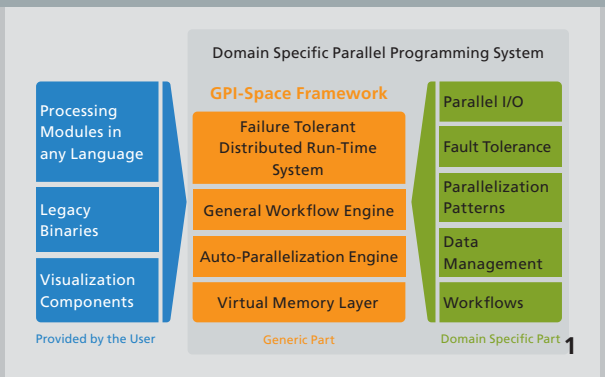
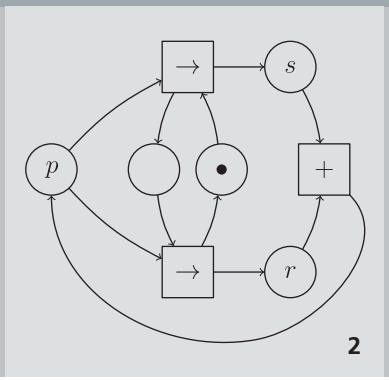
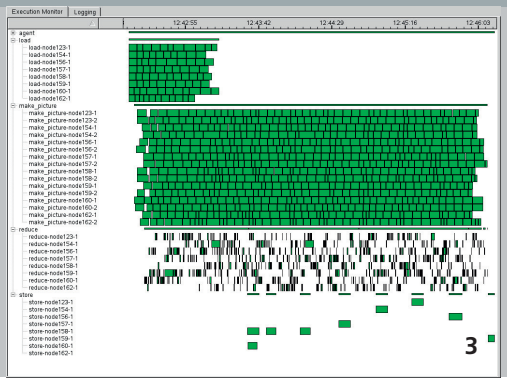
Fraunhofer
ITWM

FRAUNHOFER INSTITUTE FOR INDUSTRIAL MATHEMATICS ITWM







1 *GPI-Space architecture*

2 *Parallel reduction without initial value as Petri net*

3 *GPI-Space runtime system monitor*

MEMORY DRIVEN COMPUTING: GPI-SPACE

Separation of concerns

To separate computation and coordination is the core idea of not only a variety of today's task based systems but goes back in time at least into the 1980ies when David Gelernter designed Linda. A single model subsumes different levels of parallelism and is orthogonal to concrete hardware configurations and to specific programming languages used to implement compute kernels. That orthogonality allows for independent optimization and for generic approaches to large scale execution in a dynamic and heterogeneous environment.

Petri net: Managed dependencies

Coordination means to describe dependencies between tasks. Petri nets are a well-known and well-understood formalism to describe concurrent systems and are used in GPI-Space as workflow description language. Data decomposition is naturally supported by Petri nets too. The workflow engine automatically parallelizes by maintaining a so called "front of activity" which contains all tasks that are ready to be executed at a given time.

Large Scale execution

The distributed runtime system of GPI-Space is fed by the workflow engine with "activities". Those are tasks bundled with their input data description. Taking into account the currently available resources and their capabilities the GPI-Space backfilling bunch scheduler dynamically assigns work to resources such that time to solution is minimized. In case resources fail or become unavailable the runtime system reassigns work to other resources without requiring any further application support.

Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM

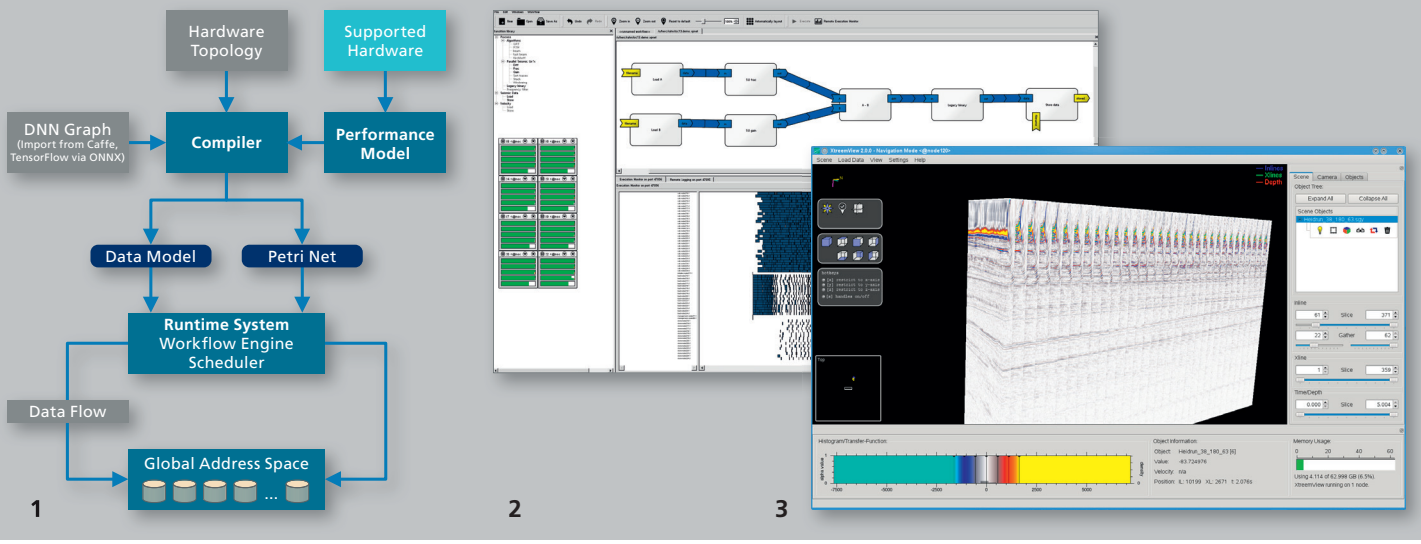
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany

Contact
Dr. Franz-Josef Pfreundt
Phone +49 631 31600-4459
pfreundt@itwm.fraunhofer.de

Dr. Mirko Rahn
Phone +49 631 31600-4553
mirko.rahn@itwm.fraunhofer.de

www.itwm.fraunhofer.de

hpc_flyer_GPI-Space_Memory driven computing_EN



Memory driven computing

Petri nets are powerful tools to describe data and task parallelism and to separate activation from execution. To make large amounts of data easily accessible for the activated executions GPI-Space has introduced an application independent memory layer (IML). This is a separate process with a dedicated API to manage memory and to provide information on data transport costs. Data transfer within a distributed system is managed with help of the IML. In the result all tasks are started with pointers to the input data in local memory but need not to worry about data transport at all. Instead data transfers are scheduled and executed asynchronously by the runtime system. At the same time the concrete physical data representation is hidden from the application. The IML can be placed in "memory segments", which are distributed in DRAM, HBM, NVRAM or even the BeeGFS parallel file system.

Coupled distinct applications

This memory driven computing approach using the IML has the advantage that applications can easily share data. Applications written in different languages or running on different architectures can be coupled via the IML. Moreover, individual tasks can be legacy code and are not required to be aware of the coupling. That means that tasks can correspond to threads in a parallel C++ program or be Matlab modules or even processes that belong to a large scale visualization program.

Domain specialization

GPI-Space is a general purpose system and is typically not used "bare bone" but as core of a further specialized domain specific framework. GPI-Space supports specialization on all levels. The Petri net workflow description is hierarchical and provides an embedded type system for user defined types and an embedded expression language that is integrated with the type system. To specialize GPI-Space for a domain means to describe the data structures and workflow patterns once and to reuse it either from a library or in a domain specific higher level language or compiler.

Example: Industrial Seismic Imaging

Seismic imaging is a demanding field and a genuine big data application. Data sizes in the order of dozens of Terabytes meet plenty of complex processing steps and optimization workflows. Strict quality measures meet short turnaround cycles and the constant request for real time analysis. Built on internal domain expertise and in close cooperation with industrial partners GPI-Space is used with great success as core technology in industrial Seismic Imaging providing scalable and robust infrastructure in highly dynamic environments and for very long running applications.

Example: Deep Learning

High performance computing technology such as fault tolerance or asynchronous data transfer has proven to be beneficial in deep learning scenarios too. The project HP-DLF uses GPI-Space as core technology. The goal of the project is to provide an abstract, scalable and robust compute infrastructure to domain scientist from the artificial intelligence community. Compiler experts will produce a python embedded interface on top of GPI-Space that will allow domain scientists to make efficient use of large heterogeneous machines. GPI-Space provides fault tolerance, scalability and distributed memory and its scheduler will be enriched with support for accurate performance models.

Example: Particle visualization

In cooperation with the Leibniz Computing Centre and the University observatory Munich GPI-Space is used as integration platform for the particle visualization tool SPLOTCH. Within 3 work weeks the end to end throughput of SPLOTCH was boosted by a factor of 10x. Crucial is the automatic advanced scheduling and communication hiding provided by GPI-Space.

1 *GPI-Space as core in the BMBF project HP-DLF*

2 *Integrated domain specific graphical development and processing*

3 *Visualization of partial results using XTreemView*