Performance Analysis Tools for Petaflop Systems

Michael Gerndt
Technische Universität München
gerndt@in.tum.de

HPCC-06
September 13-15, 2006
Munich, Germany

Topics of interest

Architectures, languages, tools, application for HPC

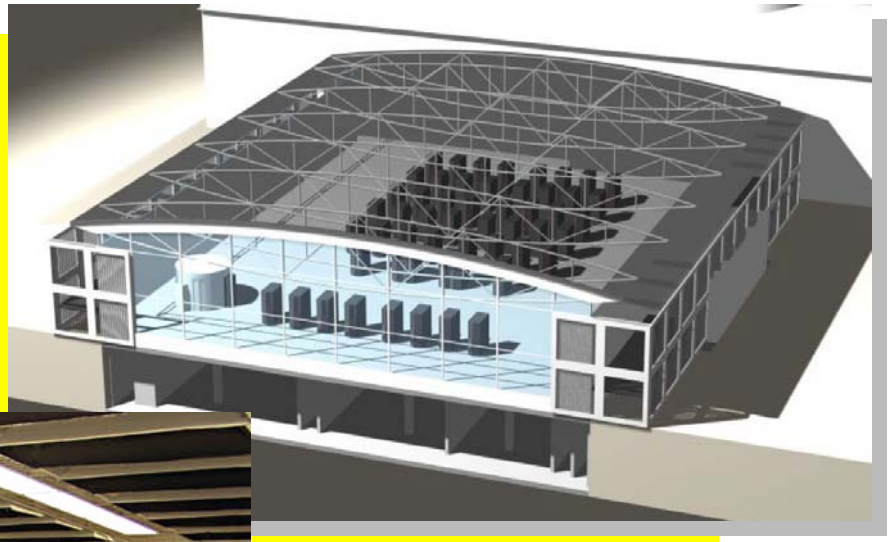
Special sessions:

- High-Performance Simulation of Reactive Flows
- Service Level Agreements
- Pervasive Computing Application & Security Service
- Automatic Performance Analysis of Parallel/Distributed Applications

