HUMBOLDT-UNIVERSITÄT ZU BERLIN

Master's Thesis Exposé

Comparison of lightweight and compute-intensive heuristics for DAG scheduling with network contention

Author: Carsten Lipka

Tutors: Prof. Dr. Ulf Leser
Carl Witt

The aim of this Master's thesis is the investigation of schedule qualities by extending the HEFT algorithm and a Genetic Algorithm to contention-aware scheduling heuristics. Both algorithms are modified in terms of transfer times for simultaneously data connections inside the cluster to explore potential advantages of the GA on the one hand, and the increasing complexity of the problem on the other hand compared to the originally proposed versions of the algorithms.

Index-terms — static scheduling, genetic algorithms, HEFT, GPU, CUDA.

1 Introduction

In contradiction to the initially examined scheduling problem from the 1950s, data transfer between tasks is not negligible for scientific workflow scheduling, due to the vast amount of data consumed by nowadays scientific workflows. The optimization problem of workflow scheduling is to distribute a set of tasks onto existing resources such that the total execution time of the workflow is minimal. Input data of a static scheduler contain the following: Jobs and data dependencies between jobs in the form of a Directed-Acyclic-Graph (DAG), the number and type of machines/resources, the transfer time of files from one machine to another, the size of the input and output data of jobs and the execution time of jobs on machines. After Cook has shown that the job scheduling problem in its general form is one of the NP-hard problems [GJ79], a set of research on the subject has been published [Bru07]. As part of these works appropriate heuristics were developed to efficiently solve the problem. Scheduling heuristics make use of characteristics of the scheduling problem or its domain in order to reduce the problem's search space and thus speed up the execution significantly. As a state-of-the-art scheduling heuristic for scientific workflows the Heterogeneous Earliest Finish Time (HEFT) algorithm makes use of the dependencies of tasks by pruning schedules that are

not executable [THW02]. Furthermore, HEFT uses a greedy approach for ranking following tasks, which prevents the heuristic from contemplate different rankings for a particular task. Therefore, the reduced search space comes at a cost of optimality in terms of the minimal makespan. HEFT is one of the most popular network-aware DAG scheduling heuristics, which assumed a complete graph network topology for simplification; however, the de facto standard for scientific computation environments is servers in clusters connected through switches. Although there are high-performance switches, that achieve nearly full bandwidth of all connected servers simultaneously, most hardware guarantees full simultaneous bandwidth only for a limited number of connections, which leads to a bottleneck. Thus, if the bandwidth of a network route is saturated by earlier scheduled connections and another connection is to be scheduled, the actual bandwidth of the new connection drops below the factor described in the static input data of the scheduling algorithm, resulting in schedules unrelated to the estimated network topology.

Unlike the HEFT algorithm, Genetic Algorithms (GA) do not limit the search space by exploiting certain properties of the problem, but follow an approach inspired by the theory of evolution. GAs are generally composed of the following phases: initialization, evaluation, selection, mutation and recombination (crossover). During initialization a set of random individuals (solutions) is generated, which is evaluated afterwards using a fitness function. The best individuals are then determined with respect to the fitness function in the selection phase, which are modified through the following mutation and the recombination phase. Mutations change random points of a solution; recombinations merge random areas of different solutions to a new solutions. Finally resulting individuals are re-evaluated with respect to the fitness function. The steps of the selection, mutation, recombination and evaluation are continuously executed until an termination condition is reached. In the case of the workflow scheduling problem individuals are represented by schedules and the fitness function calculates the total execution time including data transfer time for a given schedule. The search space of the scheduling problem solved by GAs is limited only by the termination condition and the associated number of iterations, respectively.

