Performance Analysis with Periscope

M. Gerndt, V. Petkov,
Y. Oleynik, S. Benedict
Technische Universität München

periscope@lrr.in.tum.de

October 2010

Outline

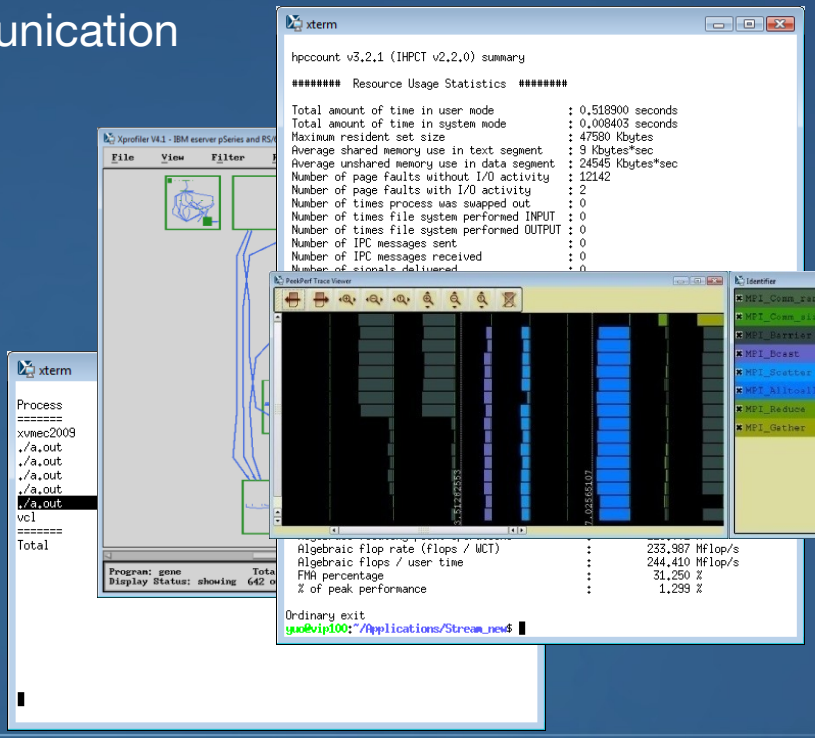
- Motivation
- Periscope overview
- Periscope performance analysis model
- Tool architecture
- Performance analysis automation
- Periscope GUI
- Hands-on exercise

Motivation

- Performance analysis procedure on POWER6 as a example:
  - Use Tprof to pinpoint time consuming subroutines
  - Use Xprofiler (GUI for gprof) to understand call graph
  - Use hpmcount (libhpm) to measure Hardware Counters
  - Use mpitrace to investigate mpi communication

- Problems:
  - Routine, time consuming
  - Error prone
  - Not scalable
  - Requires deep hardware knowledge

