Internet-scale Distributed Systems Seminar

Table of Contents

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 **s*et 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 one week before the date of your session** (cf. Schedule, below).

Each week, we run three presentations in our session. **You will present your topic (and your video) during one of these 30 min slots.** Please prepare 10 minutes video + 20 minutes presentation. Think of creative ways to integrate the video into your presentation. It may be used as an introduction, a hands-on tutorial or a summary, etc...

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 (except company logos and motivational pictures) 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
- Take a look at the documents before sending an e-mail; we will not respond to questions that are answered in information we already made available
- If you have further questions, please approach the instructor at the end of the lessons (we cannot entertain one-on-one correspondence by email)
- Constructive criticism is welcome. However, we will not respond to anonymous complaints. Please use your official TUM Email-account.

Room

- MI - Hörsaal 3

ECTS

- 5 points

Grading

- **Grade assignment is done by team**
- Course grade is based on (You find detailed judgement criteria in the grading rubric):
    - 25 % Video
    - 25 % Presentation
    - 25 % Slides
    - 25 % Report

Attendance

- Full attendance required every week
- One absence is permissible, signature lists will be passed around, and spot-checks may be conducted;
- 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 link must be uploaded.
- 20 - 30 slides (approx. 10 slides for video + 10 - 20 for the full presentation)
    - Submission as 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
- Write a report about the topic
    - (6 full pages, ACM proceedings style, including references) www.acm.org/sigs/publications/proceedings-templates ; please use sample-sigconf.tex
    - 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)

Presentation Tips

- Start with a nice introduction, explain shortly: Which system are you presenting? Who developed the system? What is the purpose of the system? In which projects is the system used at the moment

- Follow up with top level explanations. Show that you have good overall knowledge. Despite of that, prepare some slides (later in the presentation), where you dive into the details. But don't overload the audience with your expertise, 2-3 slides about detailed aspects are enough.

- System visualization:
    - **Architectural view:** A picture presenting the top level view of the architecture. This is the perspective of a software engineer. You can show the components of the system and explain their tasks. For a distributed file system you could show the master and the slave nodes. But only on top level. No detailed descriptions here!
    - **Conceptual view:** You can show a picture presenting the logical concepts of the system. This is usually the perspective of a programmer, using the systems API. Presenting a distributed file system, you could show that files are stored under a certain path in a hierarchical manner, just like in a normal file system.
    - **Physical view:** You can show how the logical concepts are mapped to the underlying hardware. This is the perspective of the administrator. Presenting a distributed file system, you would depict a cluster with x nodes. Then you could explain, where the master and where the slaves are running. Moreover, you could show, how replication is achieved with the help of multiple machines.

- System processes:
    - **Write Path:** You can show how a client update travels through the architecture. Start with the client, issuing the update, and then explain, the subsequent steps until the update is persisted safely.
    - **Read Path:** You can show, what happens when the client requests a certain resource from the system. How the query is executed, how the system is reading information from disk/memory.

- Include a little analysis about consistency, performance and fault tolerance if it makes sense. These properties are usually very important in distributed systems. If your system is a database, refer to the **CAP theorem**.

Report Tips

- Here is some general stuff about how to write a paper: https://sdqweb.ipd.kit.edu/wiki/Wissenschaftliches_Schreiben. If you don't know German, there are further links at the bottom of that page. You can also find plenty of pages about how to write a paper on the internet.
- You are not supposed to write a complete paper. A report is a small version of a paper.
- Here are some hints for the content:
    - The abstract should compress the most interesting aspects of your system. It should arouse the curiosity of the reader.
    - Like in the presentation the introduction should answer the following questions: Which system are you presenting? Who developed the system? What is the purpose of the system? In which projects is the system used at the moment?
    - Build the sections on top of each other. Good papers/reports are often compared to a pyramid, starting at a small top and going down to a broad basis.
    - Create your own pictures, don't copy from papers. Keep everything in black and white if possible. (If you are smart, you use the same pictures for report and presentation. You can style them in the presentation)
    - Prove some of your statements with references. Keep your references clean at the bottom of the report.
    - Don't use terms that are not explained. As soon as you introduce a new term enclose it with \textit{} and explain it.
    - Write a short explanation at the beginning of every section. Describe what the section is about.
    - End up with a conclusion and summarize your findings

Submission

- There are three separate tasks on Moodle to submit your video link, presentation slides (full presentation) and report.
- Provide access to your video (via GD/Dropbox or web server); upload the link in a text file. Make sure your video is both available and accessible for your tutor (don't forget to grant the file permissions).
- Because every team has different deadlines, Moodle won't remind you of the deadline
- Name the uploaded files according to the following schema: "Topic_GroupNo"

Schedule

- Video and presentation slides are due one week before your session **(exactly 7 days before**, cut-off time is 23:59 o'clock).
- Report is due two weeks after your session (exactly 14 days after, cut-off time is 23:59 o'clock)

	Topic	Date	Time	Room
	Goals and Expectations	**24.02.2020**	**14:00**	**HS-3**
1	**CAP Theorem**	**08.05.2020**	**14:00**	**HS-3**
2	**Consistency Models**	**08.05.2020**	**14:30**	**HS-3**
3	**Formats: JSON, PB, Avro, Thrift, Trevni, Parquet**	**08.05.2020**	**15:00**	**HS-3**
4	**Distributed Services: Zookeeper vs. Chubby**	**15.05.2020**	**14:00**	**HS-3**
5	**Datacenters: Google vs. Facebook**	**15.05.2020**	**14:30**	**HS-3**

