What do real life Hadoop workloads look like?

Interactive Analytical Processing in Big Data Systems: A Cross-Industry Study of MapReduce Workloads

cloudera

Yanpei Chen, Sara Alspaugh, Randy Katz

Background

- Hadoop use cases spread beyond tech industry

"The Alot is Better Than You at Everything", http://hyperboleandahalf.blogspot.com/2010/04/alot-is-better-than-you-at-everything.html.

- How do we design Hadoop to target real life use cases?
- This talk – seven workloads across several industry sectors

MapReduce Examples Established and New

- Established, automated, e.g. reverse web link graph
  - map: input web page content, output ⟨target, source⟩
  - reduce: input [⟨ target, *⟩], output ⟨target, list(source)⟩

- New, human assisted, e.g. media outlet track audience behavior
  - compute most watched media items (automated)
  - demographic analysis on audiences (ad hoc exploration)
  - talk with producers, reflect on nature of content (human)
  - feed back into planning for new content

Workload Characterization Important

- Design better, more efficient systems
- Workload-specific info creates new design opportunities
  - auto tune/config. Hadoop ecosystem for performance
  - per-job or per-workflow dynamic optimization
  - per-dataset auto format and layout over time
  - build hybrid systems of Hadoop + RDMBS + others
- Understand how systems are used in real life

This Talk: Seven Hadoop Deployments

	Trace	Date	Length	# machines	# jobs
ecommerce, telecomm, media, retail	Cloudera customer A	2011	1 month	<100	5759
	Cloudera customer B	2011	9 days	300	22974
	Cloudera customer C	2011	1 month	700	21030
	Cloudera customer D	2011	2 months	450	13283
	Cloudera customer E	2011	9 days	100	10790
growth over a year	Facebook	2009	6 months	600	1129193
	Facebook	2010	1.5 months	3000	1169184
non-trivial	Total		~ 1 year	>5000	2372213

Key Questions to Ask

data patterns
- typical data set size?
- uniform or skewed access?
- temporal locality?

arrival patterns
- regular cycles over time?
- bursts in workload?

compute patterns
- common job types?
- query-like frameworks (Hive/Pig)?

cross-workload comparison
- variations between workloads?
- representative characteristics?

Covered in This Talk

data patterns
- typical data set size?
- uniform or skewed access?
- temporal locality?

arrival patterns
- regular cycles over time?
- bursts in workload?

compute patterns
- common job types?
- query-like frameworks (Hive/Pig)?

cross-workload comparison
- variations between workloads?
- representative characteristics?

Covered in This Talk

data patterns
- **typical data set size?**
- uniform or skewed access?
- temporal locality?

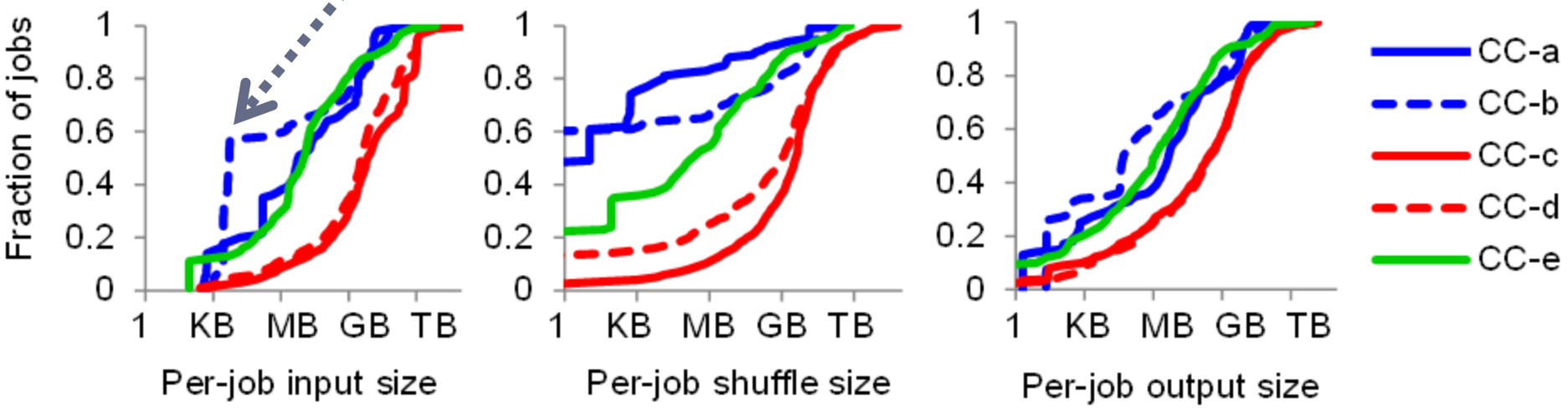
arrival patterns
- regular cycles over time?
- bursts in workload?