Submission Deadline: **March 13th, 2005**

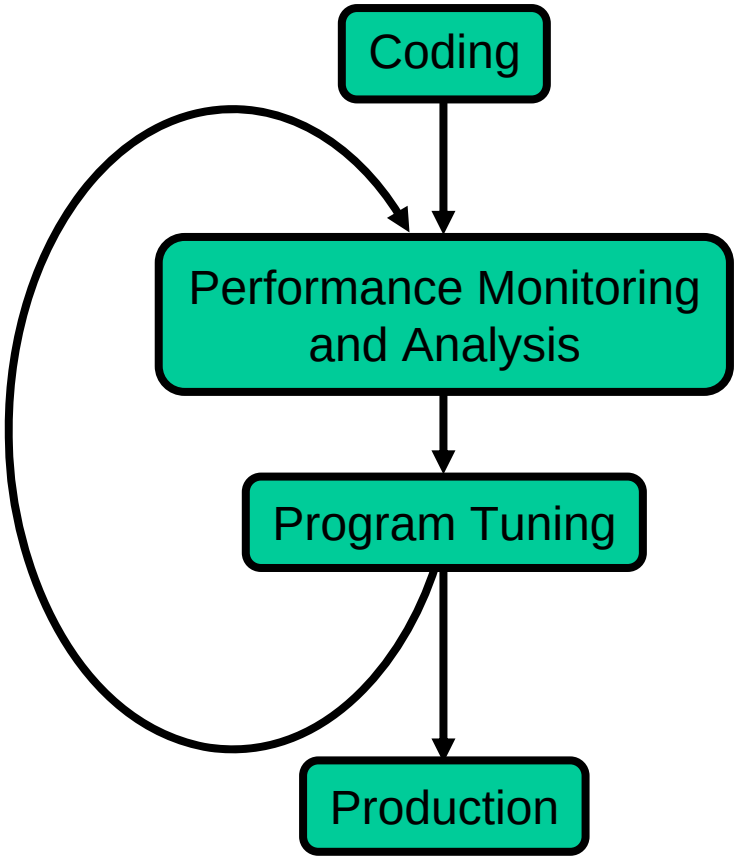
hpcc06.lrr.in.tum.de

Performance Analysis is Essential



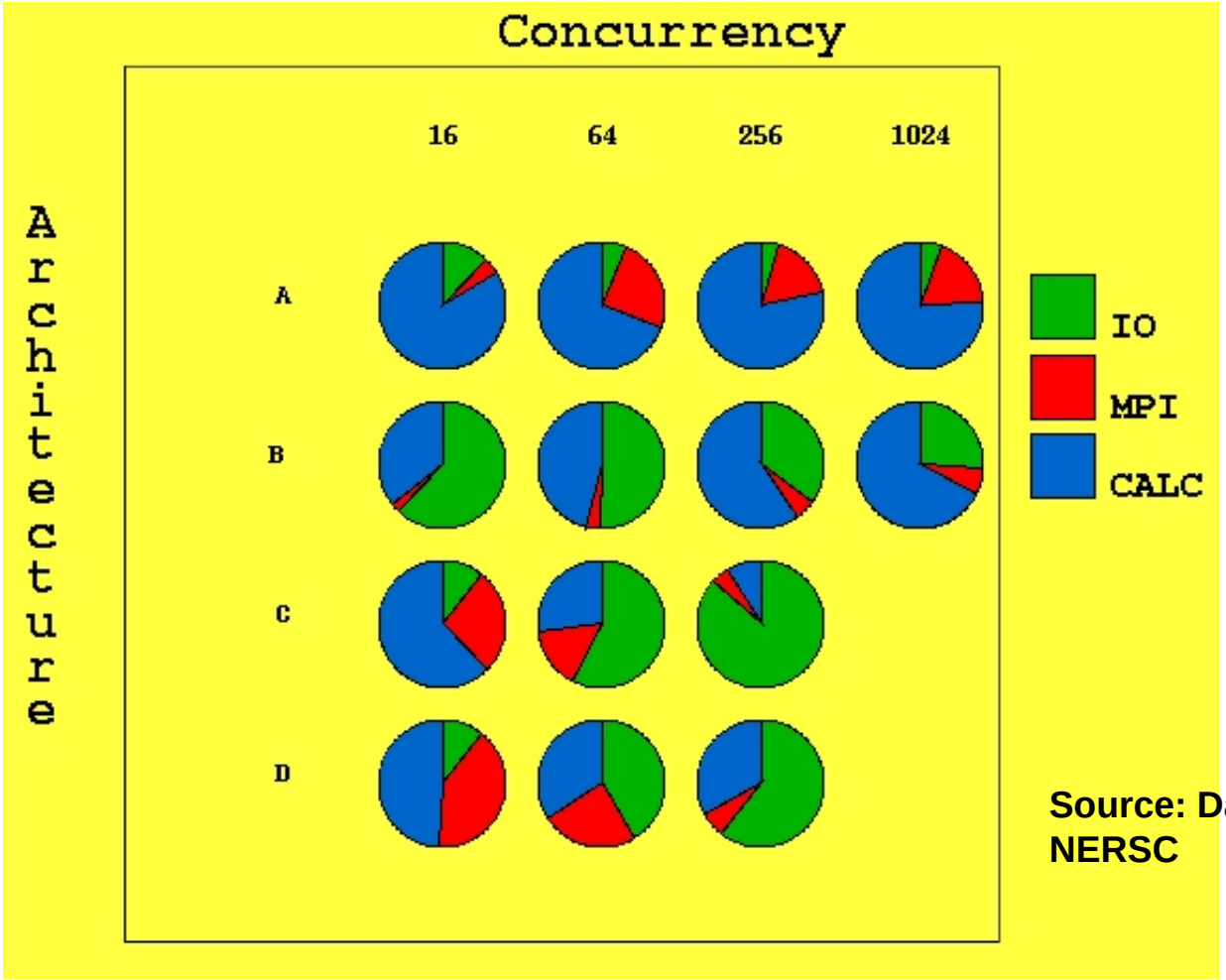
Performance Analysis for Parallel Systems

- Development cycle
  - Assumption: Reproducibility
- Instrumentation
  - Static vs Dynamic
  - Source-level vs binary-level
- Monitoring
  - Software vs Hardware
  - **Statistical profiles vs event traces**
- Analysis
  - Source-based tools
  - Visualization tools
  - Automatic analysis tools



