technische universität dortmund

computer science 12

Bachelor/Master Thesis

Implementation and Evaluation of Stationary GANG Scheduling

Niklas Ueter
Prof. Dr. Jian-Jia Chen
Otto-Hahn Str. 16
Technische Universität Dortmund
Email: niklas.ueter@tu-dortmund.de

Introduction

In hard real-time systems, it is mandatory to verify the temporal behavior of the application, e.g., the compliance to deadline constraints. In parallel task scheduling, inter- and intra-task parallelism has to be considered in the timing analysis, where inter-task parallelism refers to the co-scheduling of different tasks and intra-task parallelism refers to parallel execution of a single task. In the context of task models for parallel computing, fork/join models [1], synchronous parallel task models, and DAG (directed-acyclic graph) based task models [2–7] have been proposed and analyzed with respect to real-time constraints.
In the gang task model, a set of threads is grouped together into a so called *gang* with the additional constraint that all threads of a gang must be co-scheduled at the same time on available processors. It has been demonstrated that gang-based parallel computing can improve the performance in many cases [8, 9].
Due to its practicability, the gang model is supported by many parallel computing standards, e.g., MPI, OpenMP, Open ACC or GPU computing.
In the *stationary gang* scheduling paradigm, each gang task is statically assigned to a set of processors, in which the cardinality of the set is equal to the gang size of the task. After this assignment is done, a gang task is only eligible to be executed on stationary processors assigned to it.

Thesis

1. In this thesis, the student should implement fixed-priority stationary GANG scheduling in LITMUS-RT[1] using the C programming language.
2. Moreover the student should devise experiments to evaluate the scheduling overheads and run-time variations using the provided tracing tools in LITMUS-RT.
3. (optional) The student could asses if it is possible to integrate memory bandwidth controllers, e.g., memguard [2] into stationary GANG scheduling to limit memory contention of co-scheduled tasks.

Ideally, this thesis should either demonstrate the benefits of the stationary constraint with respect to memory contention, scheduling overheads and run-time variations or hint to the performance problems of this approach. If you are interested, please do not hesitate to contact me for further information and literature.

[1] https://www.litmus-rt.org/
[2] https://github.com/heechul/memguard

Required Skill

- Basic knowledge with Linux and systems programming
- Comfortable with C and Python
- Interested in Real-Time scheduling

Acquired Skills after the work

- Knowledge of parallel real-time scheduling algorithms
- Knowledge of real-time operating systems
- Design, Analysis, and Implementation of system software for real-time systems

References

[1] K. Lakshmanan, S. Kato, and R. R. Rajkumar, "Scheduling parallel real-time tasks on multi-core processors," in *Proceedings of the 2010 31st IEEE Real-Time Systems Symposium*, RTSS '10, pp. 259–268, 2010.

[2] J. Fonseca, G. Nelissen, and V. Nélis, "Improved Response Time Analysis of Sporadic DAG Tasks for Global FP Scheduling," in *Proceedings of the 25th International Conference on Real-Time Networks and Systems*, 2017.

[3] J. C. Fonseca, G. Nelissen, V. Nélis, and L. M. Pinho, "Response time analysis of sporadic DAG tasks under partitioned scheduling," in *11th IEEE Symposium on Industrial Embedded Systems, SIES*, pp. 290–299, 2016.

[4] S. Baruah, "The federated scheduling of constrained-deadline sporadic DAG task systems," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition, DATE*, pp. 1323–1328, 2015.

[5] S. Baruah, "Federated scheduling of sporadic DAG task systems," in *IEEE International Parallel and Distributed Processing Symposium, IPDPS*, pp. 179–186, 2015.

[6] V. Bonifaci, A. Marchetti-Spaccamela, S. Stiller, and A. Wiese, "Feasibility analysis in the sporadic dag task model," in *ECRTS*, pp. 225–233, 2013.

[7] A. Melani, M. Bertogna, V. Bonifaci, A. Marchetti-Spaccamela, and G. C. Buttazzo, "Response-Time Analysis of Conditional DAG Tasks in Multiprocessor Systems," in *Proceedings of the 2015 27th Euromicro Conference on Real-Time Systems*, 2015.

[8] M. A. Jette, "Performance characteristics of gang scheduling in multiprogrammed environments," in *Proceedings of the 1997 ACM/IEEE Conference on Supercomputing*, SC '97, 1997.

[9] D. G. Feitelson and L. Rudolph, "Gang scheduling performance benefits for fine-grain synchronization," *Journal of Parallel and Distributed Computing*, vol. 16, pp. 306–318, 1992.