compute patterns
- common job types?
- query-like frameworks (Hive/Pig)?

cross-workload comparison
- variations between workloads?
- representative characteristics?

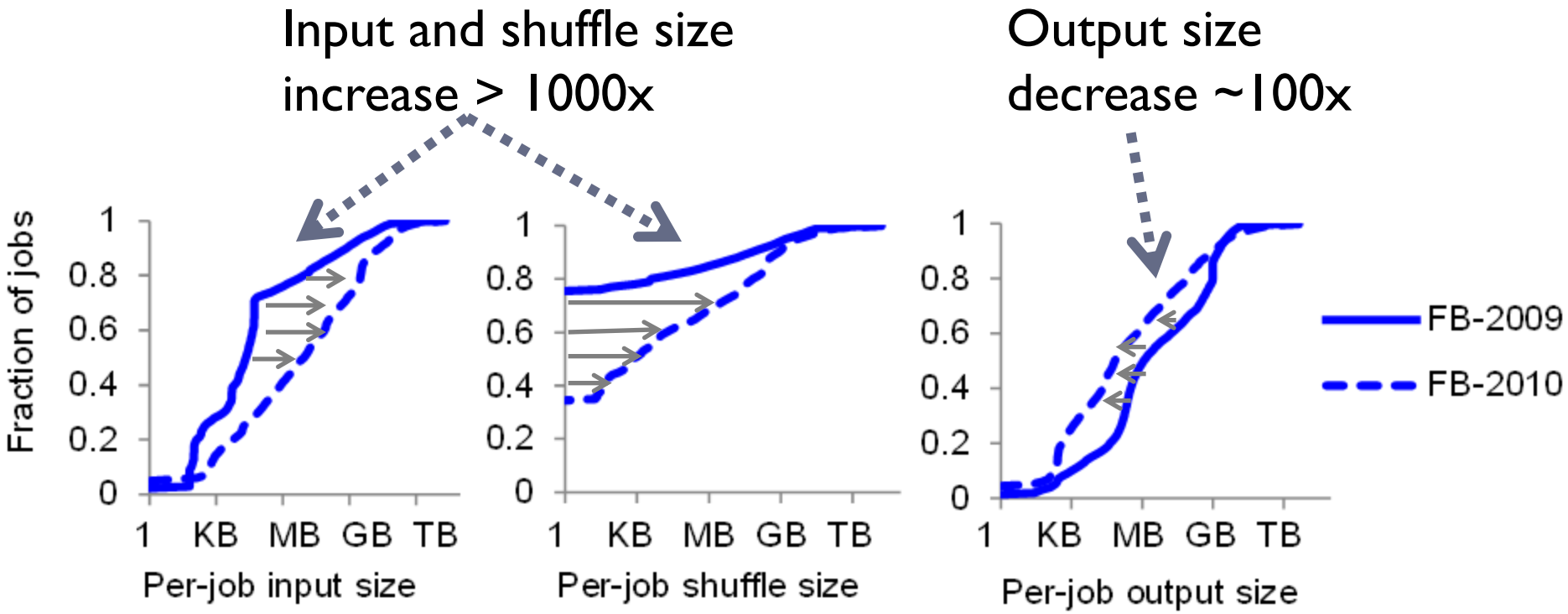
Typical Data Set Size

Read as: ~60% of jobs have an input size of several KBs.



