MASSIVELY PARALLEL WORKFLOWS WITH GPI-SPACE

Authors:

B. Lörwald, R. Machado, M. Rahn, T. Rotaru, M. Zeyen

Minisymposium 28:
Massively Parallel Methods in Geometry and Applications

Fraunhofer
ITWM

Outline

- Introduction
- Architecture
- Application Workflow
- Summary

Outline

- **Introduction**
  - What is GPI-Space?
  - Where is GPI-Space?

- Architecture
- Application Workflow
- Summary

Introduction - What is GPI-Space?

- GPI-Space is a software development and execution platform developed at Fraunhofer ITWM
  - Task-based workflow management system
  - Automatic workflow parallelization

- Built on the concept of separation between coordination and computation
  - Programming Model for parallel applications
  - Parallel execution framework optimized for HPC environments
  - Fault tolerant task execution

Introduction - Where is GPI-Space?

Introduction - Where is GPI-Space?

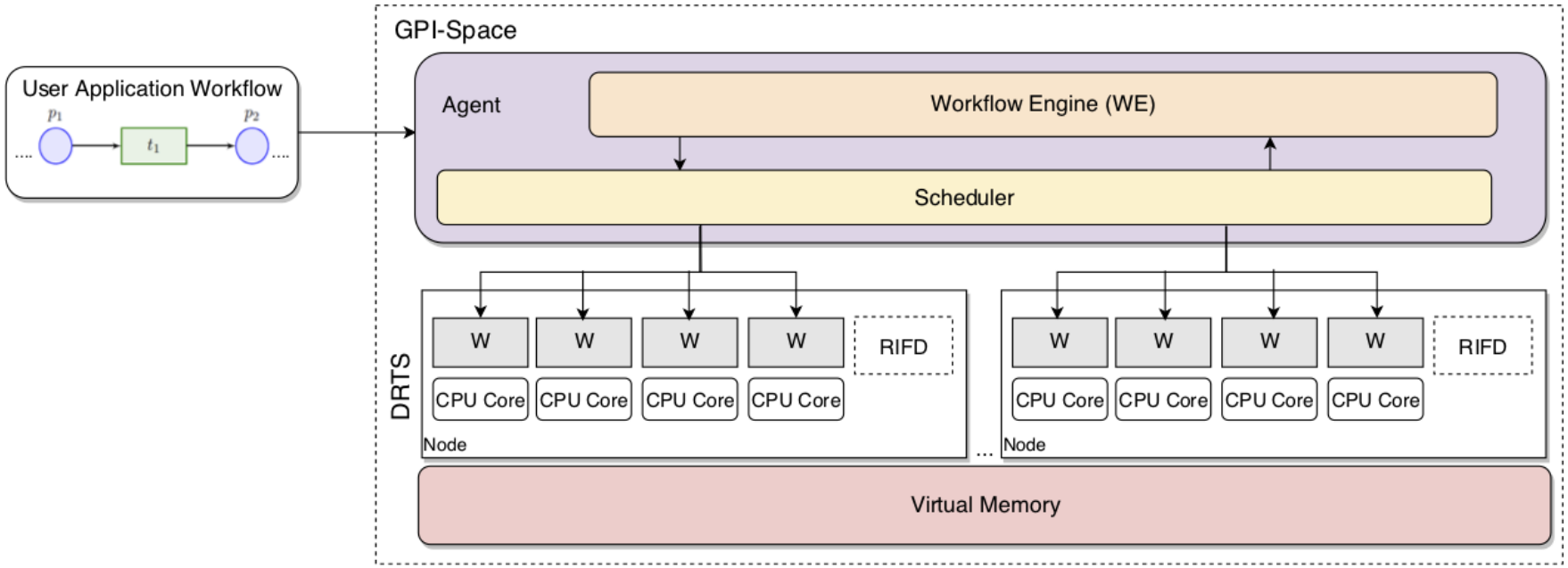
https://www.gpi-space.com

https://github.com/cc-hpc-itwm/gpispace

Outline

- Introduction
- **Architecture**
  - Agent
  - RIFD & Worker
  - Virtual Memory Layer
- Application Workflow
- Summary

