

# Performance Analysis with Periscope

M. Gerndt, Y. Oleynik  
Technische Universität München

[oleynik@in.tum.de](mailto:oleynik@in.tum.de)

October 2013

# Outline

- Periscope overview
- Periscope performance analysis model
- Performance analysis automation
- Periscope GUI

# Periscope

- **Distributed architecture**
  - Analysis performed by multiple distributed hierarchical agents
- **Iterative online analysis**
  - Measurements are configured, obtained and evaluated on the fly
  - no tracing files needed
- **Automatic bottlenecks search**
  - Based on performance optimization experts' knowledge
- **Enhanced GUI**
  - Eclipse based integrated development and performance analysis environment
- **Instrumentation**
  - Done by Score-P

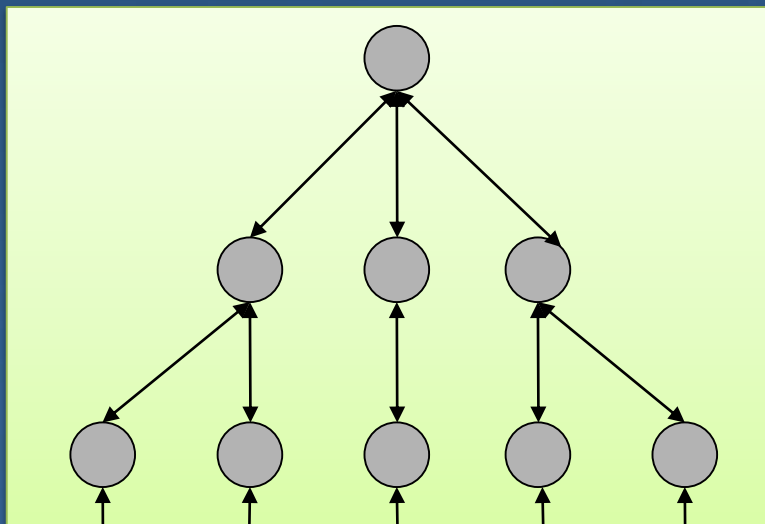
# Distributed Architecture

Graphical User Interface

*Eclipse-based GUI*

Interactive frontend

*Analysis control*

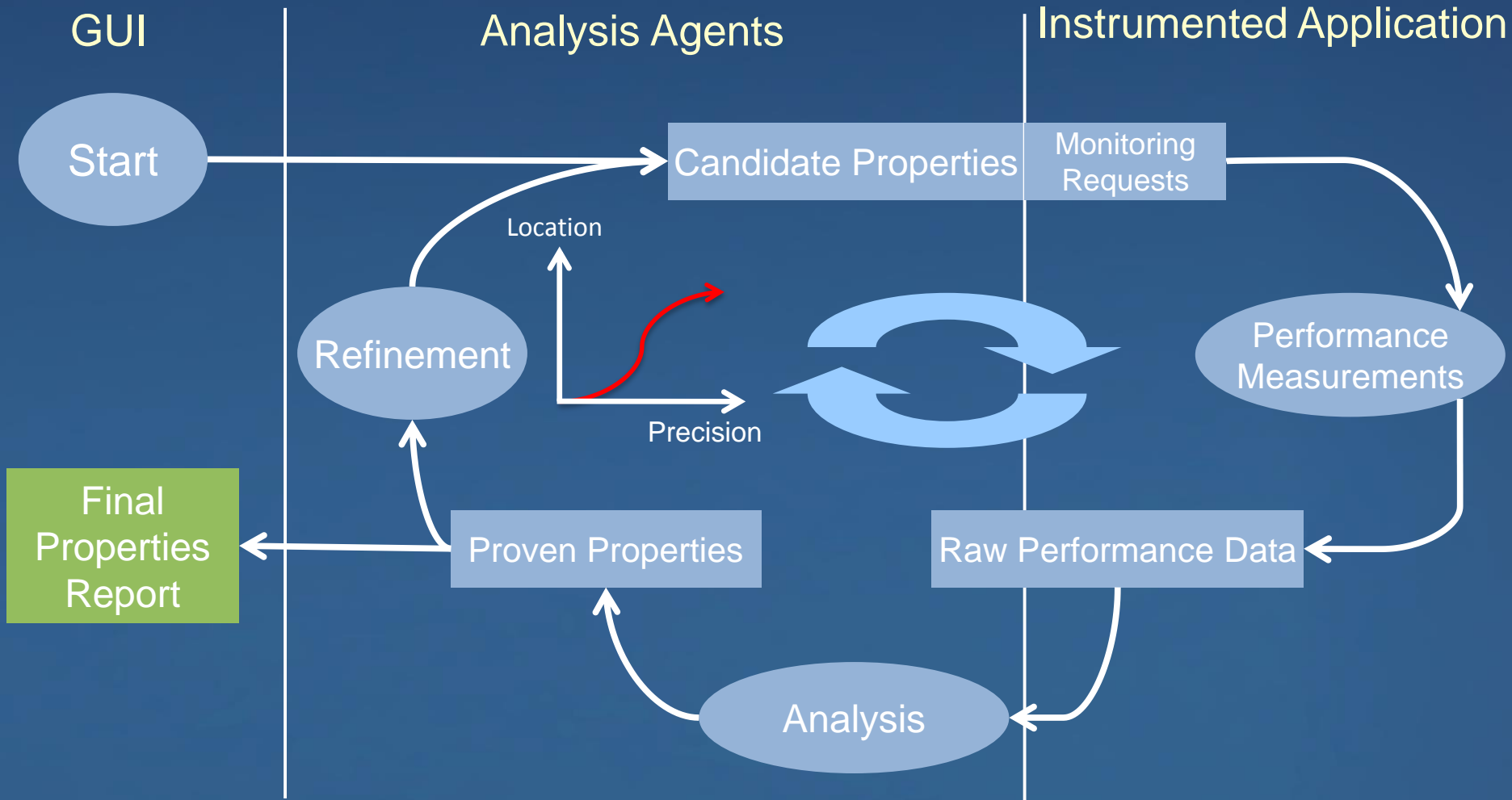


*Agents network*

MRIMonitor/Score-P  
Application

*Monitoring Request Interface*

# Iterative Online Analysis



# Periscope Phases

- Periscope performs multiple iterative performance measurement experiments on the basis of *Phases*:
  - All measurements are performed inside phase
  - Begin and end of phase are global synchronization points
  - Automatic restart might be necessary
- Region needs to be marked as an Online Access Phase to use the Score-P Online Access Interface
  - Typically main loop of application → no need for restart, faster analysis
  - Unnecessary code parts are not measured → less measurements overhead
  - Severity value is normalized on the main loop iteration time → more precise performance impact estimation



# Definition of Online Access Phases

```
#include <scorep/SCOREP_User.h>
void foo()
{
    SCOREP_USER_REGION_DEFINE( my_region_handle )
    for(i=0;...
    {
        SCOREP_USER_OA_PHASE_BEGIN( my_region_handle, \
            „ITERATION“,SCOREP_USER_REGION_TYPE_COMMON )
        // do something
        ...
        SCOREP_USER_OA_PHASE_END( my_region_handle )
    }
}
```

# 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
  - Westmere Single-node Analysis
  - Itanium2 Stall Cycle Analysis