- Solution:
  - Performance analysis automation



Periscope

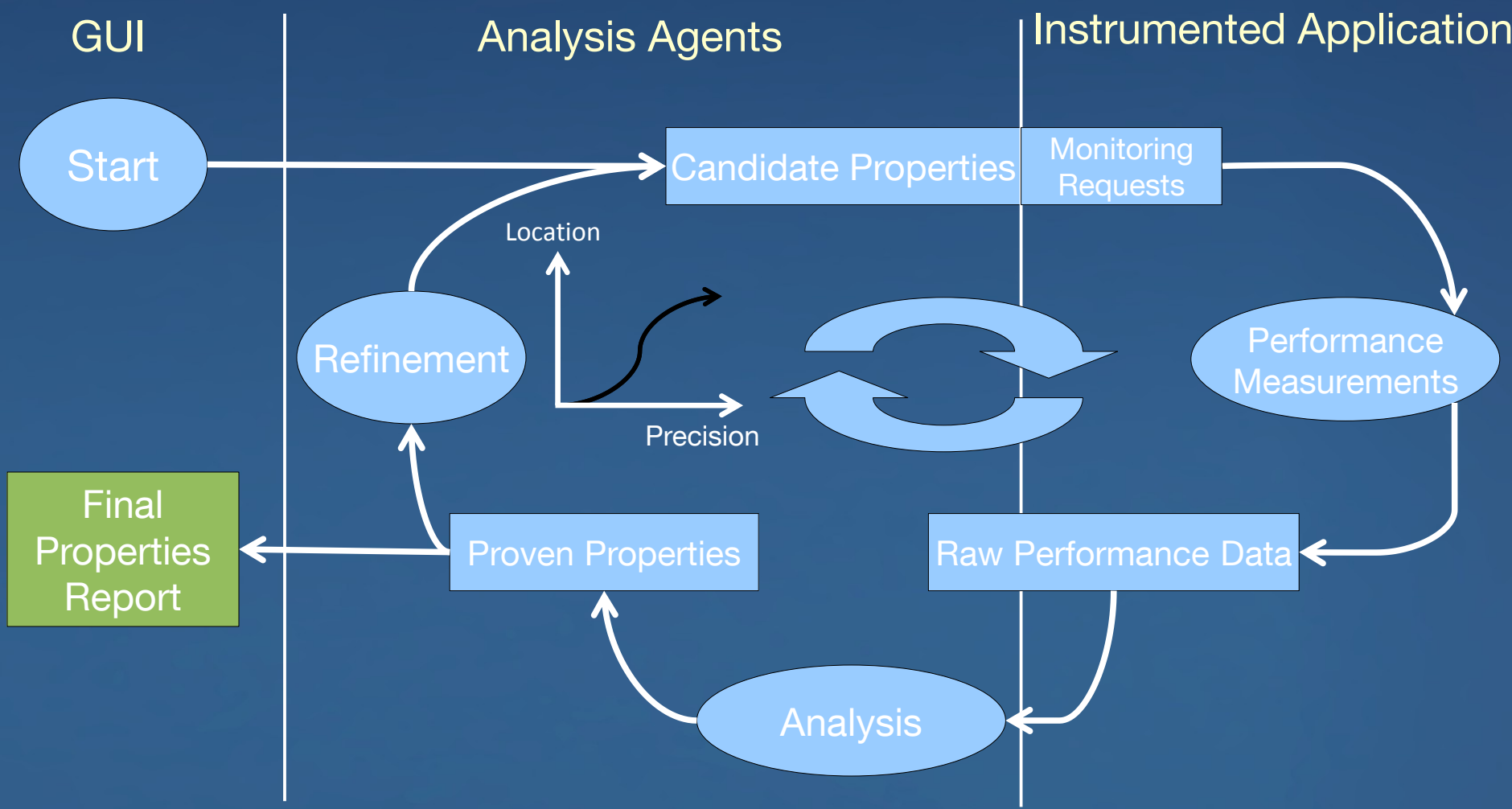
- Iterative online analysis
  - Measurements are configured, obtained and evaluated on the fly
  - no tracing!
- Distributed architecture
  - Analysis performed by multiple distributed hierarchical agents
- Automatic bottlenecks search
  - Based on performance optimization experts' knowledge
- Enhanced GUI
  - Eclipse based integrated development and performance analysis environment
- Instrumentation
  - Fortran, C/C++

Previous Development Projects

- Previous projects (2001-2005)
  - Peridot
  - EP-Cache
- Periscope (2005-2008)
  - Develop an automated distributed performance analysis tool.
- ISAR (2009-2011)
  - Productization of Periscope
- SILC (2009-2011)
  - Development of a scalble monitoring infrastructure for Vampir, Scalasca, and Periscope

Prepare your application

- Automatic source code instrumentation
  - Selective in region type
  - Generates information file about code structure (SIR – Standard Intermediate Program Representation)
- Modification of compiler call in the makefile
  - psc_instrument
  - Controlled by psc_inst_config
- Link with monitoring library and MPI profiling library


GUI
Analysis Agents
Instrumented Application
Start
Candidate Properties
Monitoring Requests
Location
Refinement
Precision
Performance Measurements
Final Properties Report
Proven Properties
Raw Performance Data
Analysis

