IOWA

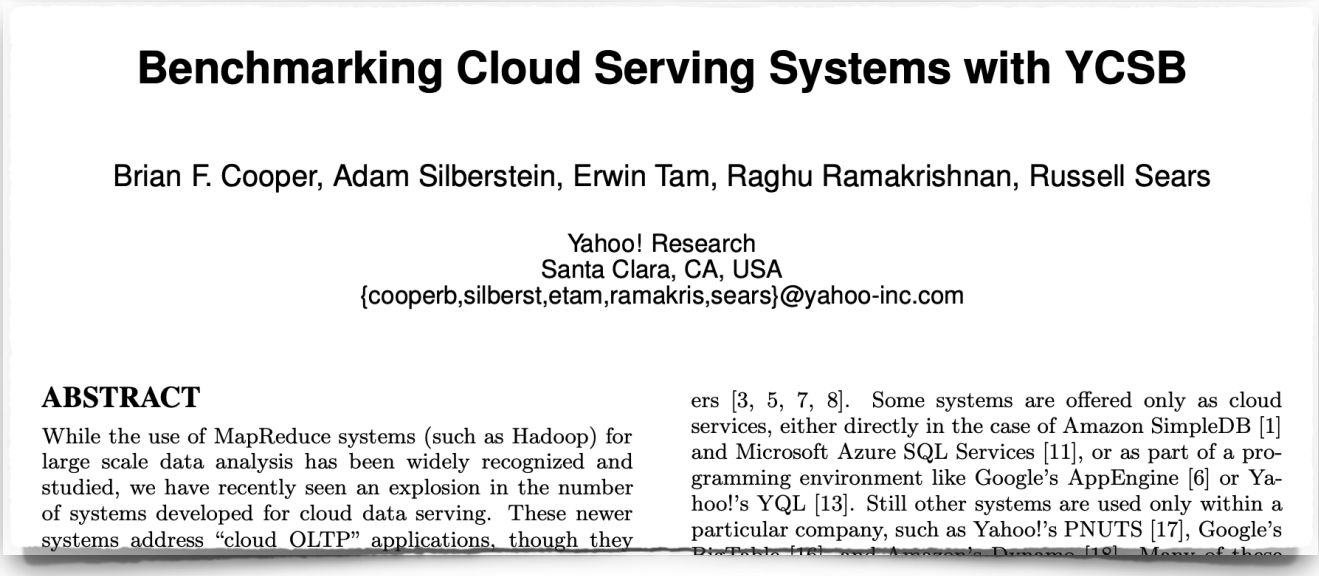
CS5630

Foundation (6): Performance in the Cloud

Prof. Supreeth Shastri
Computer Science
The University of Iowa

Lecture goals

- *Measuring the performance of applications and systems in the **cloud***
- *How to read and digest research papers*



Benchmarking Cloud Serving Systems with YCSB

Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears

Yahoo! Research
Santa Clara, CA, USA
{cooperb,silberst,etam,ramakris,sears}@yahoo-inc.com

ABSTRACT

While the use of MapReduce systems (such as Hadoop) for large scale data analysis has been widely recognized and studied, we have recently seen an explosion in the number of systems developed for cloud data serving. These newer systems address "cloud OLTP" applications, though they

