Internet-scale Distributed Systems Seminar

Table of Content

Introduction

Large-scale, distributed systems have become the foundation of today's Internet applications, easily processing petabytes of data every day. In this seminar, we explore these systems. We take a look at consistency, distributed file systems, messaging and locking-services, entity caches, different types of databases, data warehouses, and data center operations, etc..

Seminar Operation

Seminar participants are organized into teams of two students. We will assign topics to teams and we will assign students to teams, unless a team has already been formed.

Each team has to provide a ten minute (no more, no less) *video* along with a *set of presentation slides* about their assigned topic (cf. exact Deliverables Specification, below). The video should present the most important aspects of the assigned topic in a concise, didactically valuable, and informative manner. We require that each team member is visible throughout at least 50% of the video, i.e., the person's face is visible next to other presentation material; the remaining 50% of the time, slides, blackboard, or other artifacts that illustrate the subject matter should be visible.

The video may be recorded with a Smartphone camera, a Webcam or other recording equipment. The video may include elements recorded with screen capture software (cf. Available Tools, below). For the sake of simplicity, you do not necessarily have to show the presentation slides in the video, explaining the topic through illustrations on a whiteboard, a flip chart, or by other means of your choosing, -- as long as the key aspects of the assigned topic are conveyed in a concise, didactic and informative manner, -- is sufficient.

The video has to be submitted exactly two weeks before the date of your session (cf. Schedule, below). There are about 5 teams submitting per week. Each week, we run two presentations in the weekly contact hour. To fill these slots, we will select two teams from among the five submissions for the two available presentation slots each week. The selection is based on the quality of the submitted material, evaluating both, the content and the presentation (i.e., the material considered are the video and the slides). The chosen teams will also receive the best marks, according to our scheme. They will present their topic during the assigned weekly session in a 45 minute slot (including 15 minutes of Q&A and discussion of the topic by the audience, which must be facilitated by the team, e.g., through raising discussion points at the end of their presentation.)

At the completion of the seminar, we plan to award a prize to the best team, which may take feedback from course attendees into consideration by allow them to vote from among the weekly submitted videos.

In addition to the video and the presentation slides, each team has to also submit a written report about their topic. The report is due two weeks after the date of the team's session (see Deliverables, below).

Preparation

You will prepare for your topic deliverables as a team. Marks are assigned by team, unless team members drop out. It is expected that each team member does his or her fair share of the work; the team must self-govern to enforce this rule. Use the ACM digital library for your research, e.g., http://dl.acm.org/, with free access from within the university network or via VPN into the university.

Expectations

Each team is expected to conduct their own research to collect a sufficient amount of information about their topic, either via the Internet, digital libraries, or by consulting the university library.

It is strictly forbidden to copy content from any source material, this includes the copying of slides, the copying of text, the copying of diagrams and graphs. All material presented should have been designed by the team members themselves. All references used must be disclosed. Should this practice not be followed, mark penalties will apply. In severe cases of plagiarized content, the team will be assigned a failing grade.

The team should aim to visualize ideas, concepts, protocols, and system architectures described in the literature of their topic by resorting to tools and techniques known by the team from their studies, e.g., UML diagrams, sequence diagrams, state charts, etc. Also, the team should consider to convey elements such as the logical architecture, the physical architecture, the read/write paths through the system, the different processes and states of the system or protocol, etc. .
Slides should not be cluttered with a lot of text or a lot of bullets, instead, slides should be simple and clear. For example, resort to animations or other effects to illustrate more complex interactions and protocols.

We are looking forward to watch plenty of nice presentations!

Seminar Information

- All seminar related information is posted on Moodle
- Please take a look at the documents before writing an Email

Room

- Interim Hörsaal 1 (5620.01.101)

Grading

- Grade assignment is done by team
- Course grade is based on the submitted material and the potential in-class presentation
- While we aim to select the two best submissions as candidates for in-class presentation, we may sometimes decide otherwise

Attendance

- Full attendance required every week
- One date of absence is allowed, more should be clarified with us
- Signature list passed around during selected weeks