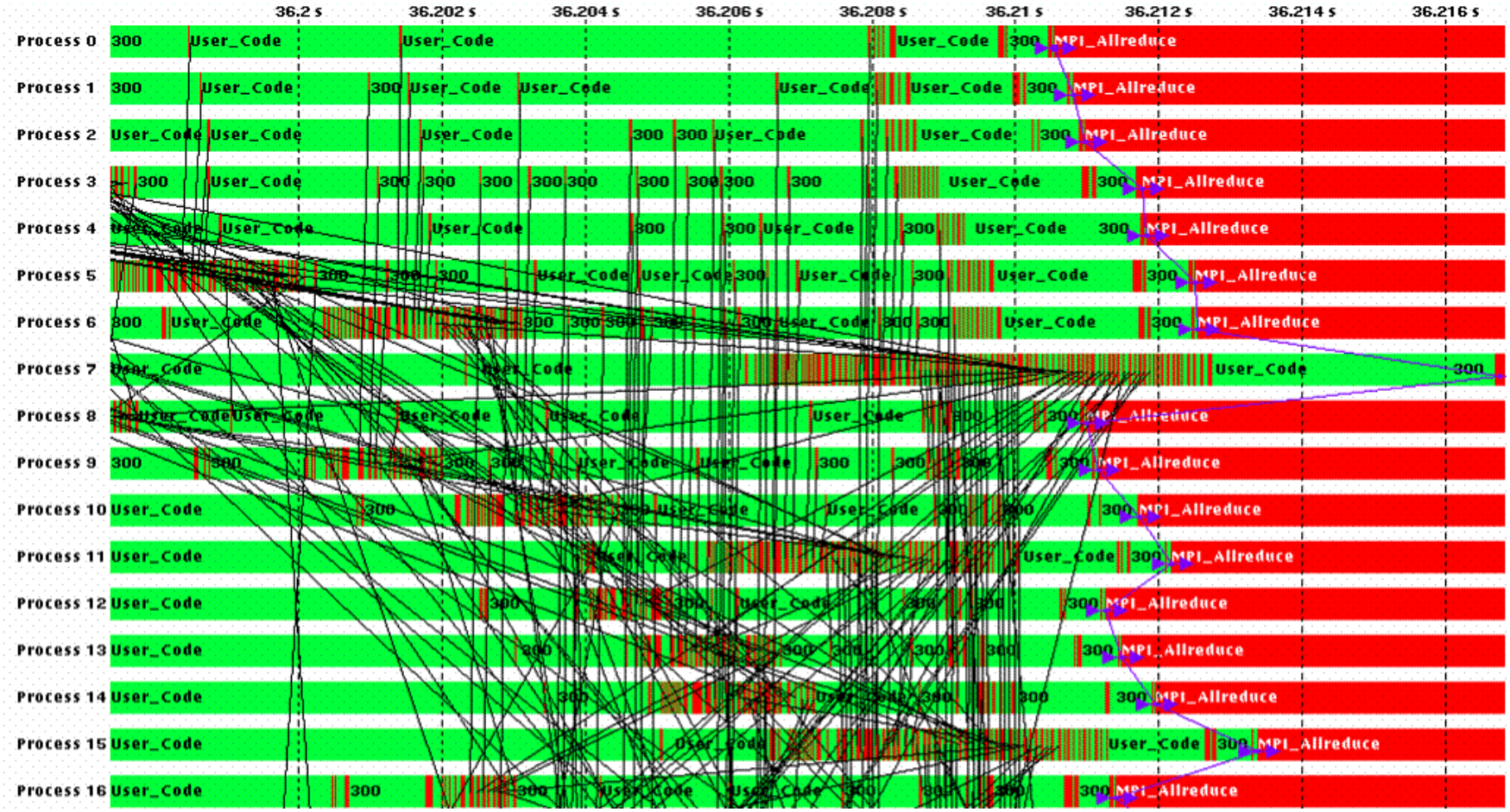
Scaling Portability: Profoundly Interesting

A high level description of the
performance of cosmology code MADCAP on four well known architectures.