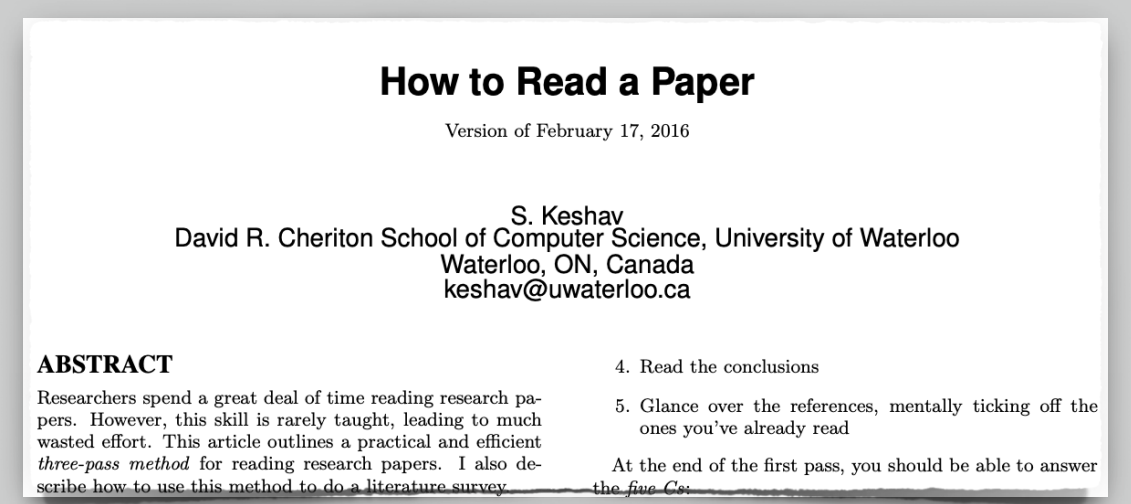
ers [3, 5, 7, 8]. Some systems are offered only as cloud services, either directly in the case of Amazon SimpleDB [1] and Microsoft Azure SQL Services [11], or as part of a programming environment like Google's AppEngine [6] or Yahoo!'s YQL [13]. Still other systems are used only within a particular company, such as Yahoo!'s PNUTS [17], Google's

ACM Symposium on Cloud Computing 2010

IOWA

How to read and *understand* **research papers**

We have papers that help you do that!



How to Read a Paper

Version of February 17, 2016

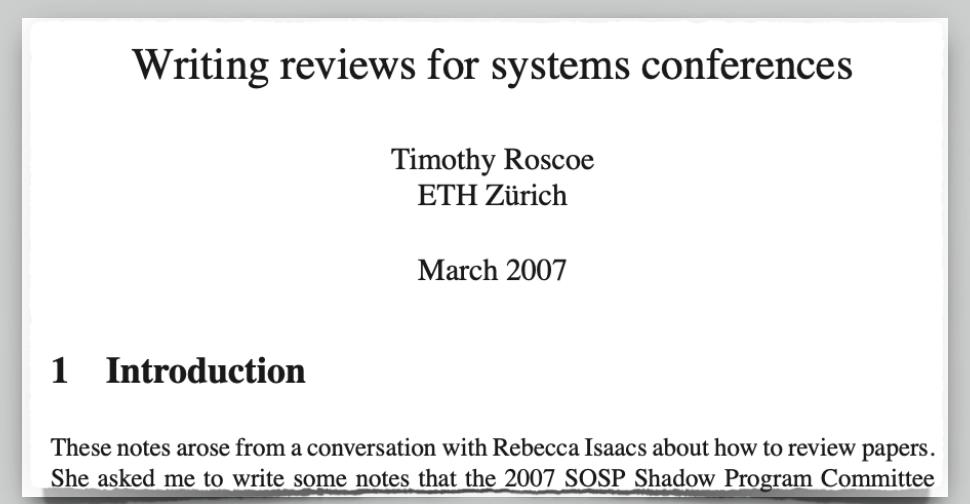
S. Keshav
David R. Cheriton School of Computer Science, University of Waterloo
Waterloo, ON, Canada
keshav@uwaterloo.ca

ABSTRACT

Researchers spend a great deal of time reading research papers. However, this skill is rarely taught, leading to much wasted effort. This article outlines a practical and efficient *three-pass method* for reading research papers. I also describe how to use this method to do a literature survey.

