CS556: Distributed Systems
Spring 2012 – Panagiota Fatourou
Student Presentations Schedule

Monday, April 23:

Manolis Surligas: (9) Baruch Awerbuch, “Complexity of network synchronization”, Journal of the ACM, 32(4):804-823, October 1985.

Herman De Beukelaer: (20) Paola Flocchini, Matthew Kellett, Peter C. Mason, Nicola Santoro, “Map construction and exploration by mobile agents scattered in a dangerous network”, IPDPS 2009: 1-10.

Wednesday, April 25:

Eugeniu Zaicanu (Evghenios): (17) L. Lamport, “The part-time parliament problem”, ACM Transactions on Computer Systems, Vol. 16, No. 2, pp. 133-169, 1998.

evghenia: (4) Kung and Robinson, “On optimistic Methods for Concurrency Control”, Transactions on Database Systems, Vol. 6, No. 2, pp. 213-226, 1981.

Monday, April 30:

Mykhailo Iaremko: K. Manassiev, M. Mihailescu, and C. Amza, “Exploiting distributed version concurrency in a transactional memory cluster”, in Proc. 11th ACM SIGPLAN symposium on Principles and practice of parallel programming (PPoPP’06), New York, NY, 2006, pp. 198–208.

Md Forhad Rabbi: C. Kotselidis, M. Ansari, K. Jarvis, M. Luj´an, C. C. Kirkham, and I. Watson, “DiSTM: A software transactional memory framework for clusters”, in Proc. 37th International Conference on Parallel Processing. IEEE Computer Society, 2008, pp. 51–58.

Wednesday, May 2:

Giorgos Papadakis: Annette Bieniusa and Thomas Fuhrmann, “Consistency in hindsight: A fully decentralized STM algorithm”, in 24th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2010, Atlanta, Georgia, USA, 19-23 April 2010 - Conference Proceedings, pages 1-12. IEEE, 2010.

Remarks:

1. Mykhailo, Forhad, and Giorgos should also read:

   Paolo Romano , Nuno Carvalho , Luís Rodrigues, "Towards distributed software transactional memory systems", Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware, September 15-17, 2008, Yorktown Heights, New York.

   They should not present this paper, but they have to compare against it during their presentation.

2. Paper Understanding:

   Each student should deeply understand the material presented in the paper s/he has undertaken. Most specifically, each student should:

   - know the algorithms and the techniques presented in the paper;
   - be able to answer to questions of the style «Why is each line of the code useful in the algorithms s/he will present and what could go wrong if any line was removed»;
   - invest time on the algorithm s/he studies, devise his/her own bad scenarios of execution and understand how the algorithms cope with these scenarios;
   - study/devise a big number of examples to deeply understand how the algorithms work; each student should include such examples in his/her presentation (and it is these examples that give a concrete idea of how deeply the student has understood the technical part of the paper);
   - invest some time to understand the high level idea of the analysis of the algorithms included in the paper;

   Each student should be able to present an intuitive description of the algorithms, their correctness and their complexity.

3. Presentation:

   Each presentation should last for 35 minutes (another 25 minutes will be allocated to questions and discussion). Each student should ensure that his/her presentation will not require more than 25 minutes (marking will be performed based on the material that each student will manage to cover in the first 35 minutes of his/her). Roughly speaking, each presentation should have the following form:

   - Description of the problem.
   - Brief description of the results.
   - Description of new algorithms (this description should be first highly intuitive and only when the speaker is sure that the audience has understood the main ideas of the algorithms, s/he can provide more technical details).
   - A lot of examples to prove that the student has achieved a good level of understanding of the algorithms.
   - Bad/difficult scenarios with which the algorithms should cope.
   - Why are all the components of an algorithm necessary? (e.g., why are all the line codes needed?, etc.)? The students may choose to answer these questions either by using examples or by providing sufficient explanation.
   - Intuitive (high-level) description of the analysis of the algorithm.
   - Conclusions– Open Problems

In a presentation, brief phrases (and not big sentences) are used. Moreover, a lot of figures and examples must be provided. In each slide, the material should be explained in a detailed way.

Each student must initiate a discussion on the topic that s/he presents. The marking will take into consideration whether this goal is accomplished. The marking will also be based on whether (and in what degree) the audience has understood the presented material.

The instructor and the other students are allowed to make questions. The time schedule should be respected despite this (so, each student should take into consideration any discussion and question that may arise during the presentation when it makes the time schedule of it).