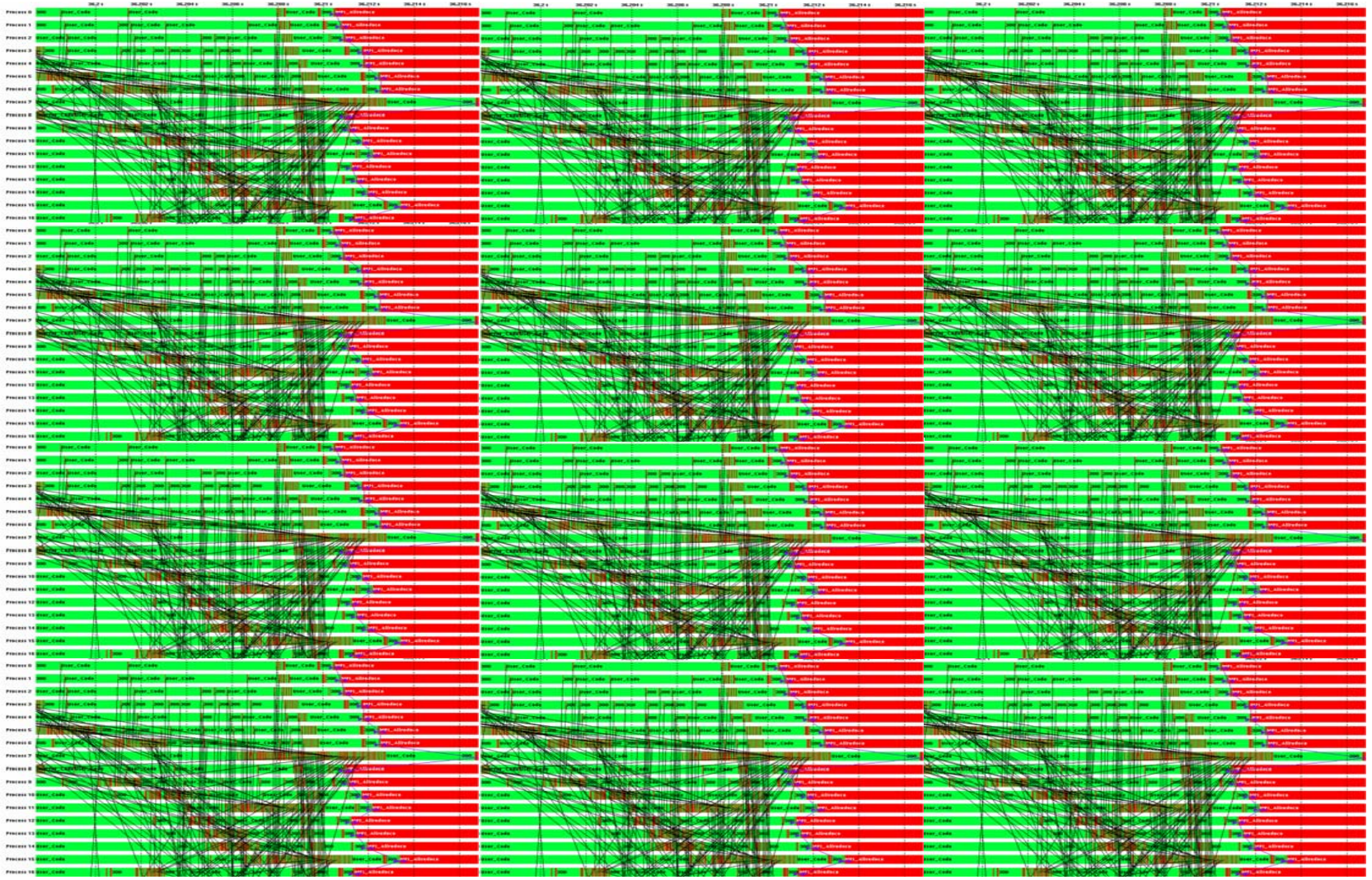
Source: David Skinner, NERSC

16 Way for 4 seconds



Source: Horst Simon

64 way for 12 seconds



Source: Horst Simon

Challenges for PA

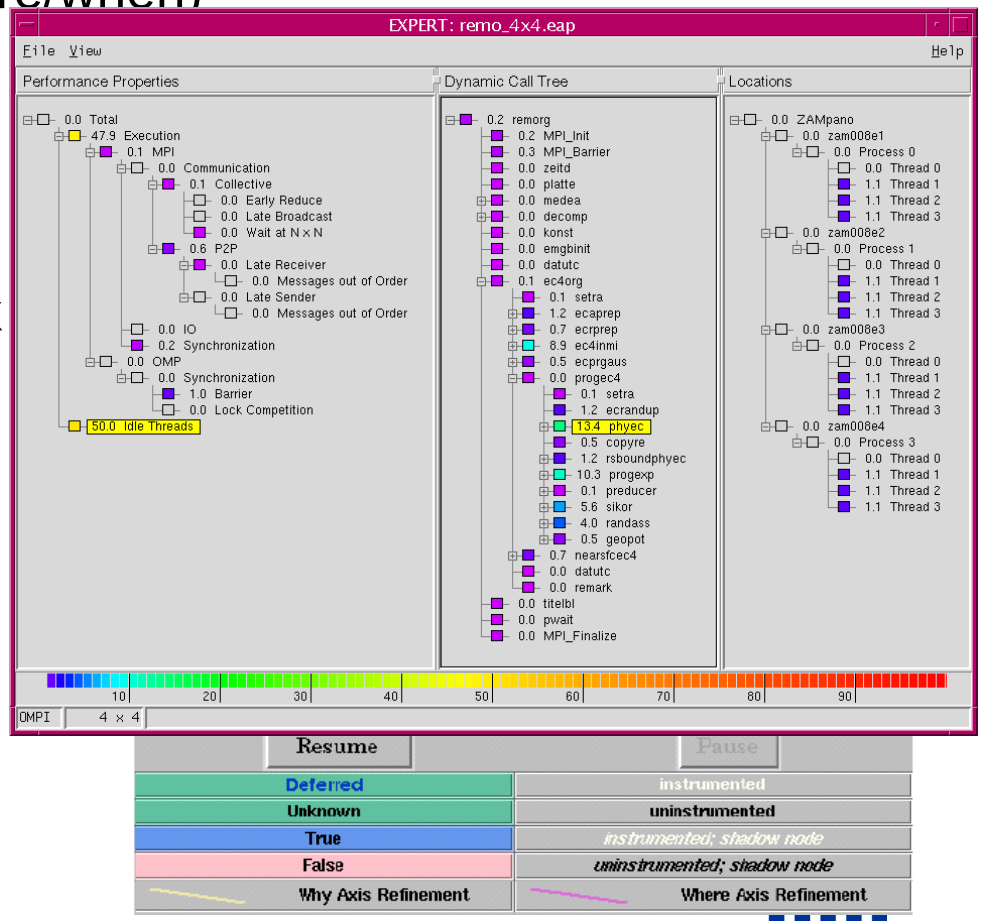
- Scalability
- Ease of use
- Single node performance

Answer - Automation

- Online analysis
  - Huge performance data sets need not be stored
- Distributed analysis
  - Data sets need not be collected
- Use expert’s knowledge
  - User need not be aware of intrinsic properties of architectures and performance tools

Automated Performance Analysis: Examples

- Paradyn/Performance Consultant:
  - Dynamic instrumentation
  - W3 Search model (why/where/when)
  - www.paradyn.org
- Expert:
  - Searching program traces
  - www.fz-juelich.de/zam/kojak