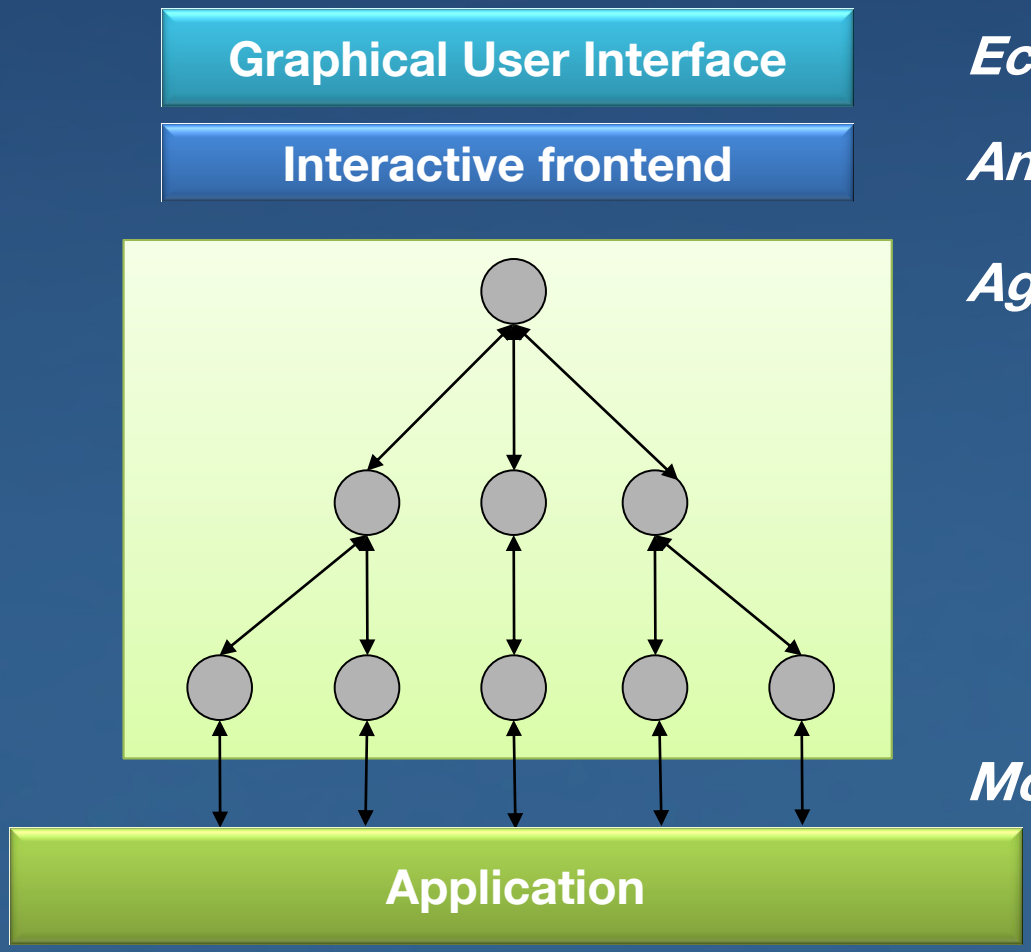
Application Phases

- Period of program‘s execution
- Phases are defined by program regions called phase regions
  - Standard regions
    - Full program
    - Functions
    - Parallel loops
  - User regions
- Phase boundaries have to be global (SPMD programs)

Multi-phase Search Strategies

- Suspend/Resume of phase region
- Or automatic restart of application (not Bluegene)