2 Aim

The aim of this Master's thesis is the investigation of schedule qualities by extending the HEFT algorithm and a Genetic Algorithm to contention-aware scheduling heuristics. Both algorithms are modified in terms of transfer times for simultaneously data connections inside the cluster to explore potential advantages of the GA on the one hand, and the increasing complexity of the problem on the other hand compared to the originally proposed versions of the algorithms. Different modification strategies, which are described in further detail in section 4, are to be examined and evaluated domain-independently to extend the closeness to reality of existing state-of-the-art Scientific Workflow scheduling approaches. Furthermore, the work will cover the appropriate parameter adjustment of the GA including the mutation rate and the total iteration count. Additionally, the thesis will approach the differences of schedules produced by both the HEFT algorithm and the GA in terms of closeness to nowadays modern commodity cluster's switch network topolgies to give insights of potential benefits resulting from additional computations and the extension of the search space compared to lightweight heuristics. In addition to the comparison of the modified algorithms, the algorithms are compared to the unmodified version of the HEFT algorithm. Another field of interest is the suboptimality of HEFT for increasing DAGs. Finally, the optional aim of the thesis is to investigate the correlation between GA's iteration count and the resulting schedule quality in terms of the minimal makespan objective. A GPU implementation can be used to experiment with large numbers of iterations.

3 Related Work

Scheduling with data transfers

In [THW02], Topcuoglu et al. introduced the HEFT algorithm as a good quality, low cost scheduler for heterogeneous processors. The HEFT algorithm selects the task with the highest upward rank value at each step and assigns the selected task to the processor, which minimizes its earliest finish time with an insertion-based approach. In [BSM10], Bittencourt et al. introduced an improvement of the HEFT algorithm with a lookahead strategy, where the locally optimal decisions made by the heuristic do not rely on estimates of a single task only. The work showed reductions of the makespan in most cases at the cost of higher computational times. In [BHR09], Benoit et al. introduced an efficient fault-tolerant scheduling algorithm that is both contention-aware and capable of supporting an arbitrary number of fail-silent (fail-stop) processor failures. In [AT06], Alkaya et al. introduced the Contention-Aware Scheduling (CAS) algorithm, with the objective of delivering good quality of schedules by considering contention on links of heterogeneous, arbitrarily-connected processors. The CAS algorithm schedules tasks on processors and messages on links by considering the earliest finish time attribute with the virtual cut-through (VCT) or the store-and-forward (SAF) switching. In [PH08], Park et al. presented a framework for data throttling that can be exploited by workflow systems to regulate the data transfers rate of interchanging tasks via a specially-created QoS-enabled GridFTP server. The authors framework included a workflow planner that constructs schedules that both specifies when/where individual tasks are to be executed, as well as when and at what rate data is to be transferred. The produced schedules were evaluated through an implementation of the Montage workflow, which led to an average

speed-up of 16 %. In [BMJD16], Bryk et al. focused on data-intensive workflows and addresses the problem of scheduling workflow ensembles under cost and deadline constraints in Infrastructure as a Service (IaaS) clouds. The authors proposed a simulation model for handling file transfers between tasks, featuring the ability to dynamically calculate bandwidth and supporting a configurable number of replicas, thus allowing them to simulate various levels of congestion. Furthermore, Bryk et al. evaluated a novel scheduling algorithm that minimizes the number of transfers by taking advantage of file locality and data caching. In [LW13], Lin et al. introduced analytical models to quantify the network performance of scientific workflows using cloud-based computing resources, and formulate a task scheduling problem to minimize the workflow end-to-end delay under a user-specified financial constraint. The authors proposed a heuristic solution to this problem, and illustrate its performance superiority over existing methods through extensive simulations and real-life workflow experiments based on proof-of-concept implementation and deployment in a local cloud test bed. In [VK13], Verma and Kaushal proposed a priority based GA for scheduling workflow applications to cloud resources with the objective of the overall cost of the workflow within a user's specified budget. The work considered data transfer between resources as well as execution costs of the different tasks. In [KKA+14], Kune et al. presented a Genetic Algorithm based scheduler for Big Data clouds that focusses data dependencies, computational resources and effective utilization of bandwidth.