4. Read the conclusions
5. Glance over the references, mentally ticking off the ones you've already read

At the end of the first pass, you should be able to answer the *five Cs*:



Writing reviews for systems conferences

Timothy Roscoe
ETH Zürich

March 2007

1 Introduction

These notes arose from a conversation with Rebecca Isaacs about how to review papers. She asked me to write some notes that the 2007 SOSP Shadow Program Committee

Tips for reading a paper

- Read the entire paper in two/three passes
  - *First pass is to get a birds eye view of the thesis, methods, and findings (**10 mins**)*
  - *Second pass is for a deeper and fuller understanding of all things (**2 hours**)*
  - *Third pass is for those reviewing/presenting the paper (**5-10 hours**)*
- Underline/highlight the important parts of the paper
- Keep notes on the margins about issues/questions
  - *Key insights, questionable claims, relevance to other topics, etc.*
- Look up references that seem to important (or missing)
  - *You may also want to check who references this paper and how*

First pass reading (on the YCSB paper)

- Read the abstract and introduction
  - *For other parts of the paper, just read the section and sub-section headings*

- After the first pass, you should be able to identify
  - *what is the category of the paper (measurements, theory, systems etc)*
  - *what is the main challenge addressed by the paper*
  - *what are its key insights & original contributions*
  - *does it appear correct (or its methods look reasonable)*
  - *should you stop or go on to further read the paper*

ABSTRACT

While the use of MapReduce systems (such as Hadoop) for large scale data analysis has been widely recognized and studied, we have recently seen an explosion in the number of systems developed for cloud data serving. These newer systems address "cloud OLTP" applications, though they typically do not support ACID transactions. Examples of systems proposed for cloud serving use include BigTable, PNUTS, Cassandra, HBase, Azure, CouchDB, SimpleDB, Voldemort, and many others. Further, they are being applied to a diverse range of applications that differ considerably from traditional (e.g., TPC-C like) serving workloads. The number of emerging cloud serving systems and the wide range of proposed applications, coupled with a lack of apples-to-apples performance comparisons, makes it difficult to understand the tradeoffs between systems and the workloads for which they are suited. We present the *Yahoo! Cloud Serving Benchmark* (YCSB) framework, with the goal of facilitating performance comparisons of the new generation of cloud data serving systems. We define a core set of benchmarks and report results for four widely used systems: Cassandra, HBase, Yahoo!'s PNUTS, and a simple sharded MySQL implementation. We also hope to foster the development of additional cloud benchmark suites that represent other classes of applications by making our benchmark tool available via open source. In this regard, a key feature of the YCSB framework/tool is that it is extensible—it supports easy definition of new workloads, in addition to making it easy to benchmark new systems.

- *there is a new category of systems*
- *which run new types of workloads*
- *but, there is no good way to measure their performance*
- *so, we built a new benchmark*
- *and made it to work for real systems*
- *Oh btw, it also has good properties and cool features!*

Now, let us take 5 minutes to read the Introduction section

Did you understand the paper better now? Did you learn anything new or deeper?

How about we take 2 hours to do the **second pass** on the paper?

YCSB: A Primer on Cloud Scale Systems

Key characteristics

- *Scale out* *(i.e., scale by using large number of servers)*
- *Elasticity* *(i.e., varying system's capacity over time)*
- *High availability* *(∵ outages are frequent and affect many users)*

Key tradeoffs

- *Read vs. write performance*
- *Latency vs. durability*
- *Synchronous vs. asynchronous replication*

	Read vs. write	Latency vs. durability	Sync vs. async
PNUTS	Read	Durability	Async
BigTable	Write	Durability	Sync
HBase	Write	Latency	Async
Cassandra	Write	Tunable	Tunable
MySQL	Read	Tunable	Async

YCSB: Workloads

	Operations	Example applications
A - update heavy	read 50%, update 50%	Session store recording user actions
B - read heavy	read 95%, update 5%	Photo tagging
C - read only	read 100%	Browsing user profiles
D - read latest	read 95%, insert 5%	User status on social media
E - range query	scan 95%, insert 5%	Threaded conversations or Q&A
F - read-modify-write	read 50%, rd-mdf-wrt 50%	User database

Benchmark metrics: I/O throughput, latency, running time, and correctness

YCSB: 11 Years later

Emerged as the *de facto* cloud benchmark

- *Open source (2100+ forks, 163 contributors on GitHub)*
- *New data systems (currently integrated w/ 35 data stores)*
- *New workloads (several new generic/custom workloads)*