Distributed Architecture



Eclipse-based GUI

Analysis control

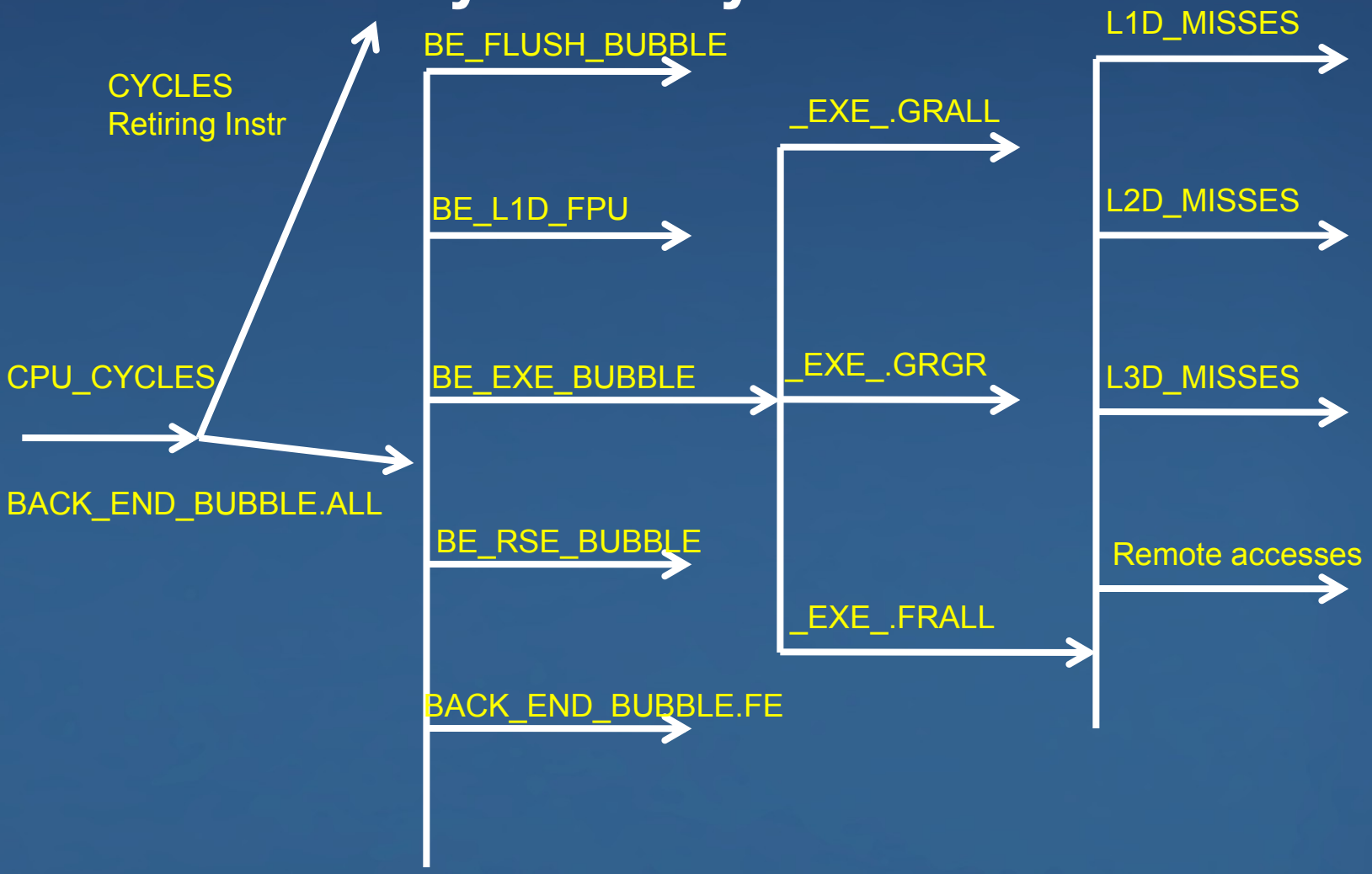
Agents network

Monitoring Request Interface

Automatic search for bottlenecks

- Automation based on formalized expert knowledge
  - Potential performance problems → properties
  - Efficient search algorithm → search strategies
- Performance property
  - Condition
  - Confidence
  - Severity
- Performance analysis strategies
  - Itanium2 Stall Cycle Analysis
  - IBM POWER6 Single Core Performance Analysis
  - MPI Communication Pattern Analysis
  - Generic Memory Strategy
  - OpenMP-based Performance Analysis
  - Scalability Analysis – OpenMP codes

