

Preface

This volume contains the papers presented at the 18th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP 2014), which was held in Phoenix, USA, on May 23, 2014 in conjunction with the IEEE International Parallel Processing Symposium 2014 (IPDPS 2014).

The proceedings of the previous workshops are also available from Springer-Verlag as LNCS volumes 949, 1162, 1291, 1459, 1659, 1911, 2221, 2537, 2862, 3277, 3834, 4376, 4942, 5798, 6253, 7698, and 8429. These volumes are available as printed books and online.

This year, we had 24 papers submitted to the workshop, of which we accepted nine. All submitted papers went through a complete review process, with the full version being read and evaluated by an average of four reviewers. We would like to especially thank the Program Committee members and the additional referees for their willingness to participate in this effort and their detailed, constructive reviews.

As a primary venue of the parallel scheduling community, the Job Scheduling Strategies for Parallel Processors workshop offers a good vantage point to witness its evolution. During these two decades, we have seen parallel scheduling grow in scope and importance, following the popularization of parallel systems. Fundamental issues in the area remain relevant today (e.g., scheduling goal and evaluation, workload modeling, and performance prediction). Meanwhile, a new set of issues have emerged, due to new workloads, increased scale, and the differing priorities of cloud systems. Together, the traditional and new issues make for a lively and discussion-rich workshop, where academic researchers and participants from industry meet and exchange ideas and experiences.

The workshop began with a keynote talk by Liana Fong, from IBM. She discussed how cognitive computing places new challenges for parallel job scheduling. These computations must not merely achieve performance levels that were unthinkable a decade ago, but do so in a very flexible manner, interactively steering the computation to support human cognitive processes.

At the paper presentations, the blend of old and new challenges in parallel job scheduling set the tone of the discussions. Even the most classical scenario of single node parallelism remained a very active area. This trend is no surprise, both as the number of cores sharing the same memory increases and special purpose parallel processors like GPUs have become prevalent.

We had three papers tackling single-core parallelism. Seo et al. introduced Bubble Task, which scheduled memory access among tasks running in a multi-core machine by throttling the tasks. Looking beyond memory contention, Herz and Pinkau addressed the problem of scheduling of task graphs on shared memory machine, with special focus on task graphs generated automatically. Finally, for the second time in the workshop history, we visited the intersection of parallel job scheduling and real-time systems. Qamhie and Midonnet investigated the effects of parallelism in real-time

systems, showing that some reservations of the real-time community regarding parallelism were not fully justified.

Moving to distributed-memory, larger scale systems, scheduling fairness was a particularly hot topic this year. Klusáček and Rudová presented a new approach that supported multiresource aware user prioritization mechanism to ensure fairness. Importantly, this approach is capable of dealing with the heterogeneity of both jobs and resources. Rodrigo et al. investigated which basic prioritization primitives would make it easier to achieve fair scheduling in large, decentralized distributed. Tóth and Klusáček had a new take on the basic question of how to evaluate different scheduling algorithms. They proposed a user-centric approach that tries to measure the deviation of job end time from what the user would expect. The expectation of the user is based on fairness, which in itself generated a very interesting discussion on whether this model favors schedulers that strive for fairness.

The last three works presented this year touched different areas of parallel job scheduling. Kumar and Vadhiyar revisited the traditional question for batch systems of wait time prediction. They pushed the state of the art in prediction accuracy by using more sophisticated statistical models. They also evaluated how much meta-schedulers can improve performance by using such predictions.

Kuzmanovska et al. addressed how to schedule parallel jobs when these jobs are written using frameworks to deal with parallelism (e.g., MapReduce, Dryad, Pregel). They proposed a two-level approach in which each framework asks for the resources needed, and then distributes them to the jobs written in the framework.

Schwiegelshohn explored his long-time experience to distill lessons for scheduler algorithm designers. Starting with the provocative observation that most research papers in the area have had negligible impact on the state-of-practice, he identified constraints, objectives, and evaluation as the key reasons for this mismatch, and prescribed solutions based on his observations. The paper concluded by applying its own advice on designing a scheduling solution for an Infrastructure as a Service provider.

Enjoy the reading!

We hope you can join us in the next JSSPP workshop, in Hyderabad, India, in May 2015.

November 2014

Walfredo Cirne
Narayan Desai

Job Scheduling Strategies for Parallel Processing
18th International Workshop, JSSPP 2014, Phoenix, AZ,
USA, May 23, 2014. Revised Selected Papers
Cirne, W.; Desai, N. (Eds.)
2015, X, 169 p. 60 illus., Softcover
ISBN: 978-3-319-15788-7