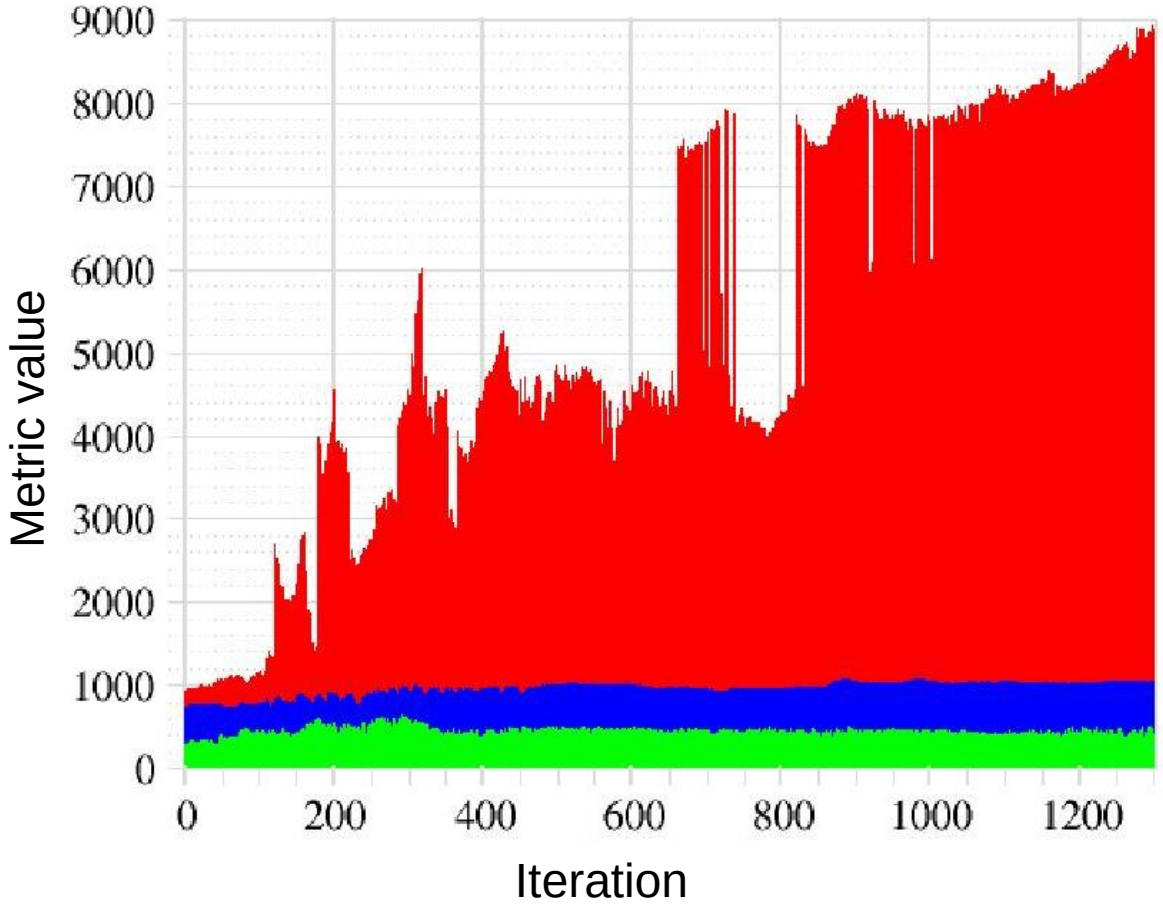
Automatic Characterization of Performance Dynamics with Periscope

Yury Oleynik
(oleynik@in.tum.de)
Chair for Computer Architecture
Techinische Universität München

Outline

- Performance Dynamics
- Periscope overview & limitations
- Introducing temporal dimension
- Automatic analysis of performance dynamics
- Examples
- Summary