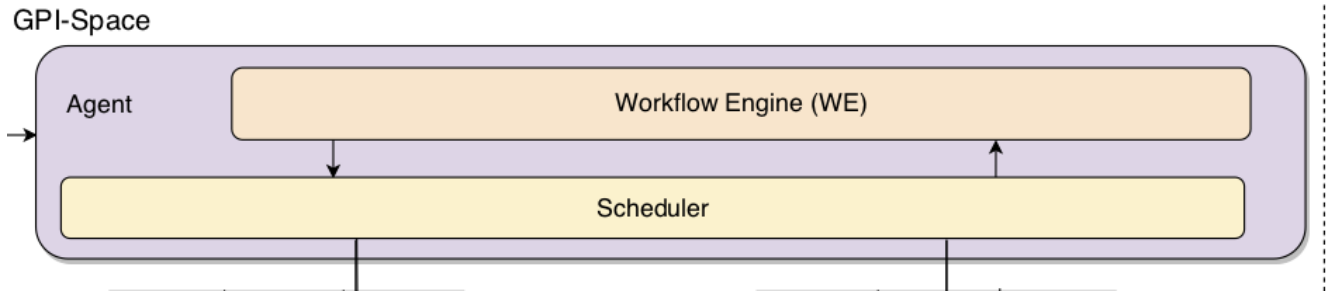
Architecture



Architecture - Agent

Agent (Workflow Management)

- Workflow Engine (WE)
  - Continously evaluates the state of a workflow by extracting available tasks for execution and handling token flow
- Scheduler
  - Drives resource allocation and assignment to tasks
  - Flexible scheduling policies adapted to the application



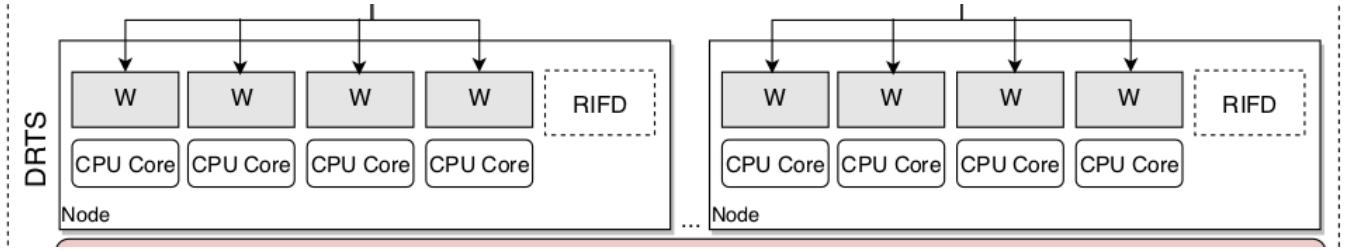
Architecture - RIFD & Worker

Remote Interface Daemon (RIFD)

- Runs on each compute node to manage the Worker processes

Worker

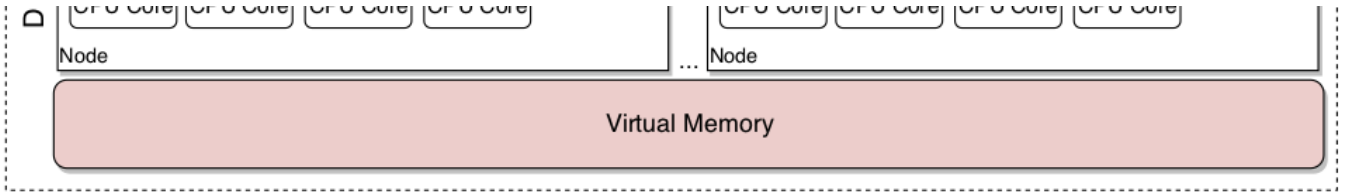
- Executes individual workflow tasks on compute nodes
- Handles automatic memory transfers



Architecture – Virtual Memory Layer

Virtual Memory Layer

- Distributed global memory
- Linear view of the aggregated memory across all compute nodes



Outline

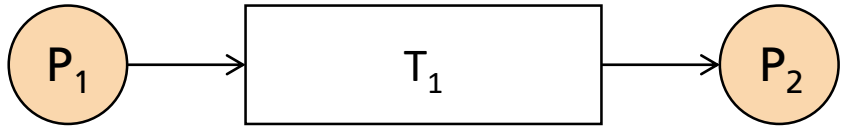
- Introduction
- Architecture
- **Application Workflow**
  - Petri Nets
  - Supported Extensions
  - Tasks
- Summary