Deliverables

- 10 minutes video presentation
    - Faces of both team members, - one at a time, - must be visible throughout at least 50% of the presentation
    - Team members may record each other with a smartphone or use of a Webcam
    - For the video recording, topic content may be explained via a white- or blackboard or via projecting slides. If a team has no projection capabilities, projecting slides is not a requirement; we also encourage innovative and original ways of conveying the topic, provided they resemble one of the listed means (i.e., the most important technical aspects of the topic are adequately illustrated and covered in the video recording.)
    - Video should be edited
    - Video must be uploaded in either .avi or .mpeg format
- 10 slides
    - Use either powerpoint or pdf
    - Entirely designed by students, no plagiarized content
    - Should a team decide to explore innovative ways to convey the technical aspects of the topic, a format different than slides to document the illustration of the work may be permissible, provided the material is professionally convey and can be digitally captured and uploaded
    - Upload material in time according to the specified schedule
- Provide a full presentation, if submission is picked
    - 25-30 minutes in-class presentation plus 15 Q&A (it is the team's responsibility to encourage and manage Q&A, e.g., by raising discussion points, etc.); the team may include their video as part of the presentation time, if desired
- Write a report about the topic
    - (6 pages, ACM proceedings style) www.acm.org/sigs/publications/proceedings-templates
    - Report should include references, and be of professional quality, e.g., no typos, few grammar mistakes, adequate use of diagrams, tables, and illustrations, no plagiarized content)

Submission

- There are three separate tasks on Moodle to submit your video, presentation and report.
- Because every team has different deadlines, Moodle will not remind you of the deadline. Please take care by yourself
- Name the uploaded files according to the following schema
    - Topic_GroupNo

Schedule

- Video and presentation slides are due two weeks before your session (exactly 14 days before, cut-off time is 23:59 o'clock)
- Selected teams are announced one week before their session
- Report is due two weeks after your session (exactly 14 days after, cut-off time is 23:59 o'clock)

Topic	Date	Time
Kick-Off session	**15.04.2015**	**13:00**
CAP Theorem	**22.04.2015**	**13:00**
Consistency Models	**22.04.2015**	**13:45**
Lock Services: Zookeeper vs. Chubby	**29.04.2015**	**13:00**
Messaging: Thialfi vs. Kafka	**29.04.2015**	**13:45**
File Systems: GFS vs. HDFS	**06.05.2015**	**13:00**
DB1: Bigtable vs. HBase	**06.05.2015**	**13:45**
DB2: Dynamo / Consistent Hashing	**13.05.2015**	**13:00**
DB3: Cassandra vs. PNUTS	**13.05.2015**	**13:45**
DB4: CouchDB and MongoDB	**20.05.2015**	**13:00**
DB5: SAP Hana vs. H-Store	**20.05.2015**	**13:45**
Caching: Redis and memcached	**27.05.2015**	**13:00**
Hadoop1: Map-Reduce and YARN	**27.05.2015**	**13:45**
Hadoop2: HadoopDB and Apache Tez	**03.06.2015**	**13:00**
Hadoop3: Apache Spark and Pig	**03.06.2015**	**13:45**
DW1: Apache Hive vs. Shark	**10.06.2015**	**13:00**
DW2: Impala vs. Presto	**10.06.2015**	**13:45**
DW3: Google's Dremel and Facebook's Scuba	**17.06.2015**	**13:00**
ML: MLBase vs. Mahout	**17.06.2015**	**13:45**
Graph: Pregel vs. Giraph	**24.06.2015**	**13:00**
Events: Apache Storm and Borealis	**24.06.2015**	**13:45**
Benchmarking: YCSB vs. BigBench	**01.07.2015**	**13:00**
Monitoring: Ambari and Chukwa	**01.07.2015**	**13:45**
Platform: Cloudera and Hottenworks	**08.07.2015**	**13:00**
Google's Spanner and Megastore	**08.07.2015**	**13:45**
Google's Percolator and Dapper	**15.07.2015**	**13:00**
Google vs. Facebook data centers	**15.07.2015**	**13:45**