Most jobs have KB to TB data size

Typical Data Set Size, Year-to-Year

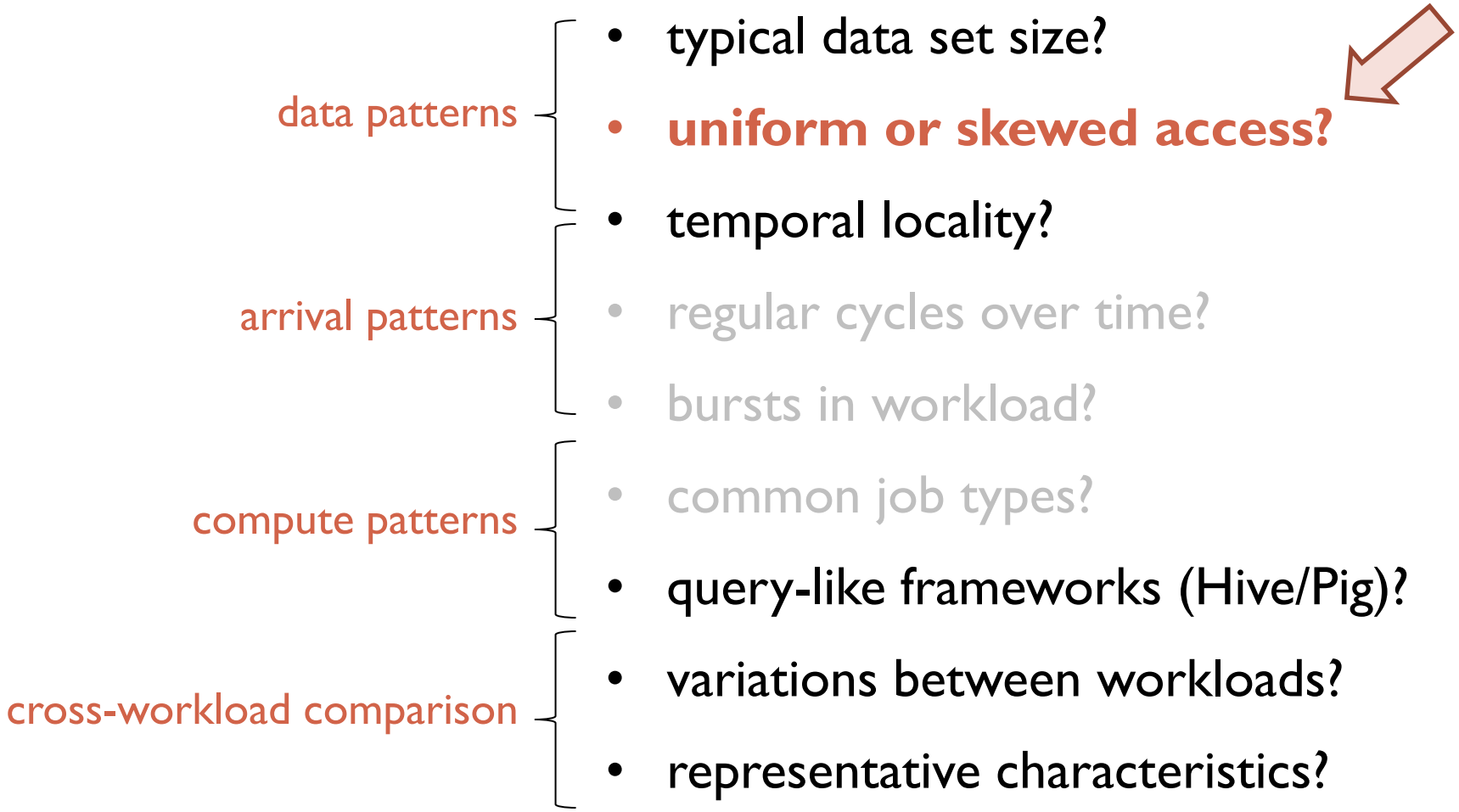


Output selectivity increase 100,000x