Application Workflow – Petri Nets

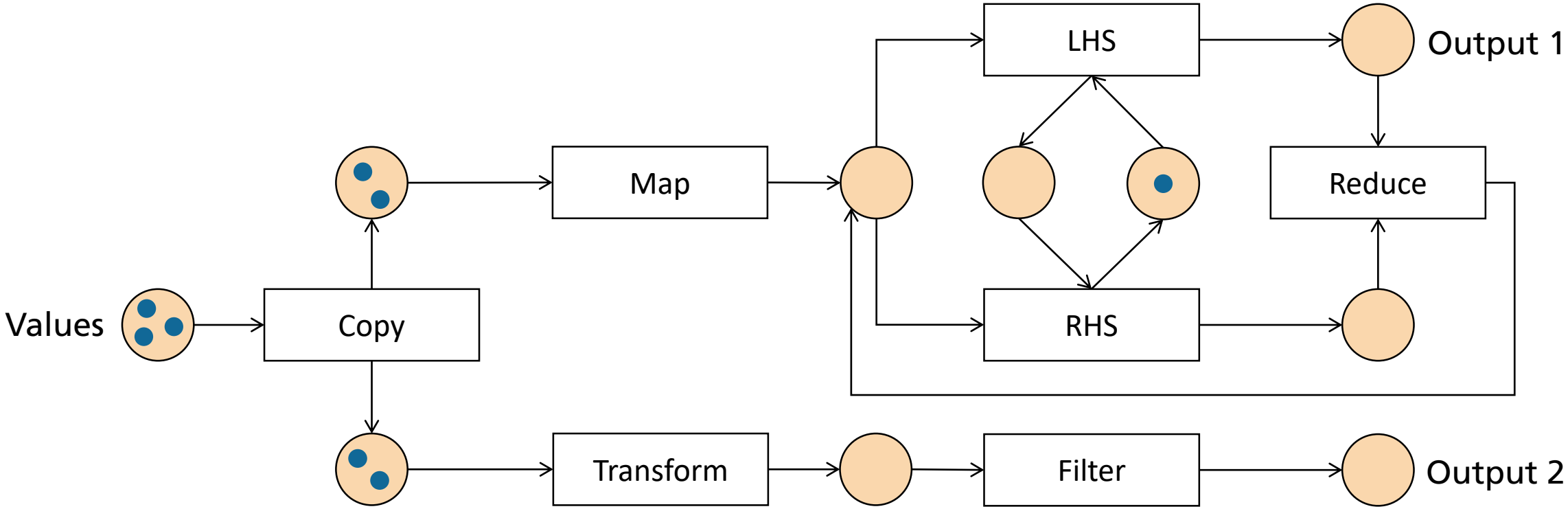
Definition:

A Petri net is a tuple $N = (P, T, F, M_0)$ where

- P is a finite set of places
- T is a finite set of transitions s.t. $P \cap T = \emptyset$
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (“flow relation”)
- $M_0 : P \to \mathbb{N}$ is a map representing the initial distribution of tokens, called the initial marking

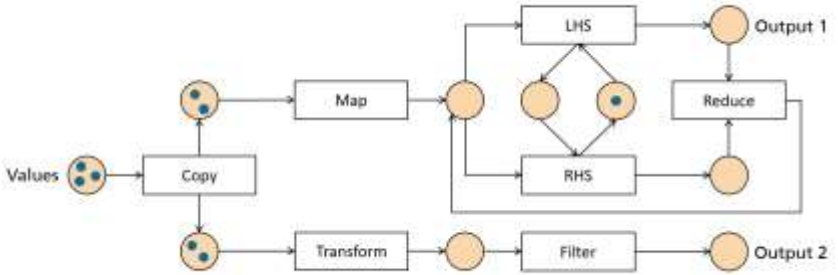


Application Workflow – Petri Nets



Application Workflow – Petri Nets

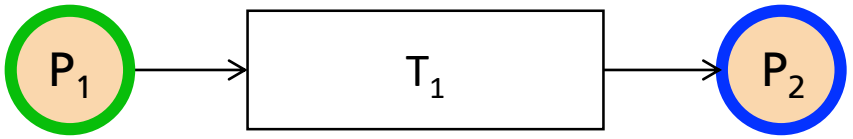
- Application workflow defined as Petri nets
  - Models parallel and distributed systems
  - Bipartite directed graph
- Task Parallelism vs. Data Parallelism
  - E.g., tasks Copy, Map, and Transform can be scheduled in parallel
  - E.g., 3 instances of Copy in parallel with 2 instances of Map and Transform



Application Workflow – Supported Extensions

Colored Petri nets:

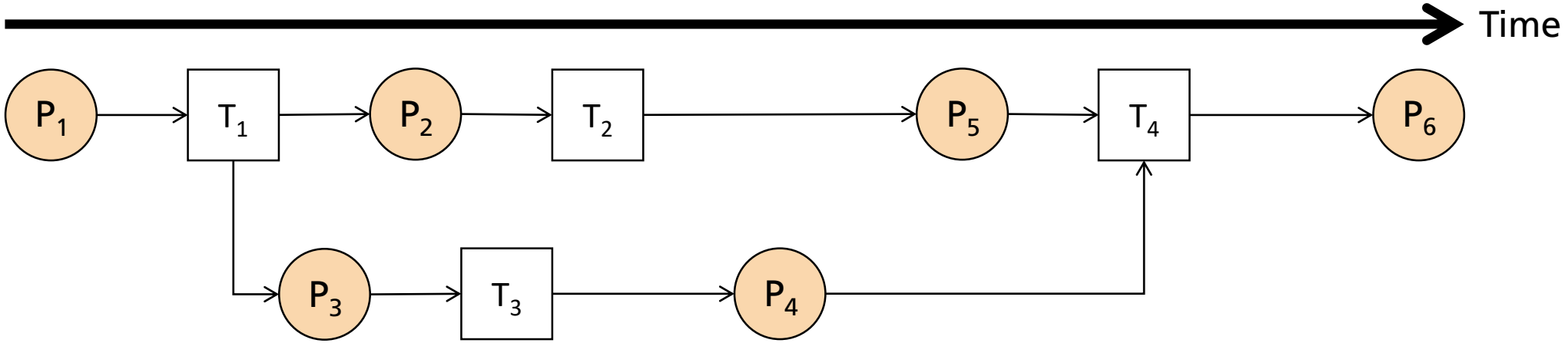
- Each place has a corresponding color set
- Each token on a place needs to have a color from the associated color set
- Colors are equivalent to data types



Application Workflow – Supported Extensions

Timed Transitions:

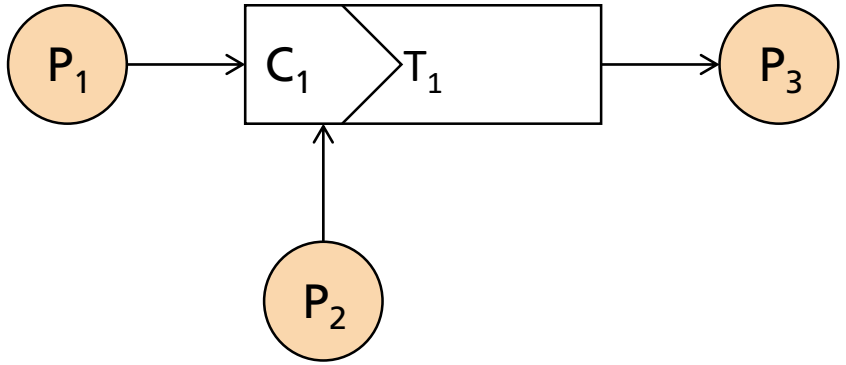
- Realistic task execution model
- Transitions have an associated runtime
- Tokens are not generated and consumed at the same time



Application Workflow – Supported Extensions

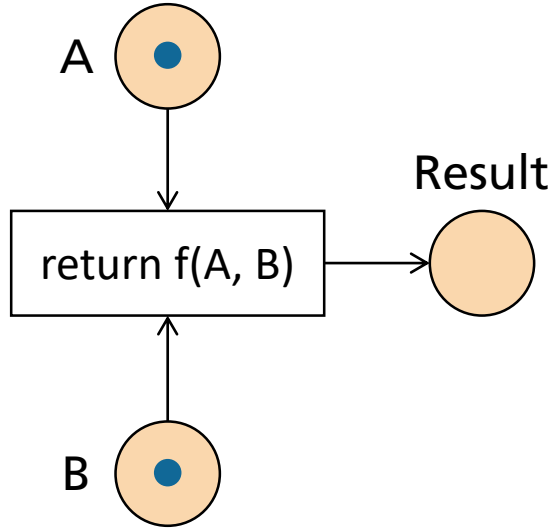
Conditional Transitions:

- Transition only triggers based on a token value pre-condition
- Turing completeness



Application Workflow – Tasks

- Tasks are extracted from the Petri net
- A task consists of:
  - 1 transition
  - 1 or more input places with 1 token each
  - 0 or more output places
- Tasks are reschedulable ⇒ Fault Tolerance
- Transitions are written in C/C++