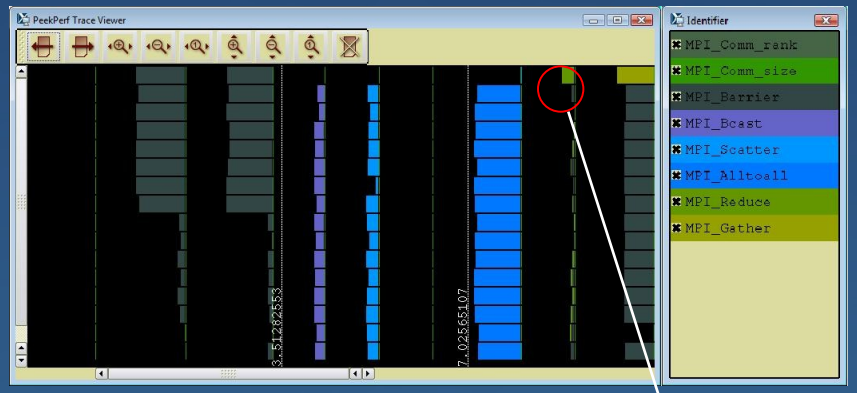
Itanium Stall Cycle Analysis



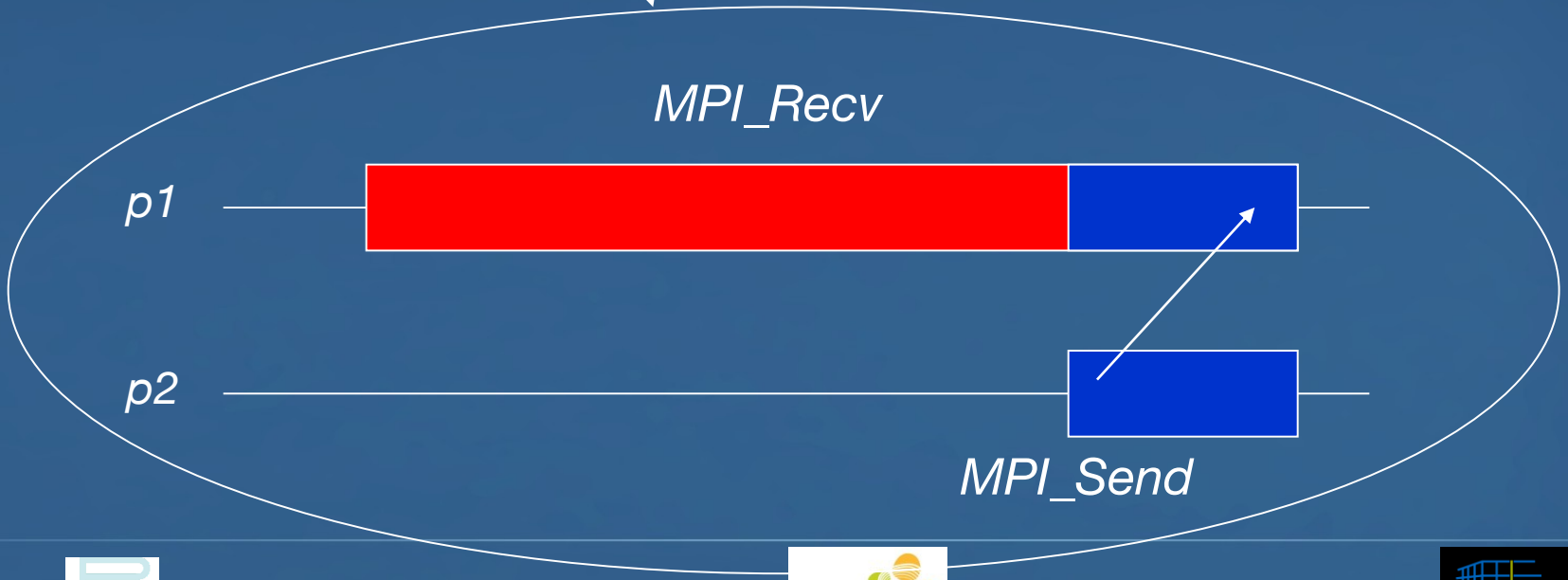
Properties

- StallCycles(Region, Rank, Thread, Metric, Phase)
  - Condition
    - Percentage of lost cycles >30%
  - Severity
    - Percentage of lost cycles
- StallCyclesIntegerLoads
  - Requires access to two counters
- L3MissesDominatingMemoryAccess
  - Condition
    - Importance of L3 misses (theoretical latencies)
  - Severity
    - Importance multiplied by actual stall cycles

MPI Communication Patterns Analysis



- Automatic detection of wait patterns
- Measurement on the fly
- **No tracing required!**