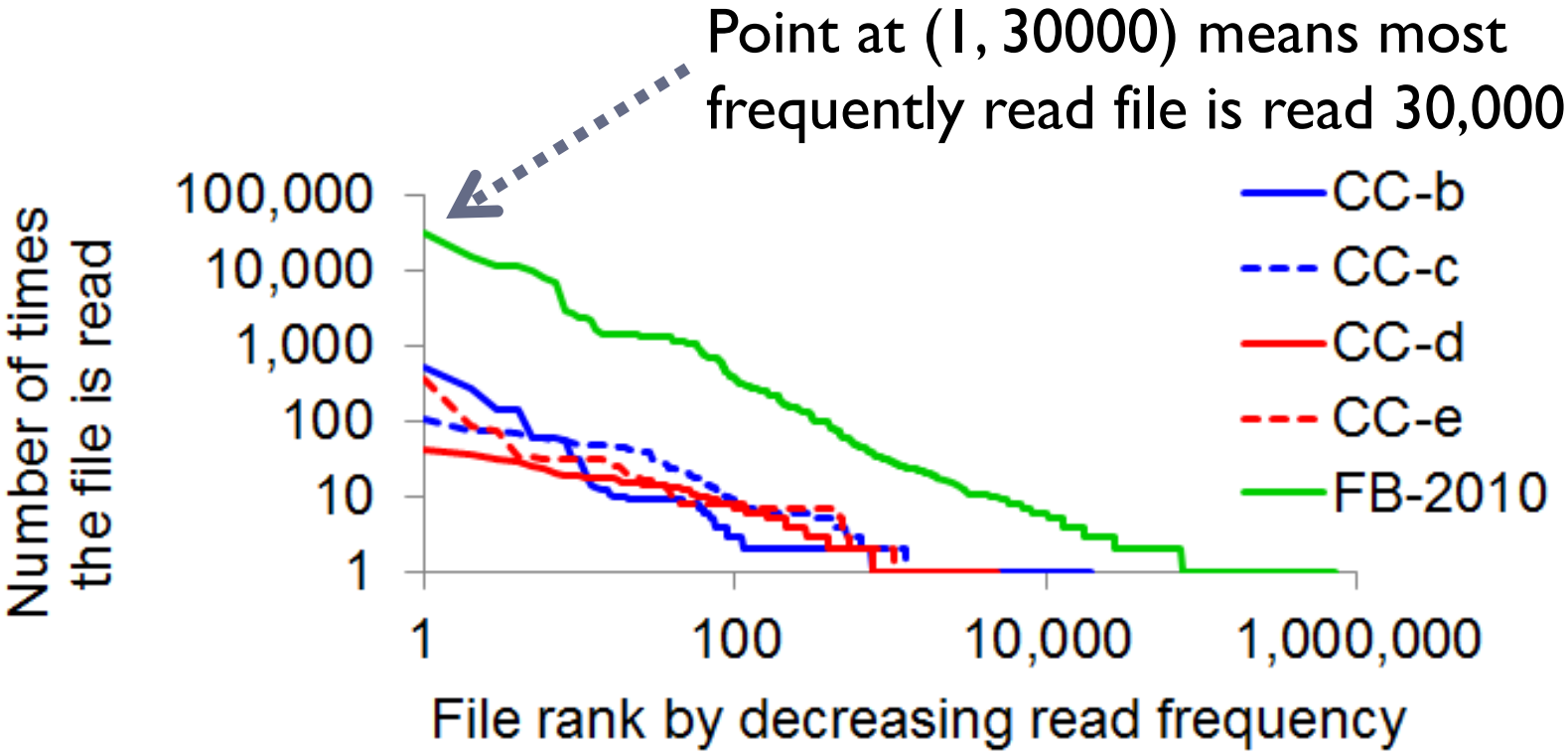
Typical Data Set Size

- Most jobs KB to TB
- Output decrease 100x, input increase 1000x over a year
  - people ask better questions over time?

