

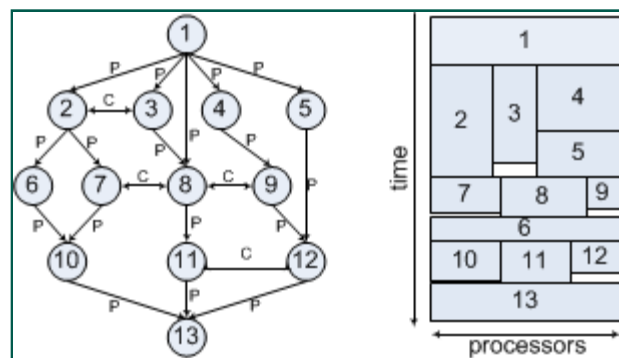


## Communicating Multiprocessor-Tasks

**Prof. Dr. G. Runger**  
Faculty of Computer Science  
Chemnitz University of Technology

### Overview

The scalability of parallel applications on large distributed memory platforms is often impacted by expensive global communication and synchronization operations. The communication overhead of many applications can be reduced by exploiting coarse grained parallelism. Many approaches based on parallel tasks have been proposed to tackle this problem, but usually the interaction between parallel tasks is restricted to input-output relations.



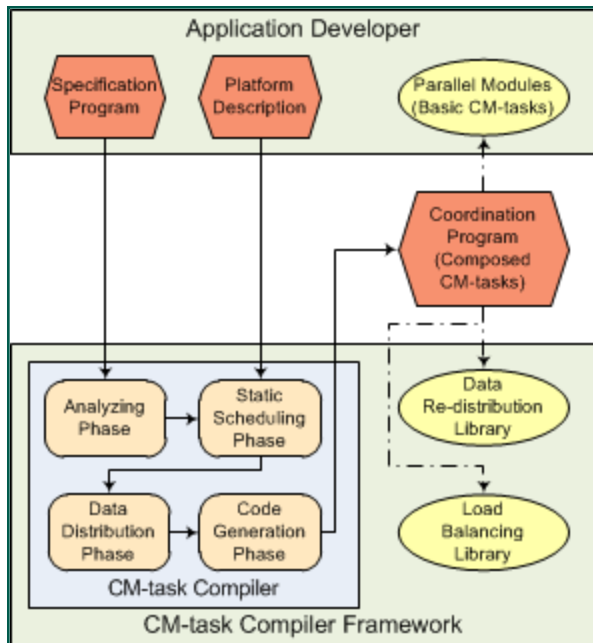
The programming model of *Communicating Multiprocessor-Tasks (CM-tasks)* extends previous models based on parallel tasks by additionally supporting an interaction between parallel tasks during their execution. This allows a more flexible structuring of parallel applications and enables the use of optimized communication patterns for data exchanges.

In particular, the CM-task programming model

- supports the exploitation of task parallelism (between CM-tasks) and data parallelism (within CM-tasks) at the same time;
- provides two possibilities for the interaction of the CM-tasks of program: input-output relations (P-relations) and communication relations (C-relations);
- allows a flexible mapping of the CM-tasks onto the execution resources of a parallel platform by leaving the flexibility to execute independent CM-tasks concurrently or one after another;
- includes software support in form of the CM-task compiler framework to

generate efficient implementations starting with a user-provided specification program.

## CM-task Compiler Framework



The CM-task compiler framework supports the application developer in creating efficient implementations of CM-task programs. The framework requires the user to provide a platform-independent specification program, a machine description and implementations of the parallel modules in form of parallel functions, e.g. as a *C+MPI* function. The core of the framework is the CM-task compiler that translates the user-provided specification program and machine description into an executable coordination program

using several transformation steps. Additionally, the framework includes runtime libraries to support the execution of the coordination program produced.

The platform-independent specification program includes

- the definition of the interfaces of the user-provided parallel functions;
- cost estimates for the parallel functions depending on the number of executing processors;
- description of the structure of the application, i.e., the interactions between the CM-tasks used.

The machine description file defines

- the number of processors of the parallel target platform;
- the computational performance of the processors;
- the communication performance of the interconnection network.

The coordination program produced includes

- an underlying schedule that is adapted to a specific parallel platform;

- management code for the creation of the processor groups and the execution of CM-tasks on these processor groups;
- data re-distribution operations to guarantee a correct data flow between CM-tasks;
- dynamic load balancing to adapt the sizes of the processor groups to the computational work performed (optional).

## Cooperation

This project is a collaboration between the Chemnitz University of Technology and the Bayreuth University.

## Publications on CM-tasks

- [DÜMMLER, J.](#): *Entwicklung effizienter gemischt paralleler Anwendungen*, TU Chemnitz, Fakultät für Informatik, Doctoral thesis, 2010. [online resource available](#)
- [DÜMMLER, J.](#); [RAUBER, T.](#); [RÜNGER, G.](#): *Mixed Programming Models using Parallel Tasks*. In: DONGARRA, J.; HSU, C.-H.; LI, K.-C.; YANG, L. T.; ZIMA, H. (Eds.): *Handbook of Research on Scalable Computing Technologies*: pp. 246-275. Information Science Reference - ISBN 978-1-60566-661-7, 2009. [online resource available](#)
- [DÜMMLER, J.](#); [RAUBER, T.](#); [RÜNGER, G.](#): *A Transformation Framework for Communicating Multiprocessor-Tasks*. In: *Proc. of the 16th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2008)*: pp. 64-71. IEEE Computer Society - ISBN 978-0-7695-3089-5. Toulouse, France, 2008. [online resource available](#)
- [DÜMMLER, J.](#); [RAUBER, T.](#); [RÜNGER, G.](#): *Communicating Multiprocessor-Tasks*. In: *Languages and Compilers for Parallel Computing: 20th International Workshop, LCPC 2007, Urbana, IL, USA, October 11-13, 2007, Revised Selected Papers* (LNCS, vol. 5234): pp. 292-307. Springer - ISBN 978-3-540-85260-5, 2008. [online resource available](#)



## [Complete list as BibTeX](#)

### **Related Publications on Parallel Tasks**

- [DÜMMLER, J.](#); [RAUBER, T.](#); [RÜNGER, G.](#): [Scalable Computing with Parallel Tasks](#). In: *Proc. of the 2nd IEEE Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS '09)*. ACM - ISBN 978-1-60558-714-1. Portland, Oregon, USA, 2009. DOI: 10.1145/1646468.1646477 [online resource available](#)
- [DÜMMLER, J.](#); [RAUBER, T.](#); [RÜNGER, G.](#): [Mapping Algorithms for Multiprocessor Tasks on Multi-Core Clusters](#). In: *Proc. of the 37th International Conference on Parallel Processing (ICPP 2008)*: pp. 141-148. IEEE Computer Society - ISBN 978-0-7695-3374-2. Portland, Oregon, USA, 2008. DOI: 10.1109/ICPP.2008.42 [online resource available](#)



## [Complete list as BibTeX](#)

### Copyright Notice

The documents distributed by this server have been provided by the contributing authors as a means to ensure timely dissemination of scholarly and technical work on a noncommercial basis. Copyright and all rights therein are maintained by the authors or by other copyright holders, notwithstanding that they have offered their works here electronically. It is understood that all persons copying this information will adhere to the terms and constraints invoked by each author's copyright. These works may not be reposted without the explicit permission of the copyright holder.

-