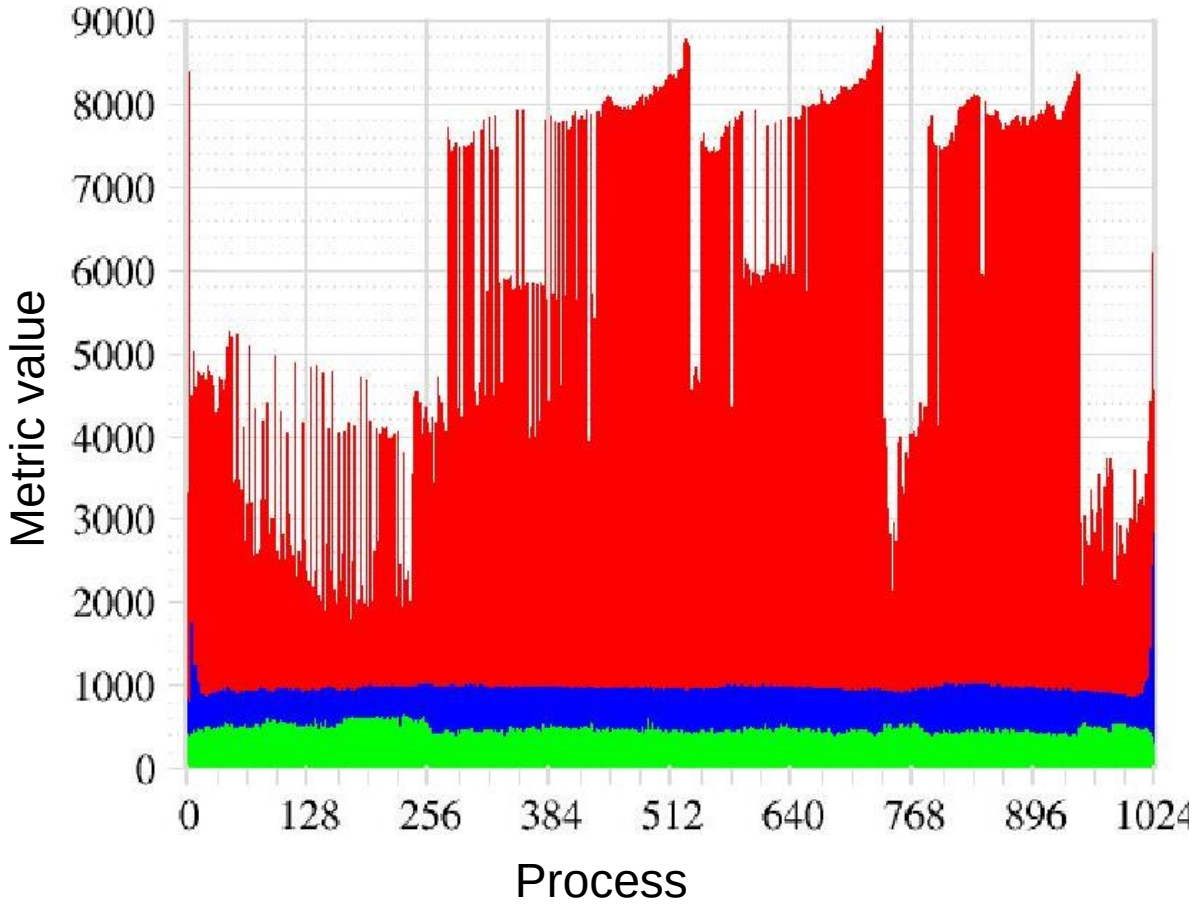
Motivating example



Min value over processes
Median value over processes
Max value over processes

[Szebenyi, Zoltán, Brian JN Wylie, and Felix Wolf. "Scalasca parallel performance analyses of PEPC." In Euro-Par 2008 Workshops-Parallel Processing, 2009.]

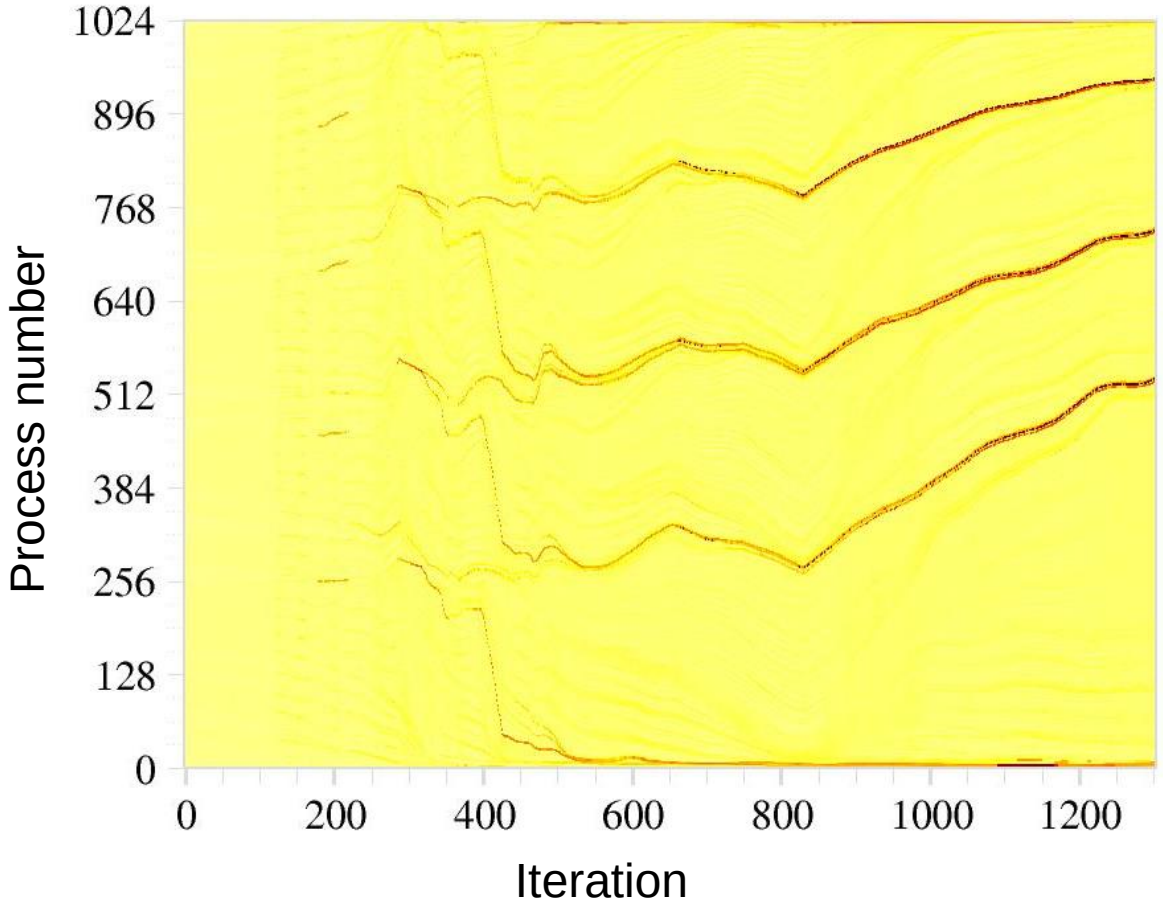
Motivating example



Min value over iterations
Median value over iterations
Max value over iterations

[Szebenyi, Zoltán, Brian JN Wylie, and Felix Wolf. "Scalasca parallel performance analyses of PEPC." In Euro-Par 2008 Workshops-Parallel Processing, 2009.]