Covered in This Talk



Access Skew

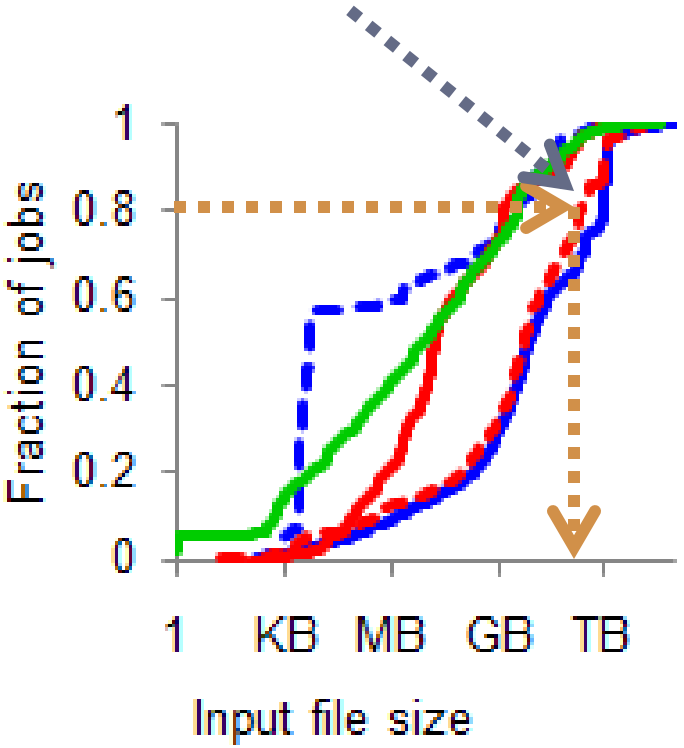


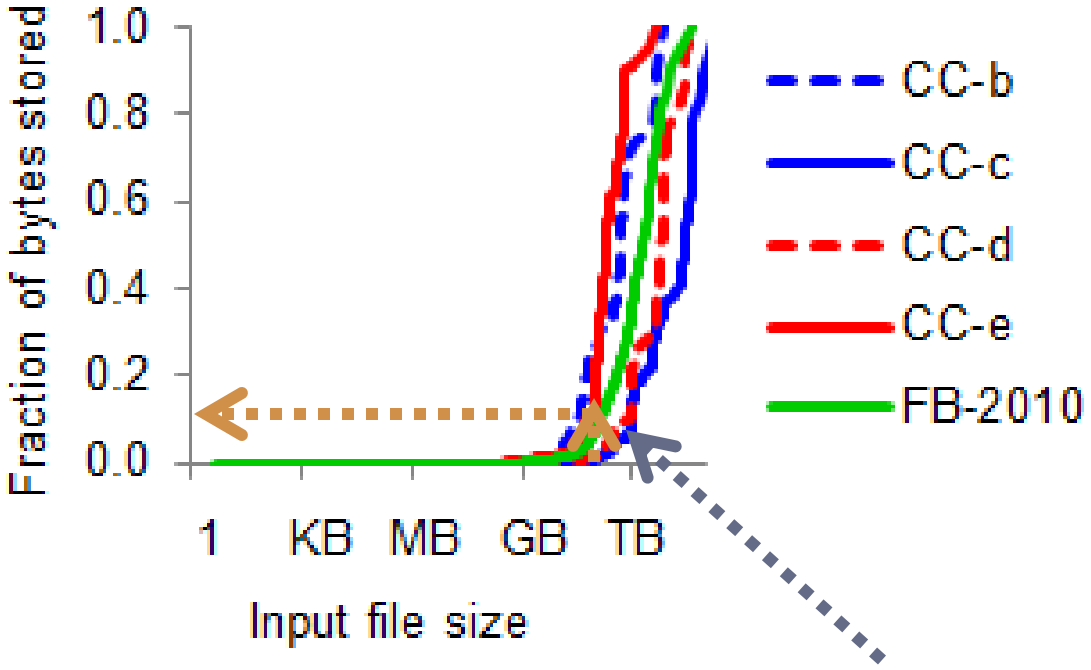
Zipf distribution → cache!!

Same slope → Zipf distributions with the same skew!!

80-10-or-less Rule in Bytes

80% of jobs read files of less than 100s GB





Files less than 100s GB form <10% of bytes stored

Access Skew

- Same Zipf distributed skew across industry sectors
  - time based analysis?
  - human analyst mental capacity?
  - consumer dictate pace of business evolution?
- 80% of jobs access 1-8% of bytes
  - skew increasing due to ever more raw data?