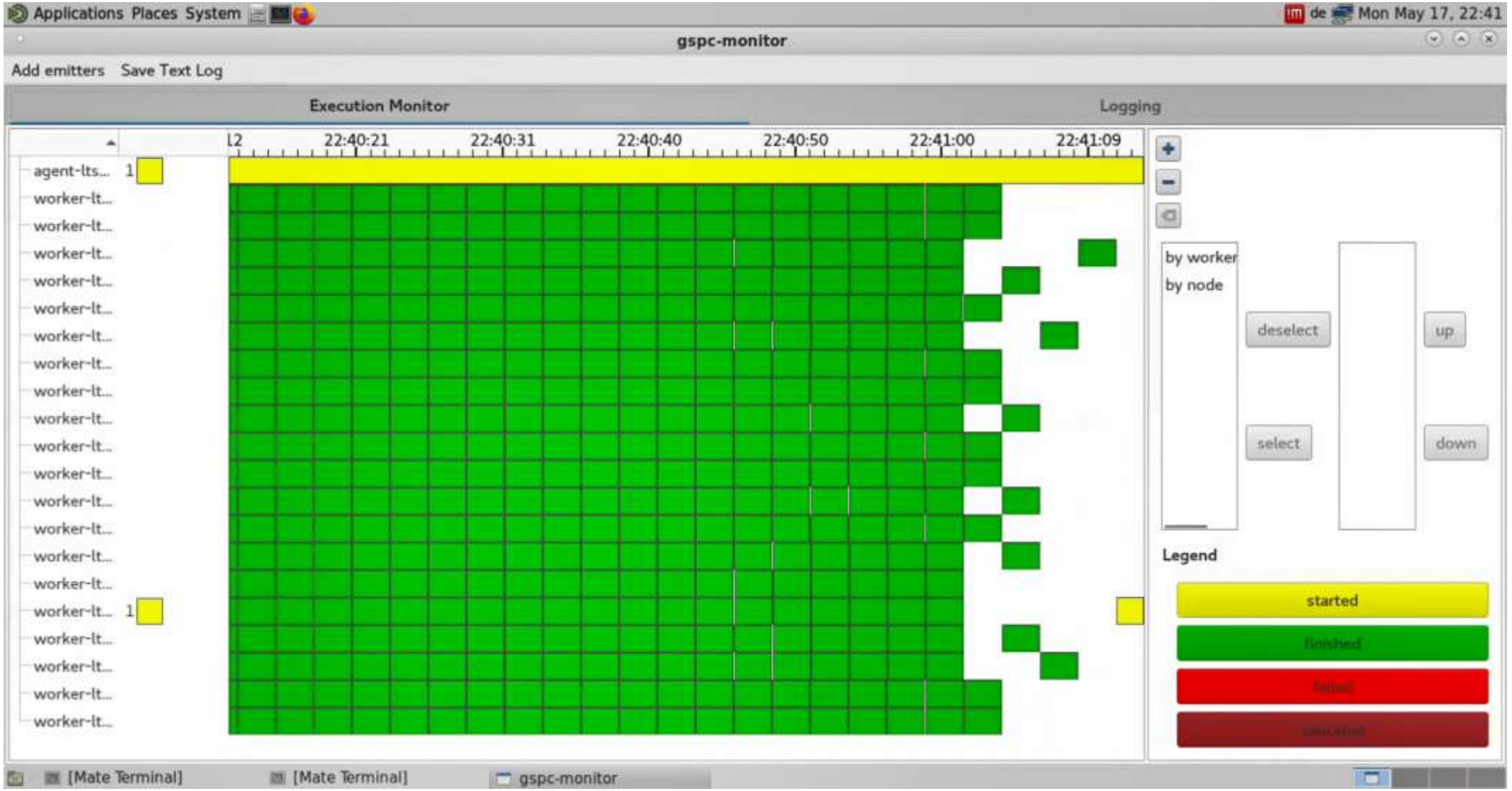
Outline

- Introduction
- Architecture
- Application Workflow
- **Summary**

Summary

- GPI-Space provides:
  - A task-based workflow management system
  - Automatic workflow parallelization
  - Fault tolerant task execution
- Application workflows are defined as Petri nets
  - Models Task and Data Parallelism
  - Automatic separation of data and compute
  - Independence of technologies used within transitions
    - OmpSs, TBB, Kokkos, Thrust, Cuda, Vulkan, …
    - GPI-2, MPI, …

Questions?


Applications Places System
Mon May 17, 22:41
gspc-monitor
Add emitters
Save Text Log
Execution Monitor
Logging
22:40:21
22:40:31
22:40:40
22:40:50
22:41:00
22:41:09
agent-lts...
worker-lt...
by worker
by node
deselect
select
up
down
Legend
started
[Mate Terminal]
[Mate Terminal]
gspc-monitor