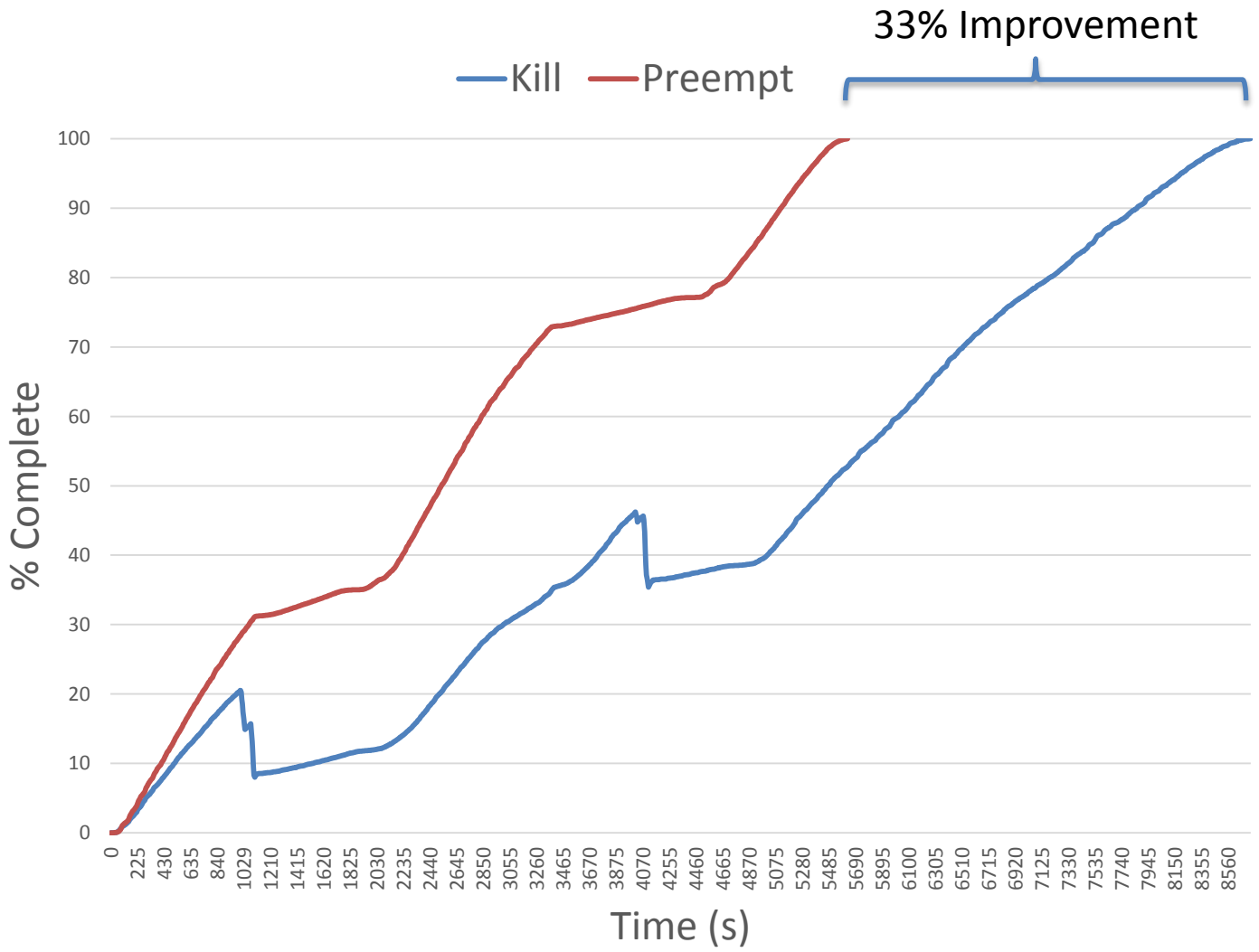
Dynamic Optimization

Leveraging checkpointing for parsimonious scheduling in MR

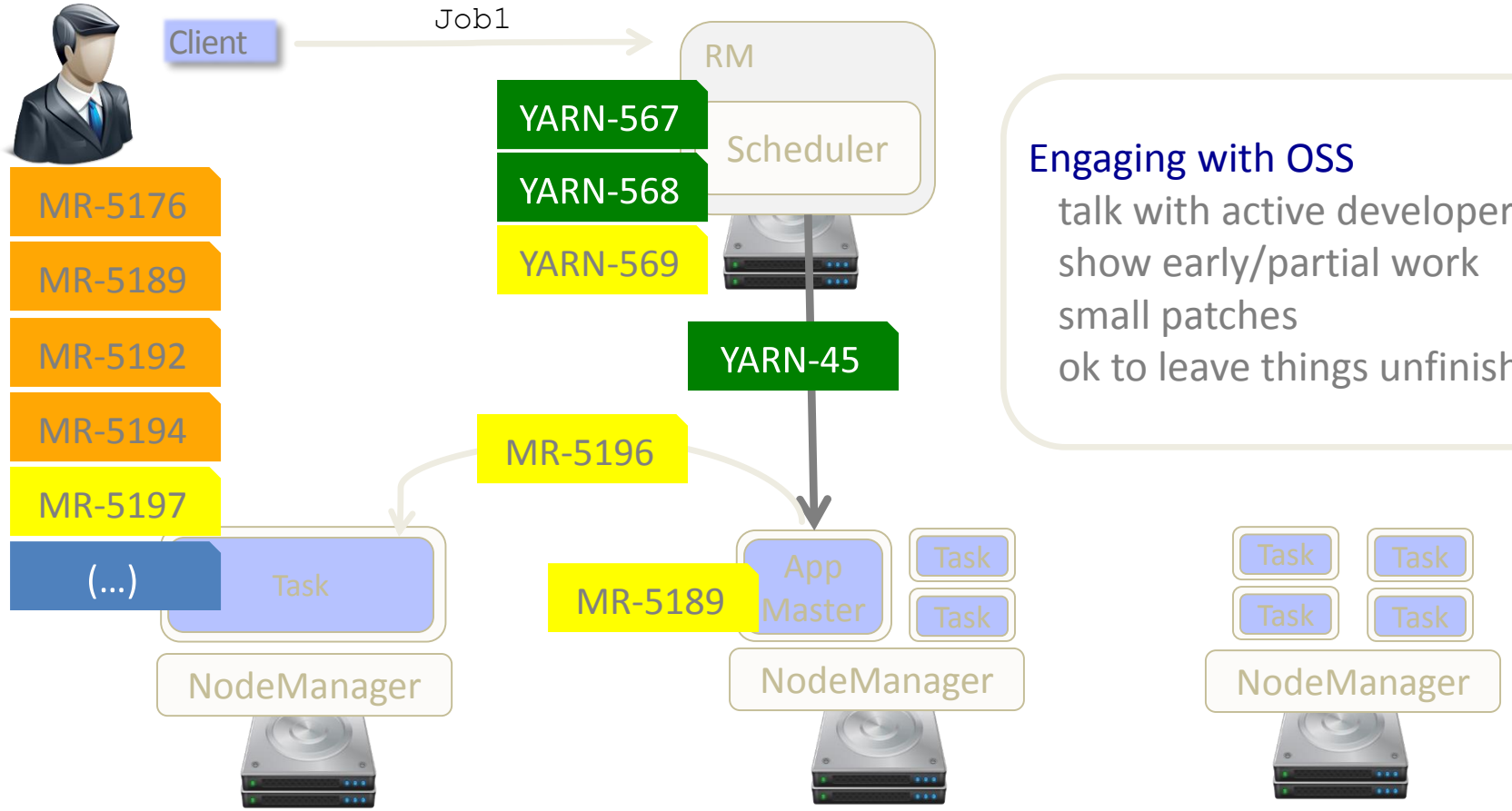




Killing Tasks vs. Preemption



Contributing to Apache



Engaging with OSS

- talk with active developers
- show early/partial work
- small patches
- ok to leave things unfinished

Collaborations

- AIP
- GSL
- Isotope team
- Galen Hunt's team (Drawbridge)
- MSR, XCG