Motivating example



[Szebenyi, Zoltán, Brian JN Wylie, and Felix Wolf. "Scalasca parallel performance analyses of PEPC." In Euro-Par 2008 Workshops-Parallel Processing, 2009.]

Lesson learned

- Location and severity of performance bottlenecks is time-dependent
- Performance dynamics can be observed at various time scales and in various time intervals
- Manual investigation is tedious due to dimensionality of measurements

Performance dynamics: tools face challenges

- Profiling
  - In general shrinks the time dimension
  - Dynamic phase profiling [2]
- Tracing
  - Trace size linearly grows with time
  - cCCG [3]
  - Wavelet compression [4]
- Visualization
  - 2D, 2D heat maps [5], 3D, …
  - New visualization techniques needed [6]

Outline

- Performance Dynamics
- Periscope overview & limitations
- Introducing temporal dimension
- Automatic analysis of performance dynamics
- Examples
- Summary

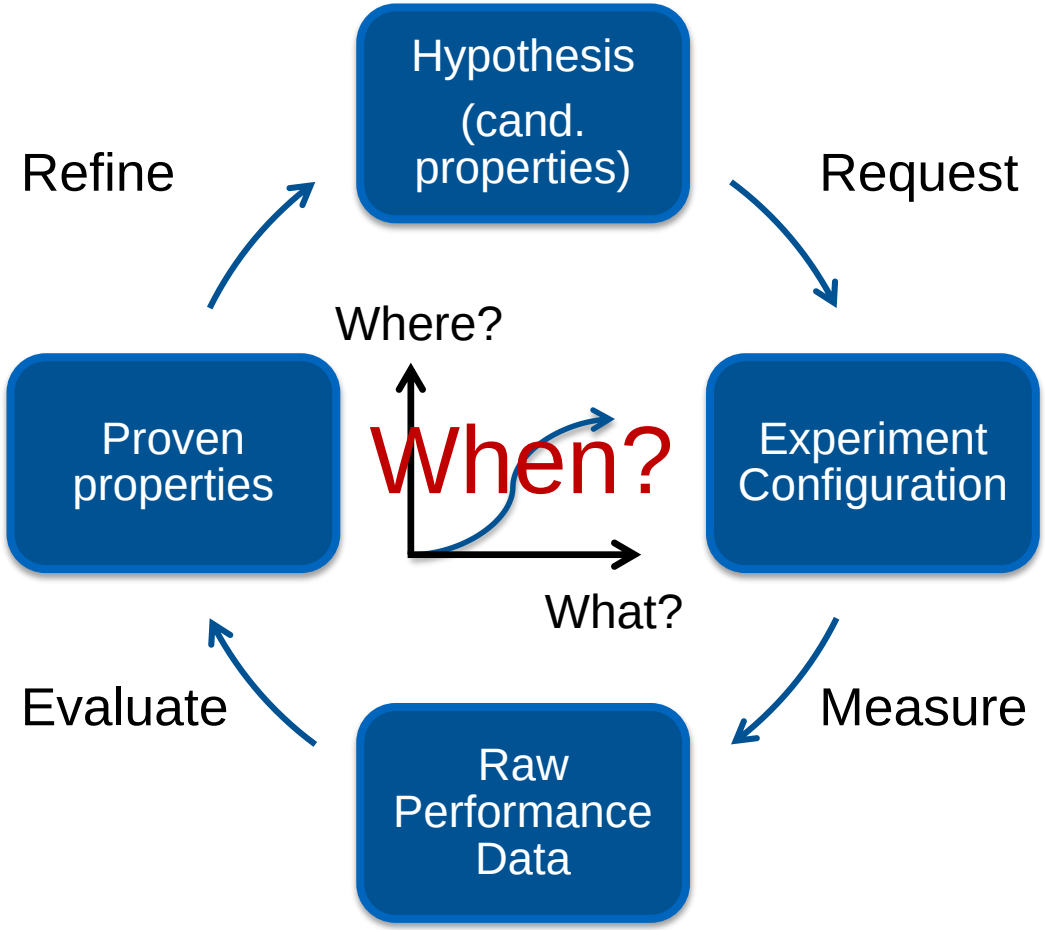
Periscope Overview

- Purpose
  - Automatically search and quantify bottlenecks
- Features
  - Profile-based measurements
  - Distributed architecture
  - Iterative online analysis
  - Automated search for predefined performance properties
  - Clustering of found properties
  - Close integration with Eclipse
- No visualization
  - High-level qualitative description of the problem instead

