Lecture Notes in Computer Science 8429

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

For further volumes:
http://www.springer.com/series/7407

Narayan Desai · Walfredo Cirne (Eds.)

Job Scheduling Strategies for Parallel Processing

17th International Workshop, JSSPP 2013
Boston, MA, USA, May 24, 2013
Revised Selected Papers

Springer

Editors
Narayan Desai
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL
USA

Walfredo Cirne
Google
Mountain View, CA
USA

ISSN 0302-9743 ISSN 1611-3349 (electronic)
ISBN 978-3-662-43778-0 ISBN 978-3-662-43779-7 (eBook)
DOI 10.1007/978-3-662-43779-7
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014941592

LNCS Sublibrary: SL1 – Theoretical Computer Science and General Issues

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers presented at the 17th workshop on Job Scheduling Strategies for Parallel Processing that was held in Boston, USA, on May 24, 2013, in conjunction with the IEEE International Parallel Processing Symposium 2013.

This year 20 papers were submitted to the workshop. 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 additional reviewers for their willingness to participate in this effort and for their detailed, constructive reviews.

As a prime venue of the parallel scheduling community, the Job Scheduling Strategies for Parallel Processors workshop offers a vantage point for one to witness the evolution in the area. When it began in 1995, parallel job scheduling was in its infancy. The first large-scale parallel machines had emerged over the preceding few years, demonstrating the practical need for parallel schedulers. Early parallel systems, and even modern supercomputers, have a very static set of resources and configuration. More recently, cloud systems have emerged in much larger scale configurations. Such systems differ from traditional supercomputers due to the frequent failure of its constituent components and a highly dynamic workload. Each kind of system has unique challenges associated with scheduling, but it seems that the workloads may be converging somewhat; supercomputers are increasingly running so-called many task workloads, whereas cloud workloads are including more workloads with task coupling. While the workload targets of these classes of systems will likely remain distinct, it is quite likely that similar scheduling techniques will be needed in both cases over the next few years, marrying dynamism with parallelism.

Another change caused by the growing importance of cloud systems is the integration of scheduling into a larger landscape of business concerns. With the extreme level of investment in cloud infrastructure, management processes like capacity planning have become coupled with scheduling into a far richer resource management landscape than we have previously seen. Open challenges remain in many areas and are often increased due to the implications of on-demand workloads. There is also an increased need for richer interactions between workloads and resource management infrastructure, which could enable dynamically moldable jobs, or other novel models for variable resource occupancy.

At the same time, large-scale systems have become far more accessible than ever before, broadening the use of large-scale computational campaigns. As more resource are used in this way, the incentive to optimize this process has grown substantially. Complex techniques are now used to optimize for cost and time to solution. This adds an economic dimension that previously did not exist in large-scale systems.

All of these areas are complex and remain unsolved. JSSPP has evolved with the area and now fully covers parallel scheduling for commercial environments while still

maintaining strong interest in its traditional areas: scientific computing, supercomputing, and cluster platforms.

The workshop began with a keynote talk by Stephen Elliot, from Amazon Web Services. Stephen gave an overview of the Amazon spot market for computing resources, and described some of the open issues in scheduling the spot market. The AWS spot market is particularly interesting in a few regards. It is built on functional economic principles, using dynamic pricing to minimize resource waste and maximize utilization. Another interesting characteristic of the spot market is the direct integration of capacity planning into the resource management process. This model is a departure from the traditional HPC systems where systems are built and then operated largely in steady state for several years. By contrast, cloud systems function in a continuous acquisition model, where hardware resources are added on a regular basis. This model has very different properties, and enables different strategies for resource management. We expect research in this area to grow in the coming years.

In addition to this topic, scheduling issues were discussed in a broader context in more established areas, from hardware scheduling, to scheduling within budget constraints, scheduling for performance, and analysis of scheduling tasks within resource management software. Adopting a broader basis for scheduling discussions was an explicit goal of the workshop this year, and will be continued in future workshops.

