Assignment Two for CS5223

July 2018

This assignment should be done in teams, where each team can have at most 3 students. Throughout the document, "you" means "your team".

1. Overview

The goal of this literature survey assignment is to train your ability in summarizing and judging research papers in the distributed systems area based your own understanding. For research-oriented Ph.D. students, this is a prerequisite for doing good research and writing your own research papers. For Master's students, even though you will not write research papers yourself, you will still be expected to be able to follow new research results and research trends, and potentially adopt the research results in your company's products. In fact, in many research conferences, many attendees are actually people who don't write papers themselves -- they attend the conference to learn about new techniques.

2. Instructions

At the end of this document, you will find a list of topics (A through E). The topics will deepen your knowledge on some of the issues not covered in the lectures.

After you pick a topic, you should read all papers listed under that topic. In some cases, I have included hyper-links. For those papers that I did not provide a link, it is your job to google those or to find those from various online digital library. This is part of the literature survey exercise as well. In particular, **ACM, IEEE, and Springer digital libraries can be accessed free of charge from any NUS machine, through NUS library web site**.

Make sure that you do not miss any paper under the topic. Different topics have different numbers of papers, which is intentional. Those papers under topics with fewer numbers of papers are harder to read. I intentionally adjusted the number of papers so that each topic will incur roughly the same load. You may need to read additional papers (cited by the papers listed) in order to better understand the papers listed.

3. General structure of the report

Overall, the report should summarize the techniques described in the papers listed under the given topic, and then present your subjective opinions about these techniques. You should try not to summarize each paper individually. Rather, whenever possible, you should provide a coherent overall summary, with proper comparisons. You may also decide to discuss more about some papers (because they have interesting techniques) and discuss less about other papers (because they don't really propose many techniques).

Following is the general structure your report should follow:

- You should first provide a **self-contained** overview of all the techniques proposed in the papers. You need to describe the techniques **in your own words**, instead of repeating discussions from the papers. In particular, because each paper was written individually, it is your job to come up with **a unified way** of explaining the techniques from different papers. As a guideline, this part can be 9-12 pages (this is not a requirement though and you may use more or less pages for this part).

- You should then provide your own (subjective) opinion about the pros and cons of the techniques. Some questions you may choose to answer are: What are the nice properties of these techniques? What are potentially problems? How do the technique compare with each other? Is the technique revolutionarily novel or is it just some minor tweaks on existing techniques? Are the techniques practically feasible? Do they make unrealistic assumptions? You can cite results from the papers, or discussions from the papers to support your own arguments. As a guideline, this part can be 2-4 pages (this is not a requirement though and you may use more or less pages for this part).

- You should identify potential research problems that are still open. What do you see as the major drawbacks of all the techniques you have read so far? What would be a reasonable direction to eliminate these drawbacks? As a guideline, this part can be about 1 page (this is not a requirement though and you may use more or less pages for this part).

Following are some more specific requirements:

- Your report must be tutorial and self-contained. The intended audience is someone who already has taken a graduate-level distributed systems course (e.g., your classmates), but who hasn't investigated deep into that particular subject. Being self-contained is very important and will be an important evaluation criterion.

- It should show your understanding of the key concepts and methods within your topic.

- Throughout the survey, you **must use your own wording and formulation**, except when you directly quote sentences from other publications and put it in quotes. In particular, you should not copy the abstract of the paper you are surveying. Note that copying is considered plagiarism.

- I expect direct quoting of sentences from other publication to be rather rare (you seldom see this in real survey papers).

- If you say something that is not a direct quote, but is stated in some other form in one of the articles, you must refer to that article, e.g., performance of the conservative PDES schemes depends on look-ahead as discussed in [give reference to the source].

- Whether you subjective judgments are intelligent will be one assessment criteria.

- You should include references to all additional articles that you have used in preparing your report.

- It is perfectly ok if the paper has some details that you don't understand, but the important thing is that you get the main idea of the paper.

- Do **not** try to understand every sentence in the papers, otherwise it will take you a whole month just to read one paper. Instead, your goal is to get the main ideas and key techniques/insights/observations of the paper, while ignoring less important details.