MPI Performance Properties

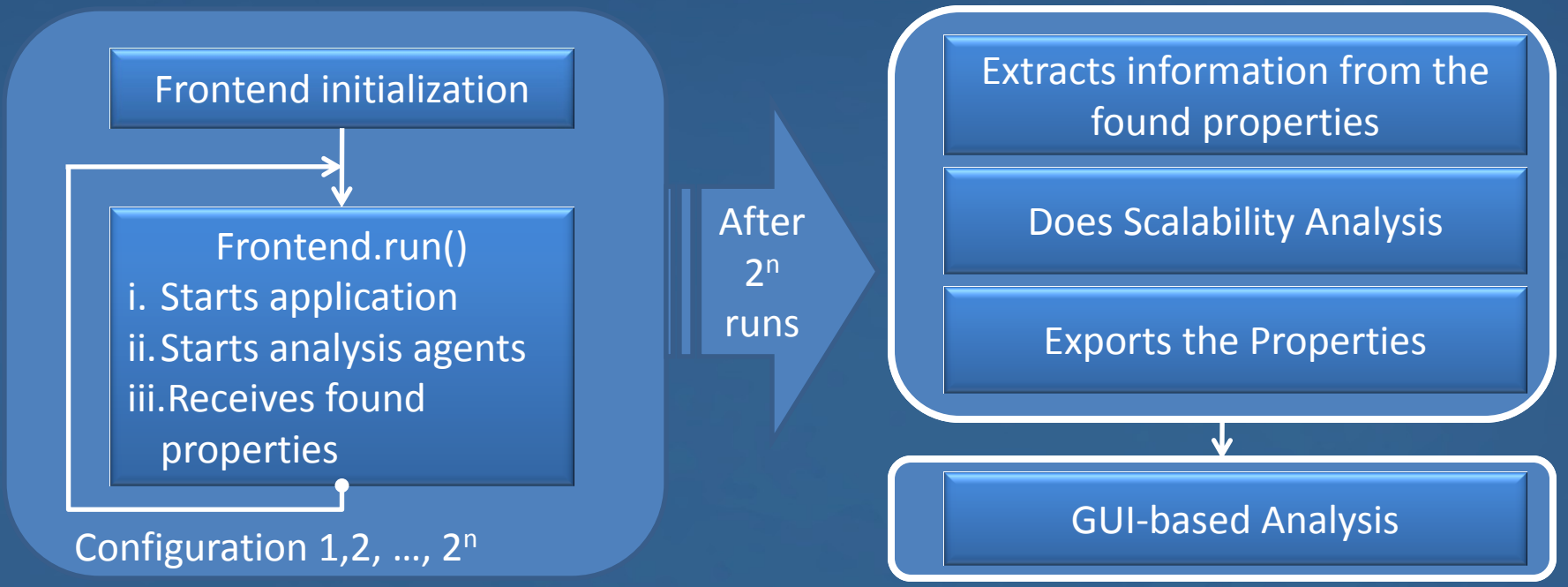
- Excessive MPI time in receive due to late sender
- Excessive MPI time due to late root in broadcast
- Excessive MPI time in root due to late process in reduce
- Excessive MPI time in ... (1xN, Nx1, 1x1, NxN)
- Excessive MPI time due to many small messages
- Excessive MPI communication time

OpenMP-based Performance Properties

- Searches OpenMP-based performance problems in a single step
- Properties are divided into four major domains
  - Startup and Shutdown Overhead
  - Load Imbalance in OpenMP regions:
    - Parallel region
    - Parallel loop
    - Explicit barrier
    - Parallel sections
      - Not enough sections
      - Uneven sections
  - Seq. Computation in parallel regions:
    - Master region
    - Single region
    - Ordered loop
  - OpenMP Synchronization properties
    - Critical section overhead property
    - Frequent atomic property

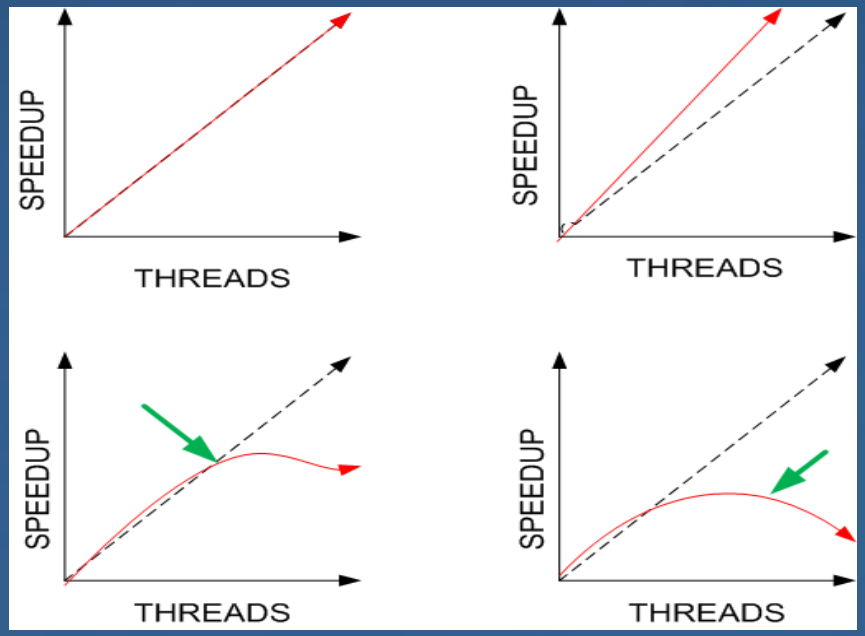
Scalability Analysis – OpenMP codes

- Identifies the OpenMP code regions that do not scale well
- Scalability Analysis is done by the frontend
- **No need to manually configure the runs and find the speedup!**