Covered in This Talk

data patterns
- typical data set size?
- uniform or skewed access?
- **temporal locality?**

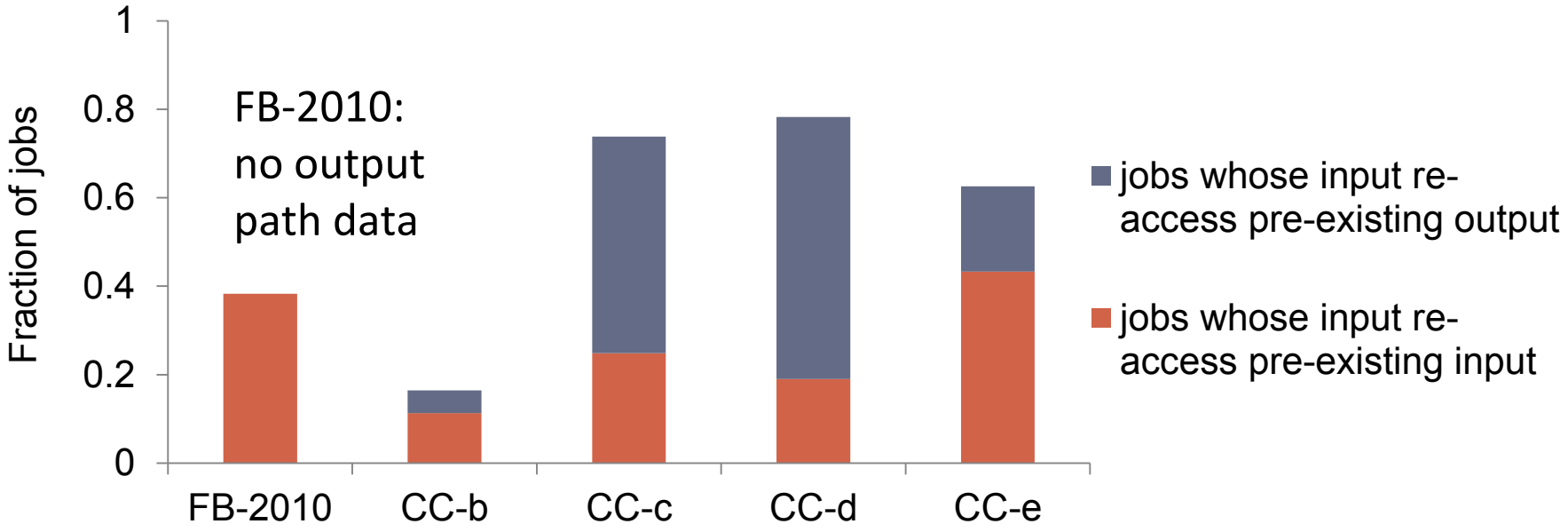
arrival patterns
- regular cycles over time?
- bursts in workload?

compute patterns
- common job types?
- query-like frameworks (Hive/Pig)?

cross-workload comparison
- variations between workloads?
- representative characteristics?