Exception handling

- We will not change your assigned topic or group. There are more frequently requested topics. Not everybody will get her/his preferred topic. Likewise, the schedule is fix and cannot be changed.
- Member leaves team
  - **All above deliverables** have to be fulfilled by remaining team member, except the report writing may be limited to 3 pages
  - Leaving member will receive a grade of 5
  - Remaining member will be graded based on his/her performance
- Late submission of video and slides
  - Counting from the deadline, every starting 24 hour period material is submitted late, a penalty of 1.0 is incurred
- No video is submitted
  - A grade of 5 is assigned to team
- No slides are submitted
  - A grade no better than 2 can be achieved, based on the submitted video and possible presentation
- No report is submitted
  - A grade no better than 3 can be achieved, based on the submitted video and possible presentation
- Chosen team does not appear for presentation
  - A grade of 4 will be assigned, given that video & slides have been submitted

Open-source and Free Tools

- VSDC Free Video Editor (http://www.chip.de/downloads/VSDC-Free-Video-Editor_67265703.html)
- AVS Video Editor (http://www.chip.de/downloads/AVS-Video-Editor_38380978.html)
- VLC Player als Screencast-Tool (http://www.chip.de/downloads/VLC-media-player-32-Bit_13005928.html)
- CamStudio Screencasts (http://www.chip.de/downloads/CamStudio_19900258.html)
- Capture Fox Screencasts for Firefox (http://www.chip.de/downloads/Capture-Fox_42736096.html)

List of Selected Papers for Most Topics

Here you can find papers related to your topic. Not every topic is covered. We work on completing the list. However, the list should not replace your literature search.

The CAP Theorem

- E. Brewer, Towards Robust Distributed Systems, in PODC Keynote Talk, 2000.
- E. Brewer, CAP Twelve Years Later: How the "Rules" Have Changed, in IEEE Computer 45(2), 2012
- S. Gilbert, N. Lynch. Brewer’s conjecture and the feasibility of consistent, available, partition-tolerant Web services. ACM SIGACT News 33(2), 2002.

Consistency Models

- D. Terry. Replicated Data Consistency Explained Through Baseball, in Communications of the ACM 56(12), pages 82-89, 2013
- H. Yu, A. Vahdat, Design and Evaluation of a Continuous Consistency Model for Replicated Services, in OSDI 2000.
- A. Tannenbaum, M. van Steen, Distributed Systems - Principles and Paradigms, 2ed, Prentice-Hall, Inc, Chapter 7. 2007.

Lock Services: Zookeeper vs. Chubby

- M. Burrows. The Chubby Lock Service for Loosely-Coupled Distributed Systems. In OSDI ‘06: 7th Symposium on Operating Systems Design and Implementation, pages 335-350, 2006.
- P. Hunt, M. Konar, F. P. Junqueira, and B. Reed. ZooKeeper: Wait-Free Coordination for Internet-Scale Systems. In USENIX ATC ‘10: Proceedings of the 2010 USENIX annual technical conference, 2010.

Messaging: Thialfi vs. Kafka

File Systems: GFS vs. HDFS

- S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google file system. In SOPS ‘03: Proceedings of the 19th ACM Symposium on Operating Systems Principles, pages 29-43, 2003.
- K. Shvachko, H. Kuang, S. Radia, R. Chansler. The Hadoop Distributed File System. In IEEE 26th Symposium on Mass Storage Systems and Technologies, pages 1-10, 2010

DB1: Bigtable vs. HBase

- F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. BigTable: A distributed storage system for structured data. In OSDI '06: 7th Symposium on Operating Systems Design and Implementation, pages 205-218, 2006.

DB2: Dynamo / Consistent Hashing

- G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In SOSP '07: Proceedings of the 21st ACM Symposium on Operating Systems Principles, pages 205-220, 2007.
- D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing, 1997.

DB3: Cassandra vs. PNUTS

- A. Lakshman and P. Malik. Cassandra: a decentralized structured storage system. SIGOPS Operating Systems Review, 44(2):35-40, 2010.
- J. Ellies. Leveled Compaction in Cassandra. Website.
- L. Alberton. Modern Algorithms and Data Structures - 1. Bloom Filters, Merkle Trees. Cassandra-London. Presentation. 2011.
- B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H.-A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. PNUTS: Yahoo!'s hosted data serving platform. Proceedings of the VLDB Endowment, 1(2):1277-1288, 2008.
- A. Silberstein, J. Terrace, B. Cooper, and R. Ramakrishnan. Feeding Frenzy: Selectively Materializing Users' Event Feeds. Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, 2010.