Analysis Automation

- Automation through formalization of typical inefficiencies
- Performance property
  - High-Level qualitative description
  - Condition
  - Severity
- Property examples:
  - Excessive time in MPI_Recv due to late send
  - Load Imbalance due to uneven sections in SECTIONS region

Periscope Analysis Model



Periscope limitations in respect to performance dynamics

- Assumes that performance is constant over time
  - May lead to wrong results when assumption doesn't hold
- Property specification disregards temporal dimension
- Analysis of temporal performance dynamics not possible

Outline

- Performance Dynamics
- Periscope overview & limitations
- Introducing temporal dimension
- Automatic analysis of performance dynamics
- Examples
- Summary

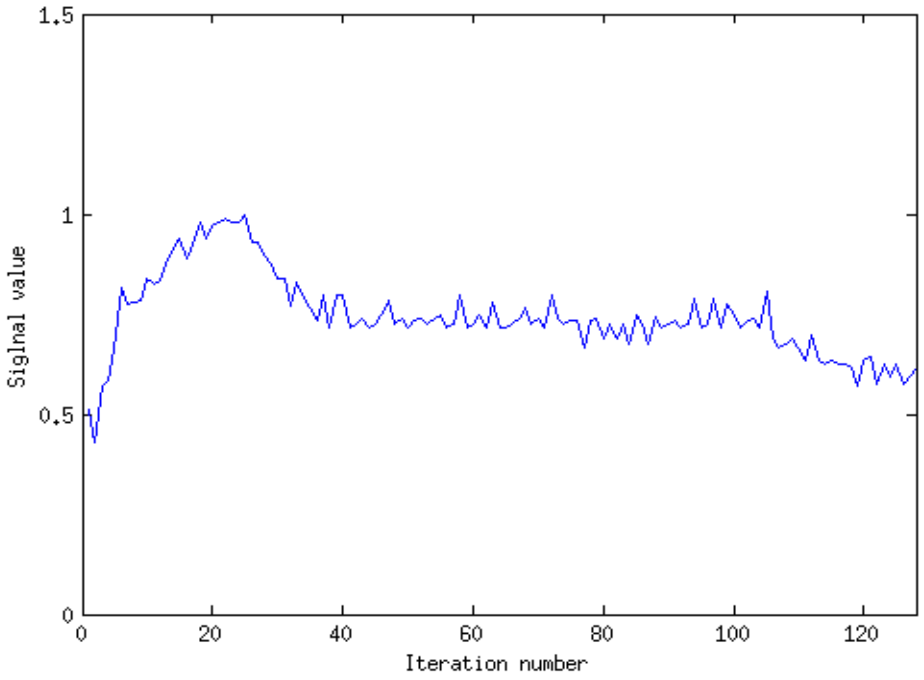
Dynamic Phase Profile

- Typically scientific applications perform simulations in iterations of a *progress loop*

- Idea: use progress loop iterations to sample the temporal dimension
  - assumes temporal evolution of performance

- $Dynamic\ Profile: (r \times p \times m \times i) \mapsto v,$

  r: application region
  p: execution location (process and thread)
  m: metric type
  i: progress loop iteration
  v: measured value



Scalability along the time dimension

- Profile size linearly grows with the time
- Solutions:
  - Online compression and post-mortem analysis
    - Clustering of dynamic profiles [5]

  or …

  - Online remote analysis

Online Dynamic Phase Profiling

- Idea:
  - Perform dynamic phase profiling
  - Transmit samples in online to the remote analysis agent

- Profile size and overheads on the application side are decoupled from time
- Extend monitoring library (Score-P) with online access interface
- Communication at the end of each iteration of the progress loop
  - Overheads are canceled out from the measurements
- Automatic online analysis by the remote Periscope agents

Outline

- Performance Dynamics
- Periscope overview & limitations
- Introducing temporal dimension
- Automatic analysis of performance dynamics
- Examples
- Summary

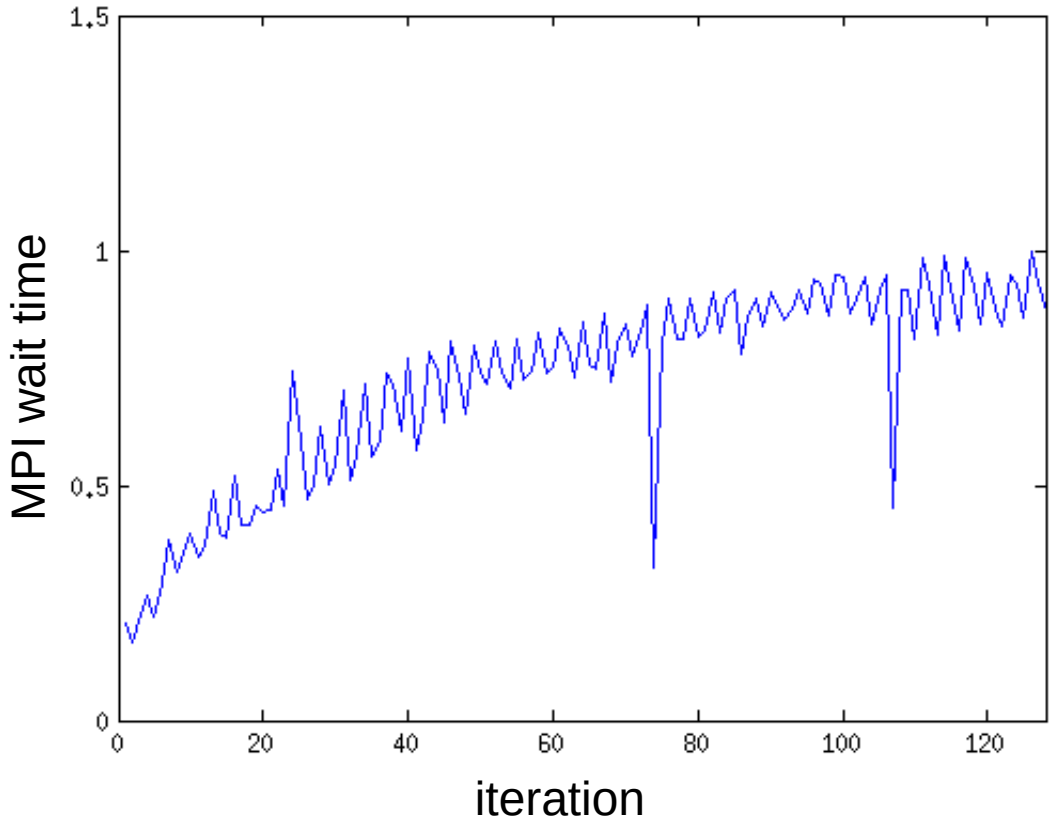
Performance dynamics analysis, you first ☺