Scalability Analysis – Properties

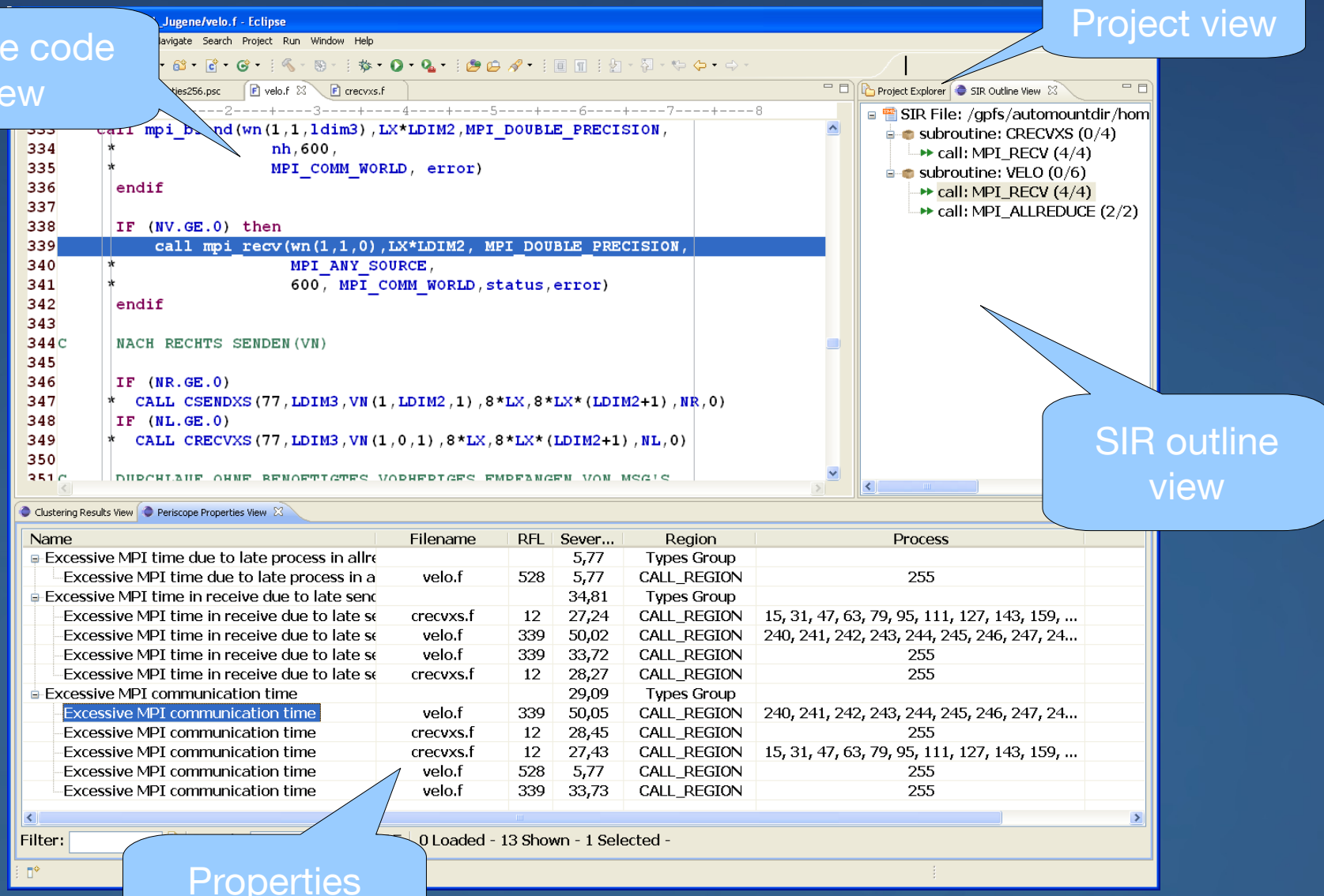
- Meta-Properties
  - Properties occurring in all configurations
  - Property with increasing severity across the configurations