4. Resources

ACM Computing Surveys usually have very good survey papers. You can look at those survey papers as models.

5. Report Submission Instructions

You will be penalized if you don't stick to these instructions. The report should be no more than 15 pages, and must use 11-pt Times New Roman font, single spacing, and single column. **All pages must be numbered, starting from 1. Sections must be numbered as well.** The 15-page limit **includes** the title page and the references. It is ok to include an appendix, which will not count toward the 15-page length limit.

Page 1 of your report should be the title page, which should **contain and only contain** the following table filled with proper information:

	Fill in all the entries below		
Topic of the survey (the topic should be one of the topics listed at the end of this document)			
Length of the survey in number of pages (excluding appendix)			
	Member #1	Member #2	Member #3
Team Member's student number (**not nusnetid!**)			

Page 2 should directly begin with Section 1 of your survey. There should be NO table of contents, table of figures, or anything like that. Also, do not list authors etc. on Page 2.

Submission file name: The only acceptable format is PDF. Other formats will be discarded. You report should be a **single file** with a .pdf suffix. The file name must be in the following form (if your team has 2 members):

[Team member 1's matriculation #]_[Team member 2's matriculation #].pdf

For example, if the matriculation numbers of team member 1 is HT111111, and the matriculation number of team member 2 is HT222222, then the file name should be

Deviation from such naming scheme may cause mistakes when processing your submission, and may result in the loss of your submission. In addition, explicit penalty will be imposed if you deviation from such naming scheme.

No multiple submissions allowed: Each team should make sure that the team **only submit exactly once**. If a team submits two versions, we will **retain the earliest version and discard all later versions.** The team will then be grade on the earliest version. For such a reason, if you want to update your submission, you should delete your old submission first and then submit a new one.

Submission directory: Your report should be uploaded to IVLE "Workbin\root\Student Submission for Assignment Two".

Timeline:
15 Oct 2018 Monday: Assignment starts.
11:59pm, 16 Nov 2018 Friday: Assignment due. You should upload your report to IVLE. Electronic submissions will, among other things, facilitate our checking for plagiarism in the submissions. Note that we will do extensive plagiarism check, **which includes, but is not limited to, checking against all resources on the web and checking against student submissions in both this semester and in all previous years. Do NOT try your luck!**

Policy on late submission: Reports submitted after the deadline but no more than 48 hours after the deadline will still be graded, with a penalty of 20%. Namely, I will first grade the report normally, and then multiple the mark by 80% to get the final mark for that report. **Reports submitted more than 48 hours after the deadline will not be accepted and will get 0 mark.**

Policy on "fair share" contribution: Given that this is a team assignment, each team member should contribute his/her "fair share" to this assignment. Namely, if the team has 3 students, then a "fair share" would roughly be 1/3. In the event that a team member is found to have contributed a much smaller amount than his/her "fair share", then the following policy applies: I will first grade the report normally and determine a temporary mark *x*. For all team members who have contributed roughly a fair share, their final mark will be *x*. For any team member who has contributed a much smaller amount than his/her "fair share", his/her final mark will be *c***x*, where *c* is the penalty factor. Depending on the severity of the problem, *c* can be 0.75, 0.5, 0.25, or 0.

List of topics

A. Cloud Computing

[1] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris. "Reining in the outliers in map-reduce clusters using mantri". In Proceedings of the Usenix Symposium on Operating Systems Design and Implemenetation, 2010.

[2] P. Costa, A. Donnelly, A. Rowstron, and G. O'Shea. "Camdoop: Exploiting in-network aggregation for big data applications". In Proceedings of the USENIX conference on Networked Systems Design and Implementation, 2012.

[3] J. Dean and S. Ghemawat. "MapReduce: Simplified data processing on large clusters". In Proceedings of the Usenix Symposium on Operating Systems Design and Implemenetation, 2004.

[4] Jiaxing Zhang, Hucheng Zhou, Rishan Chen, Xuepeng Fan, Zhenyu Guo, Haoxiang Lin, Jack Li, Wei Lin, Jingren Zhou, and Lidong Zhou. "Optimizing Data Shuffling in Data-Parallel Computation by Understanding User-Defined Functions". In Proceedings of the USENIX Symposium on Networked Systems Design and Implementation, April 2012.