- Real-world application: SPEC MPI2007 129.tera_tf
  - Run for 128 iterations on 32 cores
- 64 MPI imbalance locations (2 per process)

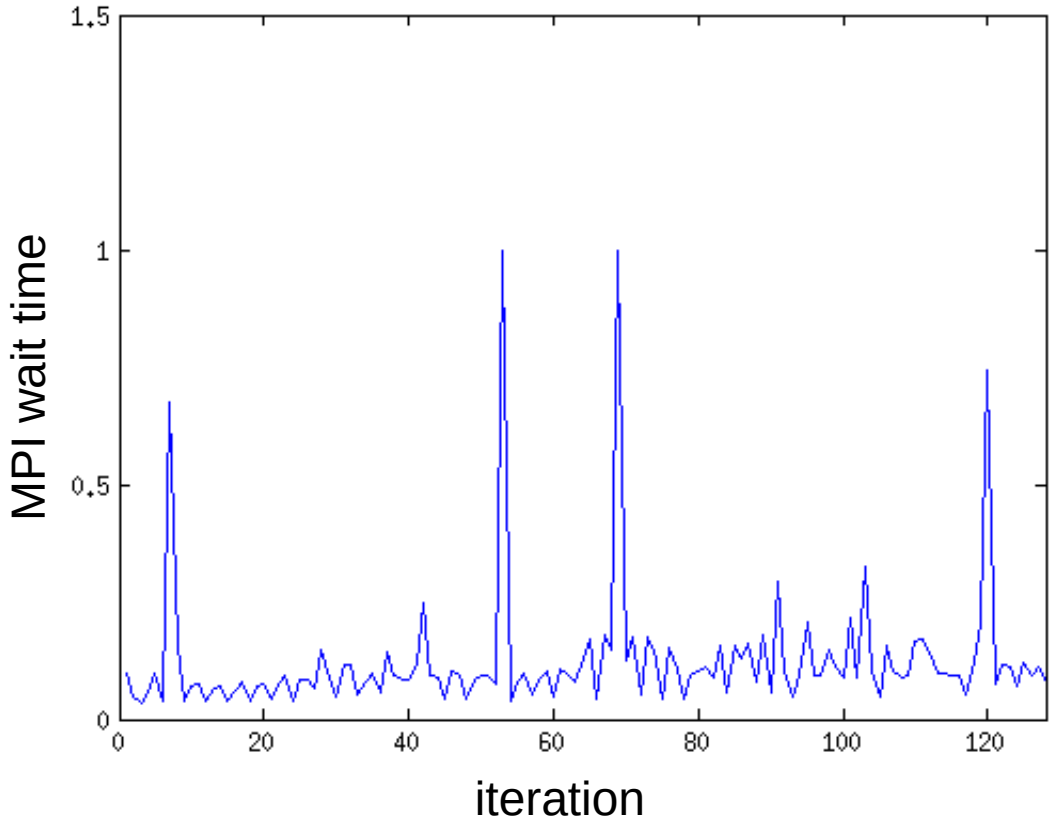
- 4 out of 64 plots of MPI imbalance vs iterations

- Having that, how would you **qualitatively** describe relevant dynamics?