DB4: CouchDB and MongoDB

DB5: SAP Hana vs. H-Store

Caching: Redis and memcached

Hadoop1: Map-Reduce and YARN

- J. Dean and S. Ghemawat. MapReduce: Simplied data processing on large clusters.In OSDI '04: Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation. 2004.

Hadoop2: HadoopDB and Apache Tez

Hadoop3: Apache Spark and Pig

- C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins. Pig Latin: A Not-so-foreign Language for Data Processing. In SIGMOD '08: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pages 1099-1110, 2008.
- M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M.J. Franklin, S. Shenker, and I. Stoica. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. In NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation.
- Alan Gates. Comparing Pig Latin and SQL for Constructing Data Processing Pipelines. 2010.

DW1: Apache Hive vs. Shark

- A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murth. Hive: A Petabyte Scale Data Warehouse Using Hadoop. In ICDE '10: Proceedings of the International Conference on Data Engineering, pages 996-1005 , 2010.
- R. Xin, J. Rosen, M. Zaharia, M. Franklin, S. Shenker, I. Stoica. Shark: SQL and Rich Analytics at Scale. In ACM SIGMOD'13.

DW2: Impala vs. Presto

DW3: Google's Dremel and Facebook's Scuba

- S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis. Dremel: Interactive Analysis of WebScale Datasets. Proceedings of the VLDB Endowment, 3(1-2):330-339, 2010.

ML: MLBase vs. Mahout

- T. Kraska, A. Talwalkar, J.Duchi, R. Griffith, M. Franklin, M.I. Jordan. MLbase: A Distributed Machine Learning System. In Conference on Innovative Data Systems Research , 2013.
- E. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, J. Gonzalez, J. Gonzalez, M. Franklin, M. I. Jordan, T. Kraska. MLI: An API for Distributed Machine Learning. In International Conference on Data Mining, 2013.
- J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton. MAD Skills: New Analysis Practices for Big Data. PVLDB 2(2). 2009.

Graph: Pregel vs. Giraph

- Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, J. M. Hellerstein. Distributed GraphLab: A Framework for Machine Learning in the Cloud. PVLDB 5(8), 2012.
- G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, G. Czajkowski. Pregel: A System for Large-scale Graph Processing. SIGMOD 2010.

- D. Nguyen, A. Lenharth, K. Pingali. A Lightweight Infrastructure for Graph Analytics. SOSP '13.

Events: Apache Storm and Borealis

- D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M Cherniack, J.-H. Hwang, W Lindner, A. S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik: The Design of the Borealis Stream Processing Engine. In 2nd Biennial Conference on Innovative Data Systems Research, 2005.
- M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, I. Stoica: Discretized Streams: Fault-Tolerant Streaming Computation at Scale. SOSP '13.
- Nathan Marz: Storm

Benchmarking: YCSB vs. BigBench

- B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking cloud serving systems with YCBS. In SoCC '10: Proceedings of the 1st ACM Symposium on Cloud Computing, pages 143-154, 2010.
- S. Patil, M. Polte, K. Ren, W. Tantisiriroj, L. Xiao, J. Lopez, G. Gibson, A. Fuchs. YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores. In SoCC '11: Proceedings of the 2nd ACM Symposium on Cloud Computing, 2011.
- A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, and H.-A. Jacobsen. BigBench: Towards an Industry Standard Benchmark for Big Data Analytics. In SIGMOD '13: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pages 1197-1208, 2013.

Monitoring: Ambari and Chukwa

Platform: Cloudera and Hottenworks

Google's Spanner and Megastore

- J. Baker, C. Bond, J. Corbett, J. J. Furman, A. Khorlin, J. Larson, J.-M. Leon, Y. Li, A. Lloyd, and V. Yushprakh. Megastore: Providing Scalable, Highly Available Storage for Interactive Services. In CIDR ‘11: Fifth Biennial Conference on Innovative Data Systems Research, pages 223-234, 2011
- J.C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, JJ. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford. Spanner: Google's Globally-Distributed Database, In OSDI'12.

Google's Percolator and Dapper

Google vs. Facebook data centers