  - IBM POWER6 Single Core Performance Analysis
  - MPI Communication Pattern Analysis
  - Generic Memory Analysis
  - OpenMP-based Performance Analysis
  - Scalability Analysis – OpenMP codes



# Example Properties

- **StallCycles (Region, Rank, Thread, Metric, Phase)**
  - Condition
    - Percentage of lost cycles >30%
  - Severity
    - Percentage of lost cycles
- **MPI Late Sender**
  - Automatic detection of wait patterns
  - Measurement on the fly
  - No tracing required
- **OpenMP Synchronization properties**
  - Critical section overhead property
  - Load imbalance



Source code view

```

File Edit Refactor Navigate Search Project Run Window Help
45528.psc rhs.f properties_OMP_55591.psc x_solve.f
-----
c This function computes the left hand side in the xi-direction
c -----
!$OMP PARALLEL DEFAULT(SHARED) PRIVATE(n,m,i,j,k,ysize)
!$OMP SHARED(dx5,dx4,dx3,dx2,dx1,tx2,tx1,dt,c1345,c3c4,con43,c1,c2,
!$OMP nx,ny,nz)
ysize = nx-1
-----
c determine a (labeled f) and n jacobians
c -----
!$OMP DO
do k = 1, nz-2
do j = 1, ny-2
do i = 0, ysize

```

Project view

- NPB3.3-BT-MZ
  - BT-MZ
  - bin
  - bin.scorep
  - common
  - config
  - jobscript
  - LU-MZ
  - SP-MZ
  - sys
  - Makefile
  - README

Periscope Properties View Clustering Results View Console Periscope Table View

Name	Filename	RFL	Severity	Confidence	Region	Process	Thread
Load Imbalance in parallel region	z_solve.f	43	2.34	1.00	Parallel region	3	0
Load Imbalance in parallel region	z_solve.f	43	4.37	1.00	Parallel region	0	0
Load Imbalance in parallel region	x_solve.f	46	6.92	1.00	Parallel region	1	0
Load Imbalance in parallel region	y_solve.f	43	7.76	1.00	Parallel region	1	0
Load Imbalance in parallel region	z_solve.f	43	3.43	1.00	Parallel region	1	0
Load Imbalance in parallel region	x_solve.f	46	7.46	1.00	Parallel region	2	0
Load Imbalance in parallel region	y_solve.f	43	7.84	1.00	Parallel region	2	0

Properties view

# Thank you for your attention!

- Current version 1.5
  - Available under: <http://www.lrr.in.tum.de/periscope/Download>
- Further information:
  - Periscope web page: <http://www.lrr.in.tum.de/periscope>
  - Contact us directly at: [periscope@lrr.in.tum.de](mailto:periscope@lrr.in.tum.de)