- Speedup-based Prop.
  - Linear Speedup
  - Super linear Speedup
  - Linear Speedup failed for the first time
  - Speedup Decreasing

Run specific properties

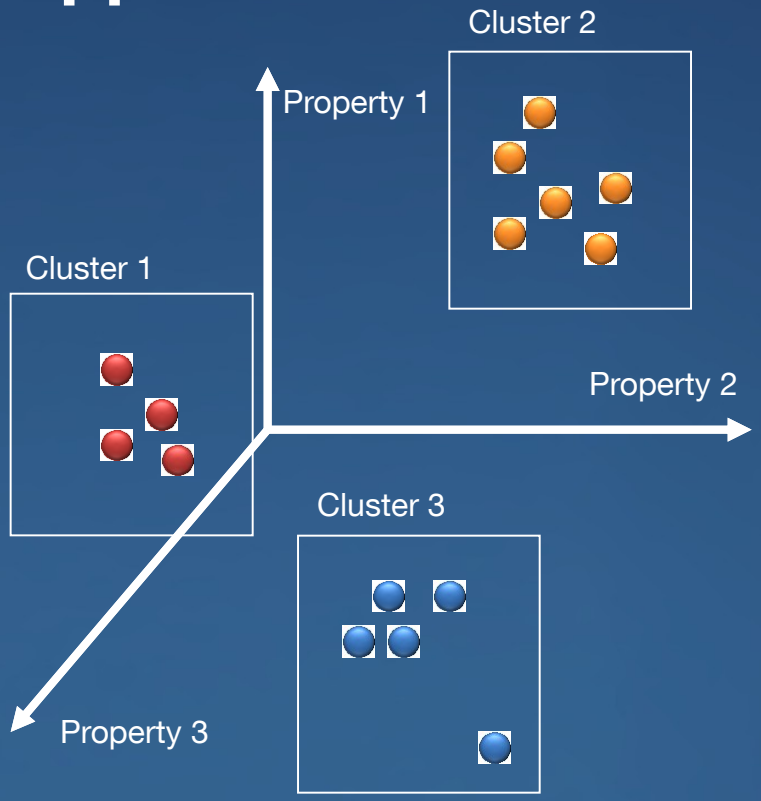
- Code region with the lowest speedup
- Low Speedup based on a threshold



Name	Filename	RFL	Sever...	Region	Process
Excessive MPI time due to late process in allre			5,77	Types Group	
Excessive MPI time due to late process in a	velo.f	528	5,77	CALL_REGION	255
Excessive MPI time in receive due to late send			34,81	Types Group	
Excessive MPI time in receive due to late se	crecvxs.f	12	27,24	CALL_REGION	15, 31, 47, 63, 79, 95, 111, 127, 143, 159, ...
Excessive MPI time in receive due to late se	velo.f	339	50,02	CALL_REGION	240, 241, 242, 243, 244, 245, 246, 247, 24...
Excessive MPI time in receive due to late se	velo.f	339	33,72	CALL_REGION	255
Excessive MPI time in receive due to late se	crecvxs.f	12	28,27	CALL_REGION	255
Excessive MPI communication time			29,09	Types Group	
Excessive MPI communication time	velo.f	339	50,05	CALL_REGION	240, 241, 242, 243, 244, 245, 246, 247, 24...
Excessive MPI communication time	crecvxs.f	12	28,45	CALL_REGION	255
Excessive MPI communication time	crecvxs.f	12	27,43	CALL_REGION	15, 31, 47, 63, 79, 95, 111, 127, 143, 159, ...
Excessive MPI communication time	velo.f	528	5,77	CALL_REGION	255
Excessive MPI communication time	velo.f	339	33,73	CALL_REGION	255

Clustering support

- Properties summarization
  →Metaproperties
- Needed for peta-scale PA
- Identify *hidden* behavior
- Supported by aggregation in the agent hierarchy



Cluster 1 CPUs: 7-10,16

Cluster 2 CPUs: 2-3,5,11,13-14

Cluster 3 CPUs:1,4,6,12,15

Thank you for your attention!

- Current version 1.3
  - Available under: http://www.lrr.in.tum.de/periscope/Download
- Supported architectures
  - SGI Altix 4700 Itanium2
  - IBM Power575 POWER6
  - IBM BlueGene/P
  - x86-based architectures
- Further information:
  - Periscope web page: http://www.lrr.in.tum.de/periscope
  - Contact us directly at: periscope@lrr.in.tum.de