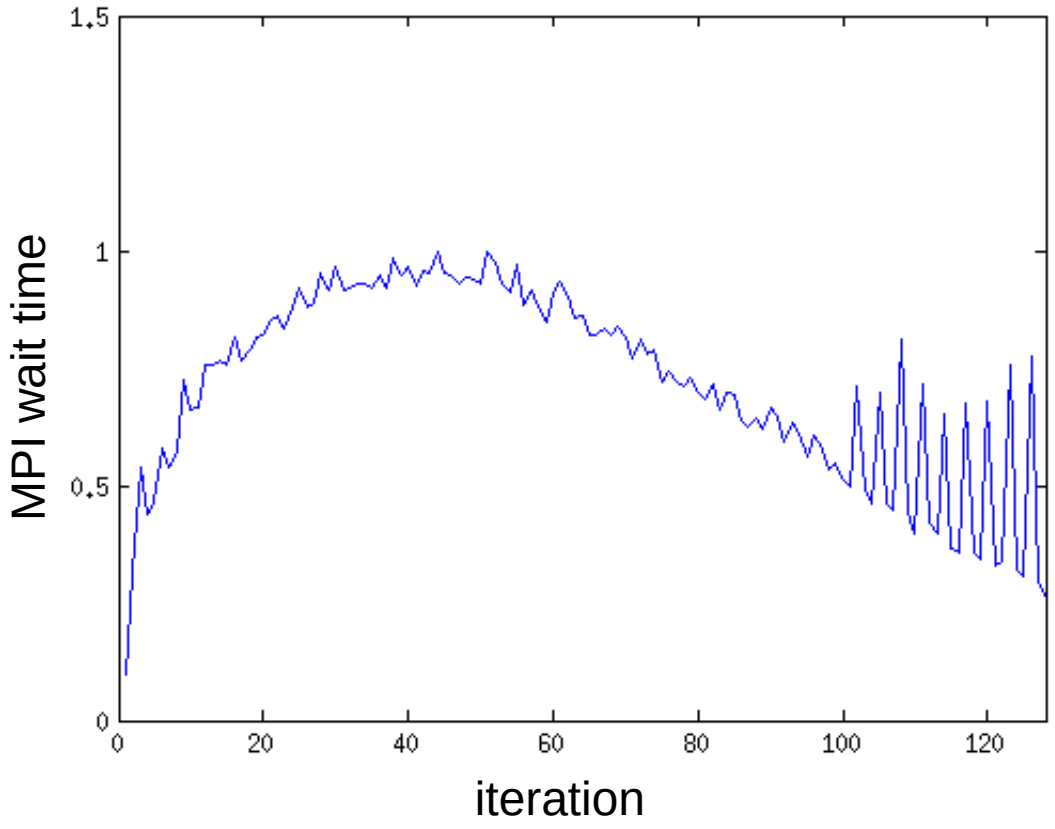
Example 1



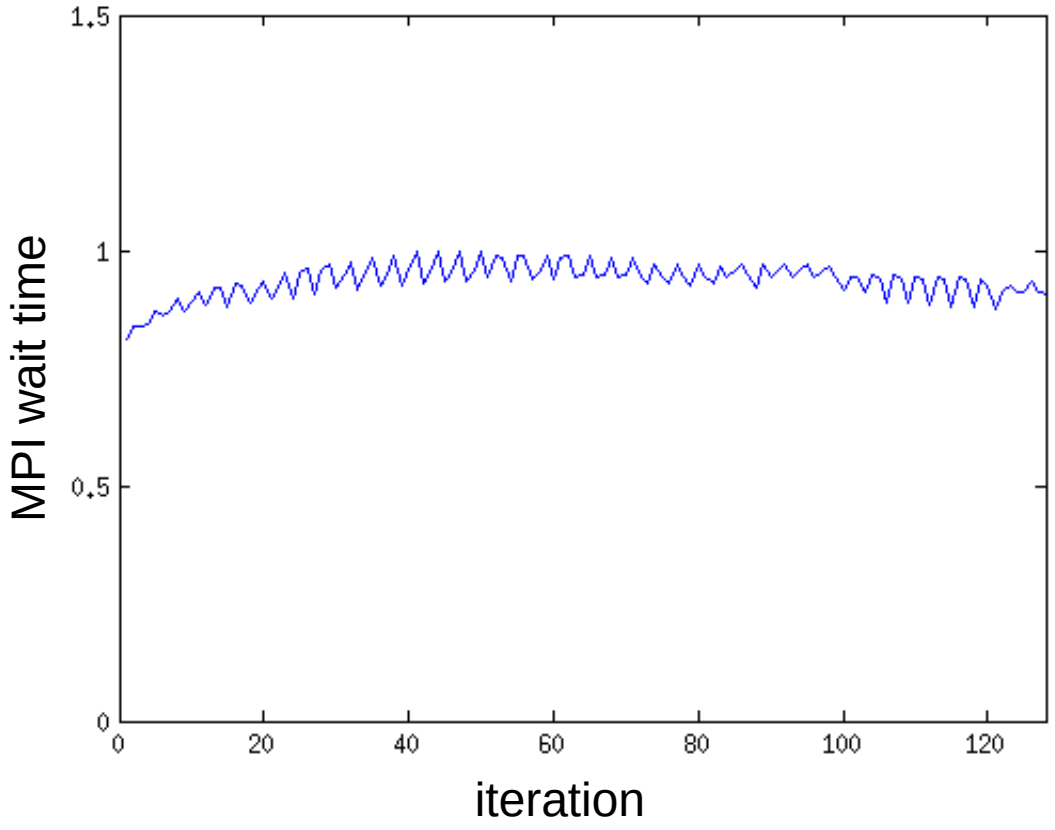
Example 2



Example 3



Example 4



Periscope performance dynamics analysis

- Analysis goal
  - Detect and characterize what you recognize as important **automatically**

- Detect relevant dynamics
- Quantify impact of performance dynamics
- Characterize dynamics qualitatively

- Get some help from signal processing world

Detect and quantify

Discrete Wavelet Transform (DWT)