GPU scheduling

In [NC11], Nesmachnow et al. demonstrated the application of parallel computing techniques using GPUs for improving scheduling heuristics for heterogeneous computing systems. The experimental evaluation for the proposed methods showed that reductions on the computing times in order of magnitude can be achieved, allowing to handle large scheduling scenarios in reasonable execution times. In [HHL12], Huang et al. presents an improved algorithm to the flow shop scheduling problem with fuzzy processing times and fuzzy due dates. The authors proposed the combination of the longest common substring method with the random key method. In [BXZ14], Bosson et al. introduced an application of GPUs for speeding up a schedule optimization problem under uncertainty and suggested a fast decision support algorithm to solve an air traffic management problem. In [QHTPT14], Quang et al. investigated power-aware task scheduling (PATS) in HPC clouds. The authors introduced a genetic algorithm with the objective of minimizing the total energy consumption of the placements of tasks.

4 Approach

As a solution to the problem of network contention, we propose an extension of scheduling algorithms by virtually splitting network bandwidth of interacting nodes. Initially, we extend the HEFT algorithm such that the transfer time of an additional connection over a saturated network route is adapted either by shifting the start time of the transfer or by scaling the transfer times accordingly. Thereupon, we modify the GA's fitness function such that the network bandwidth of simultaneous connections is shared out fairly between all interchanging nodes of the network. Finally, these modifications get implemented on both algorithms. The mutation phase of the GA is realized through single random machine-task reassignments or by swapping tasks in respect to the dependency DAG within the intermediate schedules. The crossover phase consists of the following steps: randomly split parent schedule A; replace all machine assignments right to the split position in A with its related task's machine assignments in B and vice versa, resulting in two offspring schedules C and D respectively. Through the approach of Design of Experiments all required parameters of the GA are adjusted.

The modified scheduling algorithms and also the unmodified version of the HEFT algorithm are evaluated using Pegasus' Scientific Workflow Generator [BCD+08]. Pegasus' generator covers Scientific Workflows from various fields such as NASA image assembly, earthquake characterization, epigenomics, gravitational waveform analyzation and sRNA search for bacterial replicons in the NCBI database. In [DCJ+14], Da Silva et al. presented Pegasus' collection of tools and data that enabled research in new techniques, algorithms, and systems for scientific workflows. These resources consist of execution traces of real workflow applications including resource usage and failure profiles, a synthetic workflow generator that can produce realistic synthetic workflows based on execution traces and a simulator framework for the execution of synthetic workflows on realistic distributed infrastructures. The variety of scientific fields and thus of their executed workflows ensures a domain-independent evaluation and comparison of the HEFT algorithm and the GA. Due to the limited time of a Master's thesis, the execution of the different Scientific Workflows is simulated using WorkflowSim [CD12] suggested by Pegasus or using RealCloudSim, an implementation through the Network Simulator 2 (NS-2). The simulation combines the scheduler's input data with its generated schedule to simulate task execution and network load for given workflows. Additionally, we evaluate the differences of scheduled and simulated makespans of the modified HEFT algorithm and the GA respectively. If the time makes it possible, the investigation of potential advantages of an decisive increase in the GA's iteration quantity is realised. The increase is achievable through a CUDA implementation of the Genetic Algorithm.

References

[AT06] Ali Fuat Alkaya and Haluk Rahmi Topcuoglu. A task scheduling algorithm for arbitrarily-connected processors with awareness of link contention. *Cluster Computing*, 9(4):417–431, 2006.

[BCD+08] Shishir Bharathi, Ann Chervenak, Ewa Deelman, Gaurang Mehta, Mei Hui Su, and Karan Vahi. Characterization of scientific workflows. In *2008 3rd Workshop on Workflows in Support of Large-Scale Science, WORKS 2008*, 2008.

[BHR09] Anne Benoit, Mourad Hakem, and Yves Robert. Contention awareness and fault-tolerant scheduling for precedence constrained tasks in heterogeneous systems. *Parallel Computing*, 35(2):83–108, 2009.

[BMJD16] Piotr Bryk, Maciej Malawski, Gideon Juve, and Ewa Deelman. Storage-aware Algorithms for Scheduling of Workflow Ensembles in Clouds. *Journal of Grid Computing*, 14(2):359–378, 2016.