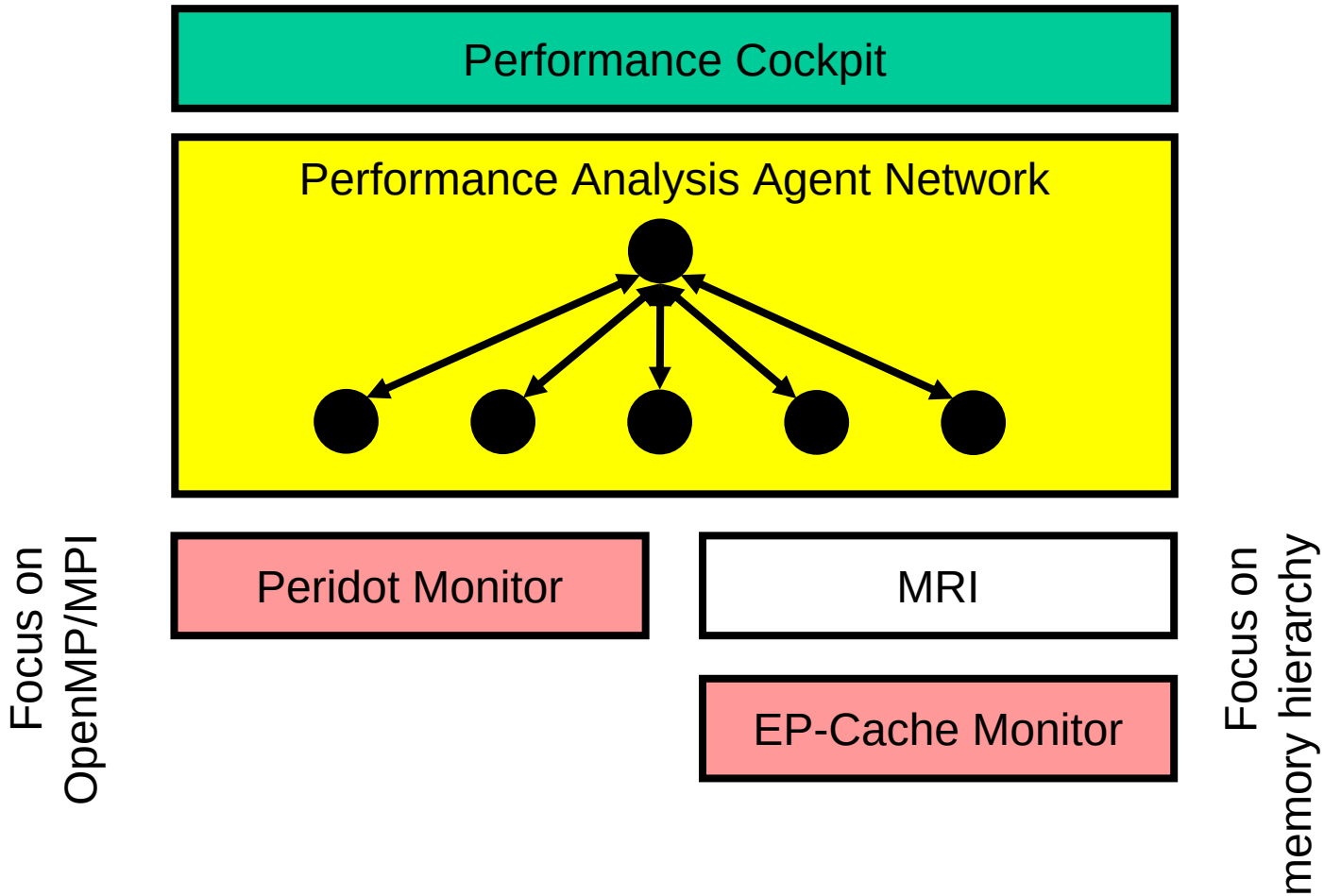
Periscope Project at Technische Universität München

- Formal specification of performance properties
- Automated search
- Online analysis
- Distributed search

ASL: APART Specification Language

```
property communication_costs (
      Region r,
      Experiment e,
      Region rank_basis)
{
LET
  float cost = summary(r,e).CommTime;
IN
  CONDITION:   cost>0;
  CONFIDENCE:  1;
  SEVERITY:    cost/duration(rank_basis,e);
}
```