- A natural framework for time-frequency (scale) analysis
- Algorithm
  - Pass signal through Band-pass filter -> Details (W) at scale *j*
  - Pass signal through Low-pass filter -> Approximation (L) at scale *j*
  - Downsample both approximation and details
  - Repeat recursively on Approximation coefficients of the prev. scale
- Complexity
  - $O(n)$
- Energy conservation property
  - $\sum_i \|X\|^2 = \sum_{j,n} \|W\|^2 + \|L\|^2,\ where\ j - scale, n\ - shift$

Qualitative Characterization

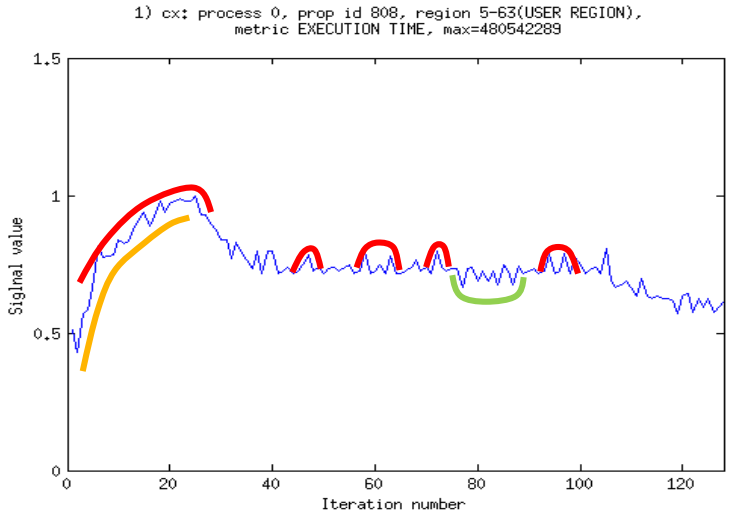
Scale-Space Filtering and Qualitative Summarization

- Systematic multi-scale analysis
  - Incremental smoothing
  - Detection of critical points
- Hierarchical qualitative representation

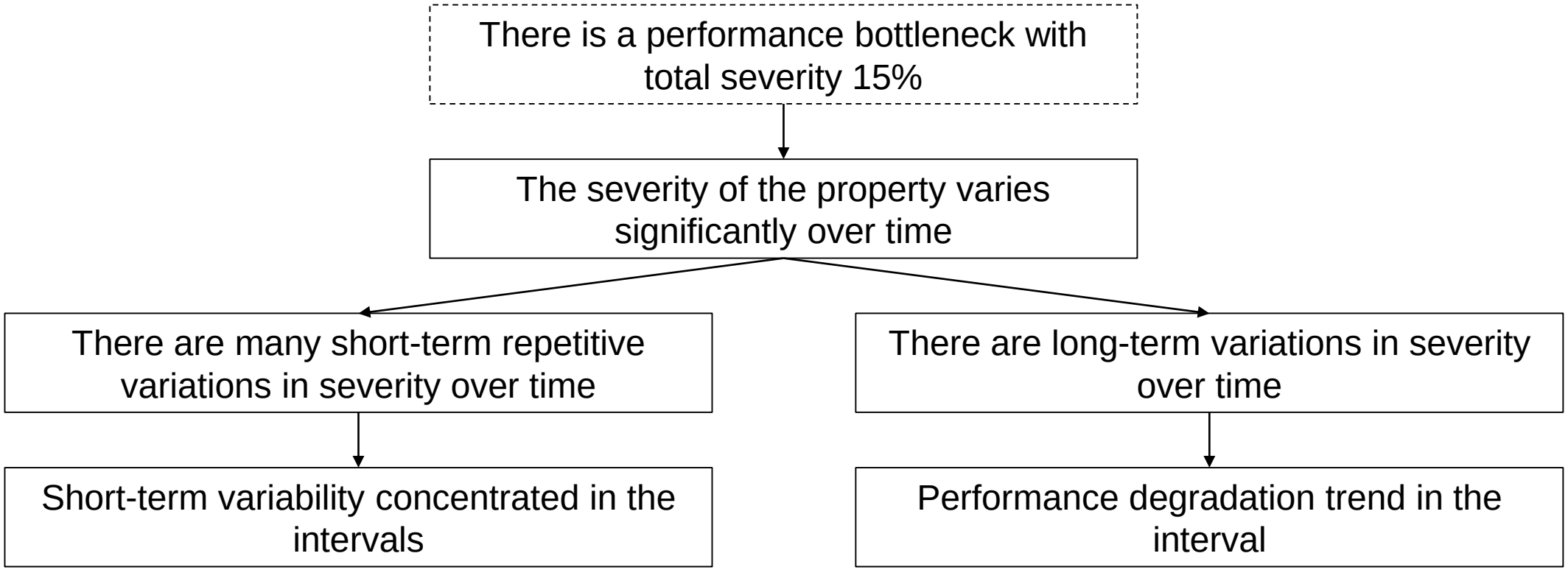
- Quantification of perceptual importance
- Complexity
  - $O(n)$

Detection of trends

- Find all peaks (AB): DABCBCDABCDABCDABCDABC
- Find the most “outstanding” valley (CD): DABCBCDABCDABCDABCDABC
- Find the largest increase (DA): DABCBCDABCDABCDABCDABC