[Bru07] Peter Brucker. *Scheduling Algorithms*, volume 93. Springer, 2007.

[BSB+01] Tracy D Braun, Howard Jay Siegel, Noah Beck, Ladislau L Bölöni, Muthucumaru Maheswaran, Albert I Reuther, James P Robertson, Mitchell D Theys, Bin Yao, Debra Hensgen, Richard F Freund, and Ladislau L Boloni. A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. *Journal of Parallel and Distributed Computing*, 61(6):810–837, 2001.

[BSM10] Luiz F. Bittencourt, Rizos Sakellariou, and Edmundo R M Madeira. DAG scheduling using a lookahead variant of the heterogeneous earliest finish time algorithm. In *Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing, PDP 2010*, pages 27–34, 2010.

[BXZ14] Christabelle Bosson, Min Xue, and Shannon Zelinski. GPU-based Parallelization for Schedule Optimization with Uncertainty. In *Atlanta, GA, 14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014.

[CD12] Weiwei Chen and Ewa Deelman. WorkflowSim: A toolkit for simulating scientific workflows in distributed environments. In *2012 IEEE 8th International Conference on E-Science, e-Science 2012*, 2012.

[CJSZ08] L.-C. Canon, E Jeannot, R Sakellariou, and W Zheng. Comparative Evaluation Of The Robustness Of DAG Scheduling Heuristics. *Grid Computing*, pages 73–84, 2008.

[DCJ+14] Rafael Ferreira Da Silva, Weiwei Chen, Gideon Juve, Karan Vahi, and Ewa Deelman. Community resources for enabling research in distributed scientific workflows. In *Proceedings - 2014 IEEE 10th International Conference on eScience, eScience 2014*, volume 1, pages 177–184, 2014.

[GJ79] Michael R. Garey and David S. Johnson. A Guide to the Theory of NP-Completeness, 1979.

[HHL12] Chieh-Sen Huang, Yi-Chen Huang, and Peng-Jen Lai. Modified genetic algorithms for solving fuzzy flow shop scheduling problems and their implementation with CUDA. *Expert Systems with Applications*, 39(5):4999–5005, 2012.

[KKA+14] R Kune, P K Konugurthi, A Agarwal, R R Chillarige, and R Buyya. Genetic Algorithm Based Data-Aware Group Scheduling for Big Data Clouds. In *Big Data Computing (BDC), 2014 IEEE/ACM International Symposium on*, pages 96–104, 2014.

[LW13] Xiangyu Lin and Chase Qishi Wu. On scientific workflow scheduling in clouds under budget constraint. In *Proceedings of the International Conference on Parallel Processing*, pages 90–99, 2013.

[NC11] Sergio Nesmachnow and Mauro Canabé. GPU implementations of scheduling heuristics for heterogeneous computing environments. In *XVII Congreso Argentino de Ciencias de la Computación*, 2011.

[PH08] Sang Min Park and Marty Humphrey. Data throttling for data-intensive workflows. In *IPDPS Miami 2008 - Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium, Program and CD-ROM*, 2008.

[QHTPT14] Nguyen Quang-Hung, Le Thanh Tan, Chiem Thach Phat, and Nam Thoai. A GPU-Based Enhanced Genetic Algorithm for Power-Aware Task Scheduling Problem in HPC Cloud. In *Information and Communication Technology*, pages 159–169. Springer, 2014.

[THW02] Haluk Topcuoglu, Salim Hariri, and M Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, 2002.

[VK13] Amandeep Verma and Sakshi Kaushal. Budget constrained priority based genetic algorithm for workflow scheduling in cloud. In *Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2013)*, pages 216–222, 2013.

[WPF05] Marek Wieczorek, Radu Prodan, and Thomas Fahringer. Scheduling of scientific workflows in the ASKALON grid environment. *ACM SIGMOD Record*, 34(3):56, 2005.

[ZLG+15] Zhi-hui Zhan, Xiao-fang Liu, Yue-jiao Gong, Jun Zhang, Henry Shu-Hung Chung, and Yun Li. Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. *ACM Computing Surveys*, 47(4):1–33, 2015.