JSSPP has a long-standing tradition of covering workload modeling and metrics analysis. After all, optimizing for the wrong metric cannot produce the right result. The work of Emeras et al. examines how to enhance our understanding of the parallel system by combing the view of the scheduler (on which resources allocated are considered used) with instrumentation of what effectively happens at the machine level (on which not all allocated resources are indeed utilized). Krakov et al. explore heatmaps as an alternative to the long-standing problem that summarizing a parallel system's behavior with one-number statistics invariably leads to losing important information, and sometimes is downright misleading.

Scheduling fairness remains a topic of interest for the community. Klusáček et al. investigate the very definition of fairness when multiple resources are taken into account, describe the problems resulting from it, and propose solutions. Rajbhandary et al. work in the nut-and-blots of the fairness problem, proposing a new scheduling algorithm that beats the state-of-art one. We expect activity in fairness to be extended in the next few years to accommodate for the cloud reality, where consumers who pay more get better quality of service. We are likely to see research on which fairness is weighted (by price or otherwise).

How to schedule big data jobs was also actively debated during the workshop, a recognition of how important it has become in recent years. Cao et al. introduce a handful of scheduling algorithm that simultaneous target throughput and budget optimization for DAG applications, which are common in big data pipelines. Agosta et al. explore how to perform task placement of MapReduce applications to improve data locality and thus performance.

As the scale of parallel systems keep increasing, a centralized scheduler overseeing the entire system starts to become a bottleneck. Balasubramanian et al. address this issue introducing decentralized scheduling strategies that backfill jobs locally and dynamically migrate waiting jobs to leverage residual resources. Likewise, the scale

of today's systems make energy consumption a major concern, both from an economic and an environmental viewpoint. Zhou et al. show how power-aware job scheduling can reduce the energy cost significantly by as much as 25% with minimal impact to system utilization.

As parallel systems grow in scale, they also grow in complexity. It is interesting to note that meta-heuristics are emerging as an effective way to deal with such complexity. Shai et al. introduce Max-Jobs, a meta-heuristic that combines simpler matching heuristics to improve the matching of jobs to machines. Deng et al. go a step further even and dynamically change the whole scheduling algorithm as to accommodate for changes in workload and conditions of the parallel system.

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

January 2014

Narayan Desai
Walfredo Cirne

Organization

Workshop Organizers

Narayan Desai	Argonne National Laboratory, USA
Walfredo Cirne	Google, USA

Program Committee

Henri Casanova	University of Hawaii at Manoa, USA
Julita Corbalan	Technical University of Catalunya, Spain
Dick Epema	Delft University of Technology, The Netherlands
Dror Feitelson	The Hebrew University, Israel
Ian Foster	Argonne National Laboratory, USA
Alfredo Goldman	University of Sao Paulo, Brazil
Allan Gottlieb	New York University, USA
Morris Jette	SchedMD, USA
Rajkumar Kettimuthu	Argonne National Laboratory, USA
Derrick Kondo	Inria, France
Zhiling Lan	Illinois Institute of Technology, USA
Virginia Lo	University of Oregon, USA
Satoshi Matsuoka	Tokyo Institute of Technology, Japan
Jose Moreira	IBM T.J. Watson Research Center, USA
Bill Nitzberg	Altair Engineering, USA
Mark Squillante	IBM T.J. Watson Research Center, USA
Dan Tsafrir	Technion, Israel
John Wilkes	Google, USA
Ramin Yahyapour	The University of Göttingen, Germany

Reviewers

Henri Casanova
Julita Corbalan
Dick Epema
Gilles Fedak
Dror Feitelson
Liana Fong
Eitan Frachtenberg
Alfredo Goldman
Allan Gottlieb
Alexandru Iosup
Morris Jette
Rajkumar Kettimuthu
Dalibor Klusáček
Zhiling Lan
Bill Nitzberg
David Oppenheimer
Uwe Schwiegelshohn
Mark Squillante
Wei Tang
Dan Tsafrir
Ramin Yahyapour

Contents