6	DB1: Bigtable vs. HBase	15.05.2020	15:00	HS-3
7	DB2: Dynamo vs. Cassandra	22.05.2020	14:00	HS-3
8	DB3: Spanner and Megastore	22.05.2020	14:30	HS-3
9	DB4: CouchDB vs. MongoDB	22.05.2020	15:00	HS-3
10	DB5: SAP Hana vs. RAMCloud	29.05.2020	14:00	HS-3
11	Logging: Bookkeeper and DistributedLog	29.05.2020	14:30	HS-3
12	Caching: Redis vs. memcached	29.05.2020	15:00	HS-3
13	Computing models: MapReduce vs. BSP	05.06.2020	14:00	HS-3
14	Analytics1: Hive and Tez	05.06.2020	14:30	HS-3
15	Analytics2: Redshift vs. Dremel	05.06.2020	15:00	HS-3
16	Analytics3: Impala vs. Presto	12.06.2020	14:00	HS-3
17	Analytics4: Spark vs. Drill	12.06.2020	14:30	HS-3
18	Analytics5: Druid vs. Scuba	12.06.2020	15:00	HS-3
19	Benchmarking: YCSB vs. BigBench	19.06.2020	14:00	HS-3
20	Machine Learning1: MLBase vs. Mahout	19.06.2020	14:30	HS-3
21	Machine Learning2: Tensorflow vs. Torch	19.06.2020	15:00	HS-3
22	Graphs: Pregel vs. Giraph	26.06.2020	14:00	HS-3
23	Messaging1: Thialfi vs. Kafka	26.06.2020	14:30	HS-3
24	Messaging2: ActiveMQ vs. RabbitMQ	26.06.2020	15:00	HS-3
25	Events: Strom, Trident and Flink	03.07.2020	14:00	HS-3
26	Configuration Management: Puppet, Chef	03.07.2020	14:30	HS-3
27	PAXOS: Fast, Multi, Byzantine Paxos	03.07.2020	15:00	HS-3
28	Searching: Lucene, Solr vs. Sphinx	10.07.2020	14:00	HS-3
29	Recommendation: Metaphor and Netflix prize	10.07.2020	14:30	HS-3
30	Clouds: AWS vs. Azure vs. Google Cloud	10.07.2020	15:00	HS-3
31	Virtualization: KVM vs. VirtualBox vs. Docker	17.07.2020	14:00	HS-3
32	Network virtualization: VXLAN vs. NVGRE	17.07.2020	14:30	HS-3
33	Resource Manager: Kubernetes, YARN	17.07.2020	15:00	HS-3
34	Resource Manager: OpenStack, Mesos	24.07.2020	14:00	HS-3
35	File Systems: GFS vs. HDFS vs. Ceph Bluestore	24.07.2020	14:30	HS-3
36	Parallel File Systems: BeeGFS vs Lustre	24.07.2020	15:00	HS-3

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, the presentation limited to 20 min
    - Leaving member will receive a grade of 5
    - Remaining member will be graded based on their 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.

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.

- HBase: The definitive guide
  http://www.nataraz.in/data/ebook/hadoop/HBase.The.Definitive.Guide.pdf

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

- Michael Leben, CouchDB - relaxed web developement
- J. Chris Anderson et al. CouchDB - The definitive guide
- Homepage MongoDB: http://www.mongodb.org/,
- MongoDB S3 - Architecture guide

DB5: SAP Hana vs. H-Store

- Franz Färber et al. SAP HANA Database - Data management for modern business applications
- Analysis of SAP Hana high availability capabilities
- Robert Kallman et al., H-Store: A High-Performance, Distributed Main Memory Transaction Processing System

Caching: Redis and memcached

- Matti Paksula, Persisting Objects in Redis Key-Value Database
- Homepage memcached: http://memcached.org/
- Rajesh Nishtala, Hans Fugal, Steven Grimm et al. Scaling Memcache at Facebook

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

- Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel Abadi, Avi Silberschatz, Alexander Rasin, HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads
- Prakasam Kannan, Beyond Hadoop MapReduce Apache Tez and Apache Spark

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

- Impala: A modern Open-Source SQL engine for Hadoop http://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper28.pdf
- Presto Homepage: https://prestodb.io/
- Presto architecture: https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920

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 Trident

- 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.

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.

Document Revisions

Date	**Revisions**
March 20th	Added highlighting and clarified instructor-student/course communication
April 13th	specified deadline of real presentation,

	defined logo and motivational pictures as exception of content plagiarism
May 4th	Added literature
May 7th	Clarified video format, submission rules
August 27th, 2015	Updated schedule
October 14th, 2015	Updated grading, seminar information
March 11th, 2016	Update schedule, seminar information
April 13th, 2016	Added a presentation and report tips section
February 27th, 2017	Update schedule, seminar information
January 23th, 2020	Formatting, fixing grammatical errors
February 4th, 2020	Updated schedule, seminar information