Temporal Locality

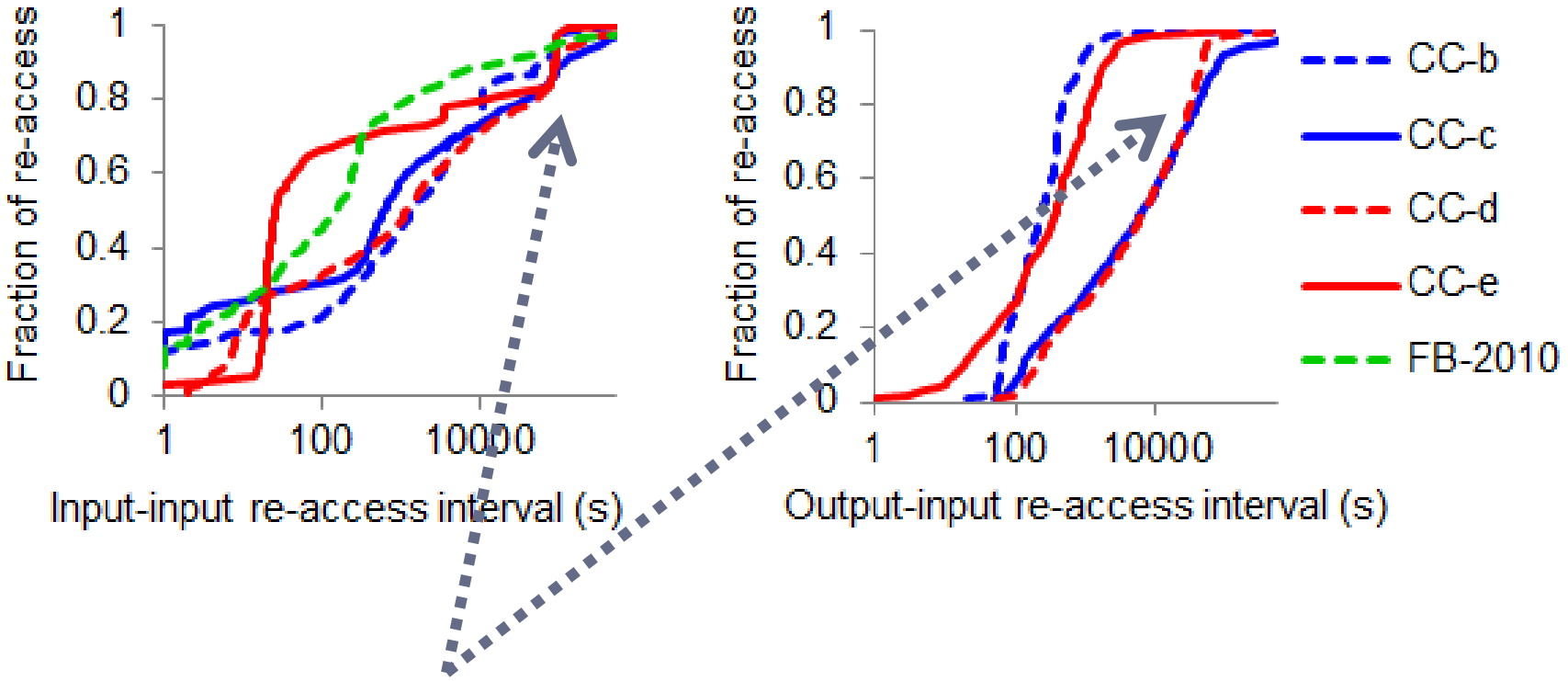
How often is data read or written re-read by later jobs?



20-80%, depends on workload

Temporal Locality

How much time before subsequent jobs re-read existing data?



80% of re-accesses are within 3 hours

Temporal Locality

- 20-80% of jobs re-access pre-existing input
- 80% of re-accesses are within ~3hrs
  - attention span of professionals?
  - different reaction time for different industries?