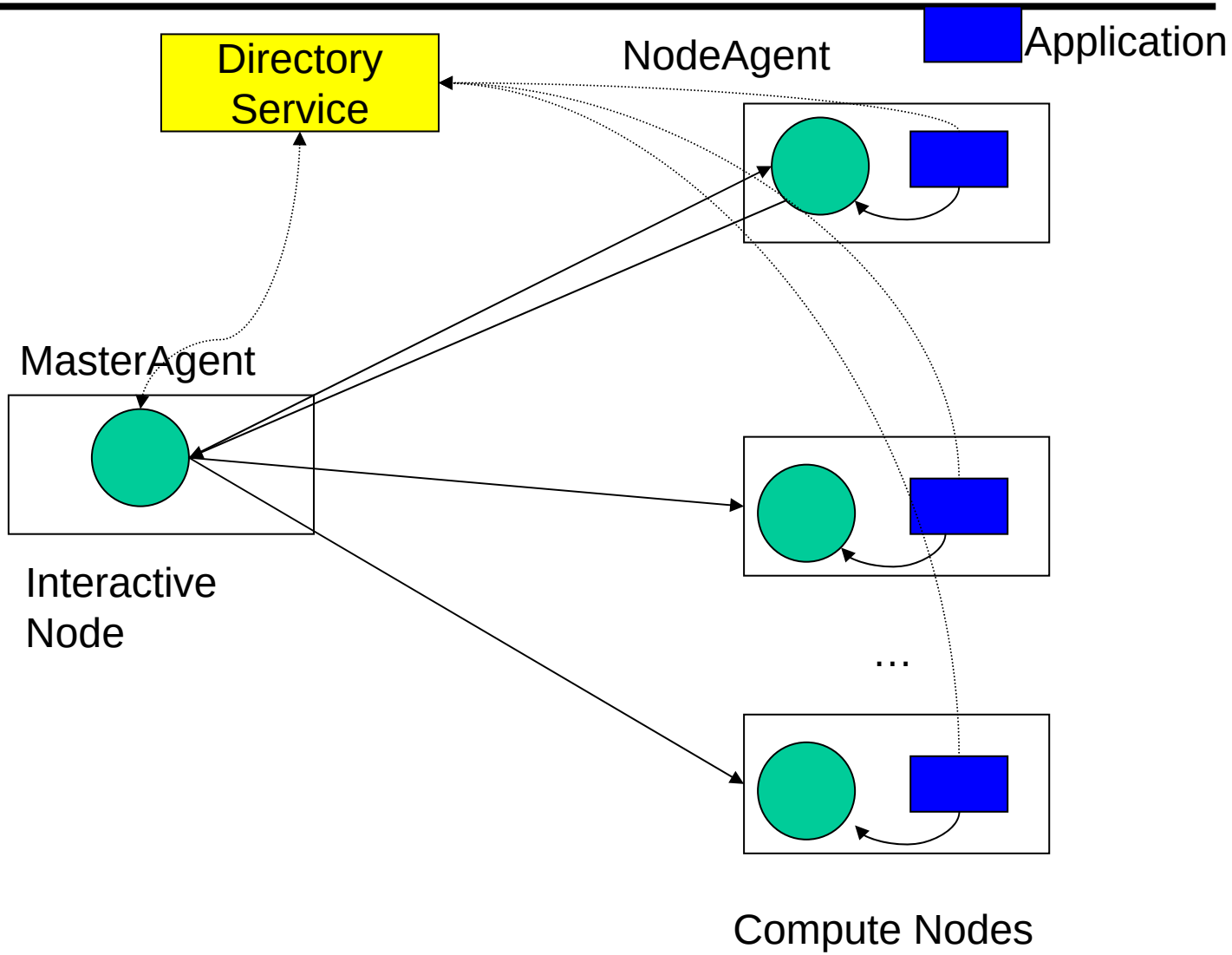
Periscope Design



Agent Search Strategies – Incremental Analysis

- Based on program phases – intervall of execution
  - Defined by phase regions
  - *Repetitive* and *execute once* phase regions
  - Phase boundaries have to be global (SPMD programs)
- Two phase-based strategies
  - Region nesting and data structure-based refinement
  - Property hierarchy-based refinement

Usage Scenario



Final Result

```
LC1MissesOverMemRef (LOOP, bound.f, 139)
  Severity: 0.68164
LC1WriteMissesOverMemRef (LOOP, bound.f, 139 )
  Severity: 0.671389
LC1WriteMissesOverMemRef (LOOP, bound.f, 139 )
  Severity: 0.671016
  Data Structure: UN
LC1ReadMissesOverMemRef (LOOP, bound.f, 673 )
  Severity: 0.108437
LC1WriteMissesOverMemRef (LOOP, bound.f, 673 )
  Severity: 0.108429
LC1MissesOverMemRef (LOOP, bound.f, 28 )
  Severity: 0.103128
  Data Structure: UN
```

Summary

- Solution to challenges for PA on Petascale Computers
  - Automation
  - Distribution
  - New hardware performance monitors