[5] Carlos Teixeira, Alexandre J. Fonseca, Marco Serafini, Georgos Siganos, Mohammed J. Zaki, and Ashraf Aboulnaga. "Arabesque: a system for distributed graph mining." In Proceedings of the ACM Symposium on Operating Systems Principles, 2015.

B. Consensus (Note: This topic is somewhat more algorithmic than other topics.)

[1] W. Bolosky, D. Bradshaw, R. Haagens, N. Kusters, and P. Li. "Paxos replicated state machines as the basis of a high-performance data store". In Proceedings of the USENIX conference on Networked Systems Design and Implementation, 2011.

[2] Jialin Li, Ellis Michael, Naveen Kr. Sharma, Adriana Szekeres, and Dan R. K. Ports. "Just Say NO to Paxos Overhead: Replacing Consensus with Network Ordering." In Proceedings of the USENIX conference on Networked Systems Design and Implementation, 2016.

[3] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. "Algorand: Scaling Byzantine Agreements for Cryptocurrencies." In Proceedings of the ACM Symposium on Operating Systems Principles, October 2017.

C. Application-level Multicast

[1] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. "Split-Stream: High-bandwidth multicast in cooperative environments". In Proceedings of the ACM Symposium on Operating Systems Principles, Oct. 2003.

[2] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat. "Bullet: High bandwidth data dissemination using an overlay mesh". In Proceedings of the ACM symposium on operating systems principles, 2003.

[3] H. Li, A. Clement, E. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. "BAR Gossip". In Proceedings of the Usenix Symposium on Operating Systems Design and Implemenetation, 2006.

[4] M. Piatek, A. Krishnamurthy, A. Venkataramani, R. Yang, D. Zhang, and A. Jaffe. "Contracts: Practical contribution incentives for p2p live streaming". In Proceedings of the USENIX conference on Networked Systems Design and Implementation, 2010.

[5] Idit Keidar, Roie Melamed, and Ariel Orda. "EquiCast: scalable multicast with selfish users". In Proceedings of the ACM symposium on principles of distributed computing, 2006.

D. Server Architecture

[1] M. Krohn, E. Kohler, and M. Kaashoek. "Events can make sense". In Proceedings of the USENIX Annual Technical Conference, 2007.

[2] Z. Li, M. Zhang, Z. Zhu, Y. Chen, A. Greenberg, and Y. Wang. "WebProphet: Automating performance prediction for web services". In Proceedings of the USENIX conference on Networked Systems Design and Implementation, 2010.

[3] R. von Behren, J. Condit, and E. Brewer. "Why events are a bad idea (for high-concurrency servers)". In Proceedings of the Workshop on Hot Topics in Operating Systems, 2003.

[4] R. von Behren, J. Condit, F. Zhou, G. Necula, and E. Brewer. "Capriccio: Scalable threads for internet services". In Proceedings of the ACM symposium on operating systems principles, Oct. 2003.

[5] M. Welsh, D. Culler, and E. Brewer. "SEDA: An architecture for well-conditioned, scalable internet services". In Proceedings of the ACM symposium on operating systems principles, Oct. 2001.

E. Chord and DHT

[1] R. Geambasu, A. Levy, T. Kohno, A. Krishnamurthy, and H. Levy. "Comet: An active distributed key-value store". In Proceedings of the Usenix Symposium on Operating Systems Design and Implemenetation, October 2010.

[2] L. Glendenning, I. Beschastnikh, A. Krishnamurthy, and T. Anderson. "Scalable consistency in scatter". In Proceedings of the ACM symposium on operating systems principles, 2011.

[3] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. "OpenDHT: A public DHT service and its uses". In Proceedings of ACM SIGCOMM Conference, August 2005.

[4] A. Rowstron and P. Druschel. "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems". In Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms, November 2001.

[5] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. "Chord: A Scalable Peer-to- peer Lookup Service for Internet Applications", In Proceedings of ACM SIGCOMM Conference, August 2001.