Covered in This Talk

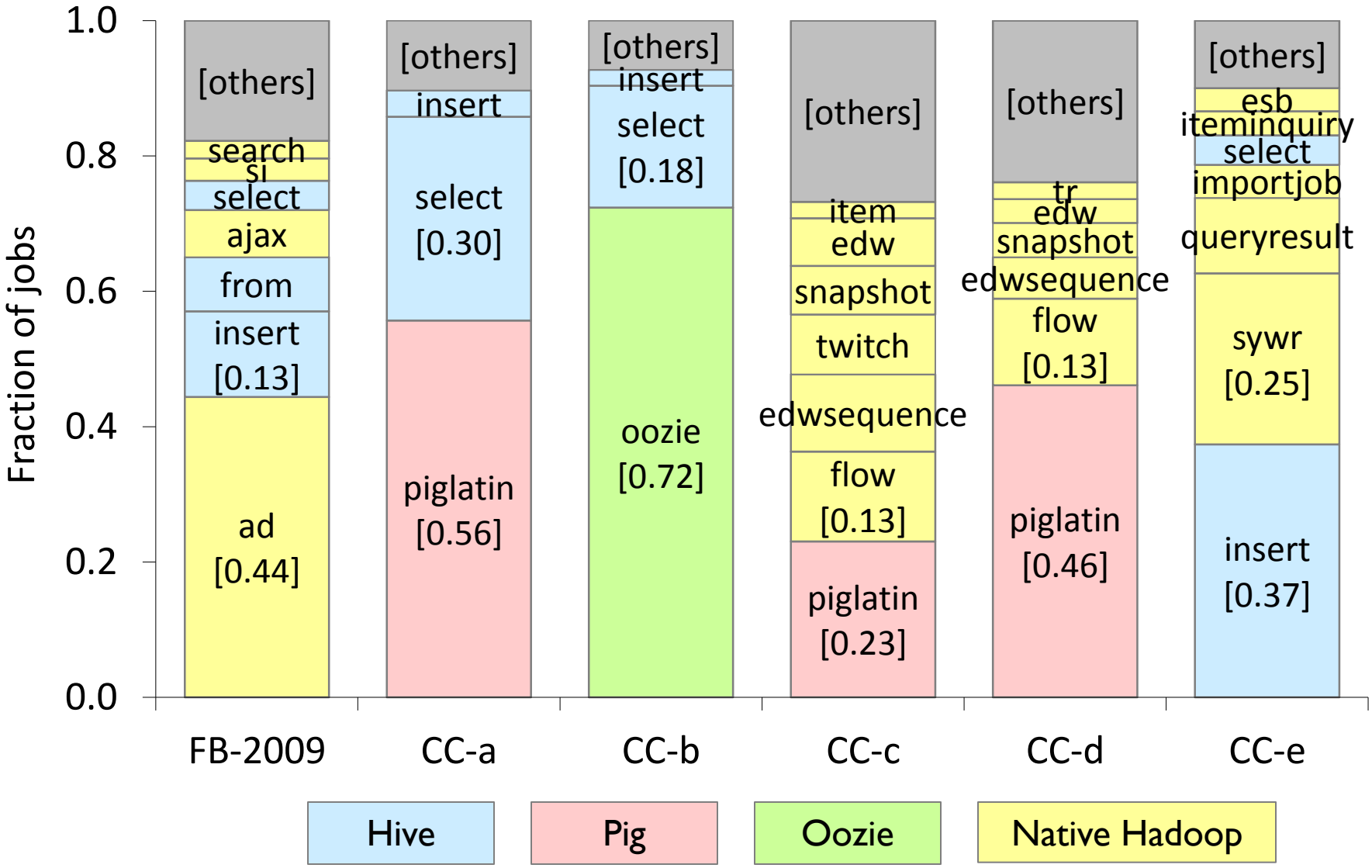
data patterns
- typical data set size?
- uniform or skewed access?
- temporal locality?

arrival patterns
- regular cycles over time?
- bursts in workload?

compute patterns
- common job types?
- **query-like frameworks (Hive/Pig)?**

cross-workload comparison
- variations between workloads?
- representative characteristics?

Hadoop versus Hive/Pig/Other



Hadoop versus Hive/Pig/Other

- Two frameworks make up most jobs in each workload
  - human learning capacity?
  - preference for "mature" frameworks?
- 20-80% of hive/pig/oozie in each workload
  - human analysts move to higher level abstractions?
  - abstractions higher than hive/pig?

Covered in This Talk

data patterns
- typical data set size?
- uniform or skewed access?
- temporal locality?

arrival patterns
- regular cycles over time?
- bursts in workload?

compute patterns
- common job types?
- query-like frameworks (Hive/Pig)?

cross-workload comparison
- **variations between workloads?**
- **representative characteristics?**

Cross-Workload Comparison

- Every workload is different
  - different distributions of data size
  - 80-1 to 80-8 rule
  - 20-80% jobs re-access existing data
  - 20-80% jobs from Hive and/or Pig and/or Oozie
  - etc.

- Should be cautious to declare "representative" behavior

Summary

data patterns
- 80% jobs have < TB data sizes
- Zipf-skew, 80-1 to 80-8 rule
- re-accesses within 3 hrs

arrival patterns
- unpredictable variation over time
- 9:1 to 260:1 peak-to-avg. load ratio

compute patterns
- small jobs are > 90% of all jobs
- hive/pig/oozie form 20-80% of jobs

cross-workload comparison
- wide variation between workloads
- hard to say what is "representative"

What's Next?

- Better performance measurement tools
  - limits of terasort-style tools should be self-evident
  - **Statistical Workload Injector for MapReduce (SWIM)**, able to replay real workloads, used at Cloudera
  - similar tools for Hive/Pig/HBase/Oozie/others?
- Insights feeds into system design
  - Cloudera Distribution with Apache Hadoop (CDH)
  - Cloudera Enterprise
  - Invite the community to contribute



Thanks!!!