Of course, it is not a panacea

- *Benchmark metrics are fixed*
- *Not all data-serving systems could be benchmarked (for e.g., AI/ML applications)*

Announcements

Paper Presentation

A list of research papers is up on the course website

Look for a teammate (e.g., on Slack)

Papers are organized along four topics; they should be presented in that order

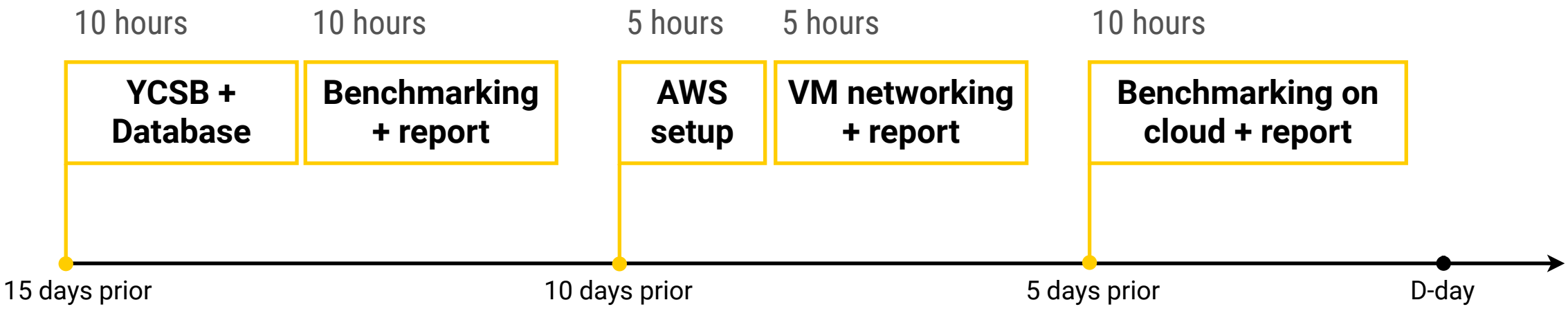
Programming Projects

Project-1 is up (due on **9/30**)

You pick a datastore or database, and benchmark its performance

Hands-on with IaaS cloud: learn to configure and use AWS EC2 VMs

Project planning



Expect to spend **~40 hours** (YMMV)

Paper presentation

30 mins **presentation** + **15** mins **discussion**

- Both team members will participate equally and actively in all tasks
- Both members get the same grade (unless there are discrepancies)
- Submit your talk slides and paper reviews for feedback. This needs to be done at least by the weekend before your talk

Tasks	Weight
Paper review	25%
Presentation	50%
Handling Q&A	25%

Remember that this activity makes up 20% of your final grade

How to prepare for your research talk

- You assume the role of the primary authors of the paper
  - *This is not as hard as it sounds 'cause you're not inventing the work, just developing expertise on it*

- Explore and learn from existing resources
  - *Most conferences archive their talk slides and multimedia*
  - *Many of these papers are classics ==> other students/faculty may have given talks on them*
  - *Authors usually create extra content such as websites, Q&A, expanded tech reports etc*
  - *Don't be shy to engage with the original authors; they love hearing from well-read readers*

- Few more advice
  - ***DO NOT copy slides** from other talks; it is ok to be inspired by them*
  - *Use Iowa branded slide template*
  - *Practice your talk several times*

How to write a good review

- The first step is to read the paper (and read it well)

- Writing down your review
  - *First, summarize the paper in your own words (neutral not judgmental)*
  - *Next, identify and evaluate the key contributions of the paper*
  - *Third, give specific comments on the content of the paper including its novelty, technical strength, limitations, and writing style*
  - *Fourth, describe the impact of the paper on the field (a retrospective view)*
  - *Finally, write your conclusion (brief; could include subjective opinions)*

- Few more advice
  - *Read sample reviews linked on the course site*
  - *Keep the writing style and tone professional*
  - *Ideal review length is between 500-800 words*

Spot Quiz (ICON)