Performance Dynamics Hypothesis Tree



Bringing it together: Analysis Strategy

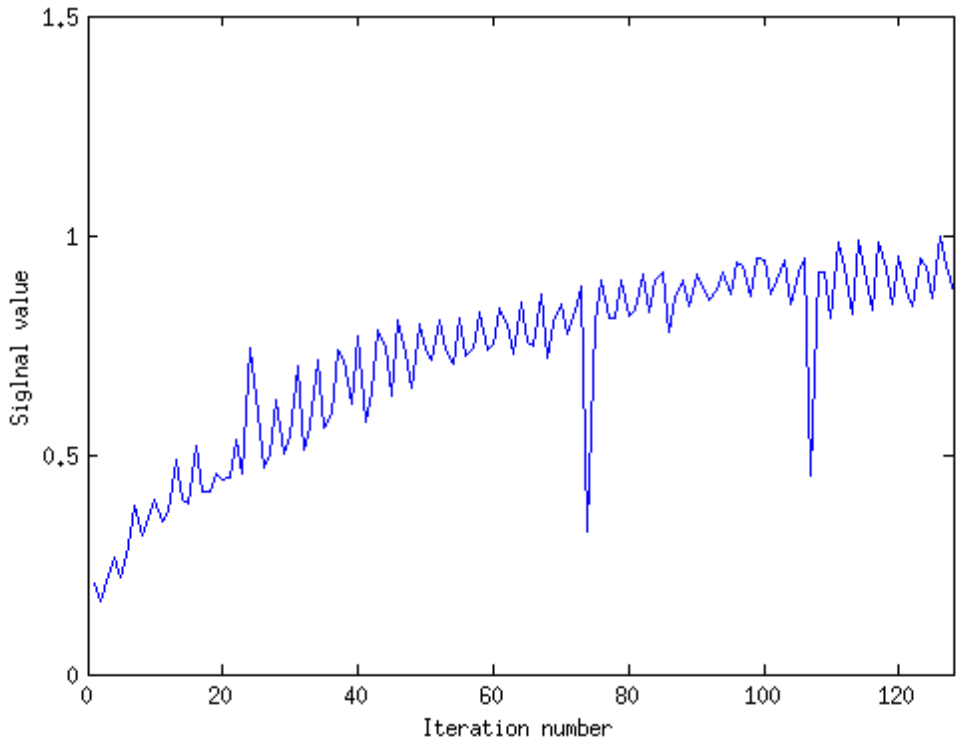
1. Online dynamic phase profiling of the N iter. of the progress loop
2. Apply a legacy static analysis strategy <*MPI, OpenMP, Single core*>
3. Re-evaluate found static properties in each iteration separately
4. Check dynamic properties on each time-series of severities over iterations
5. Repeat for the next N iterations
6. Group dynamic properties

Outline

- Performance Dynamics
- Periscope overview & limitations
- Introducing temporal dimension
- Automatic analysis of performance dynamics
- Examples
- Conclusion

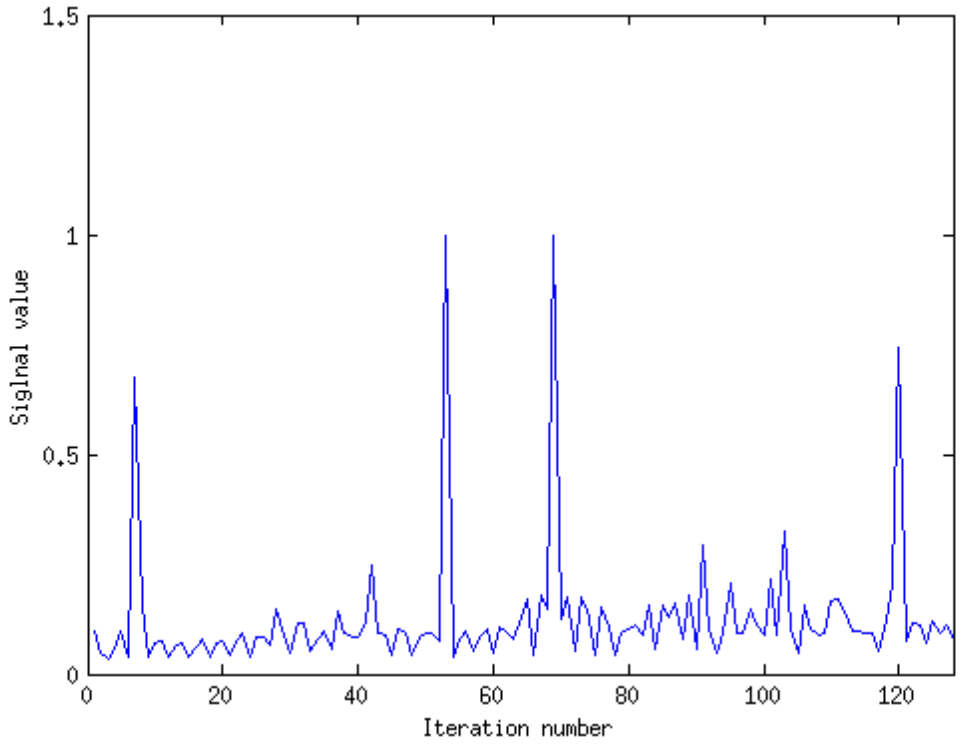
Periscope turn: Example 1

- Performance degradation trend in the interval [0,128]



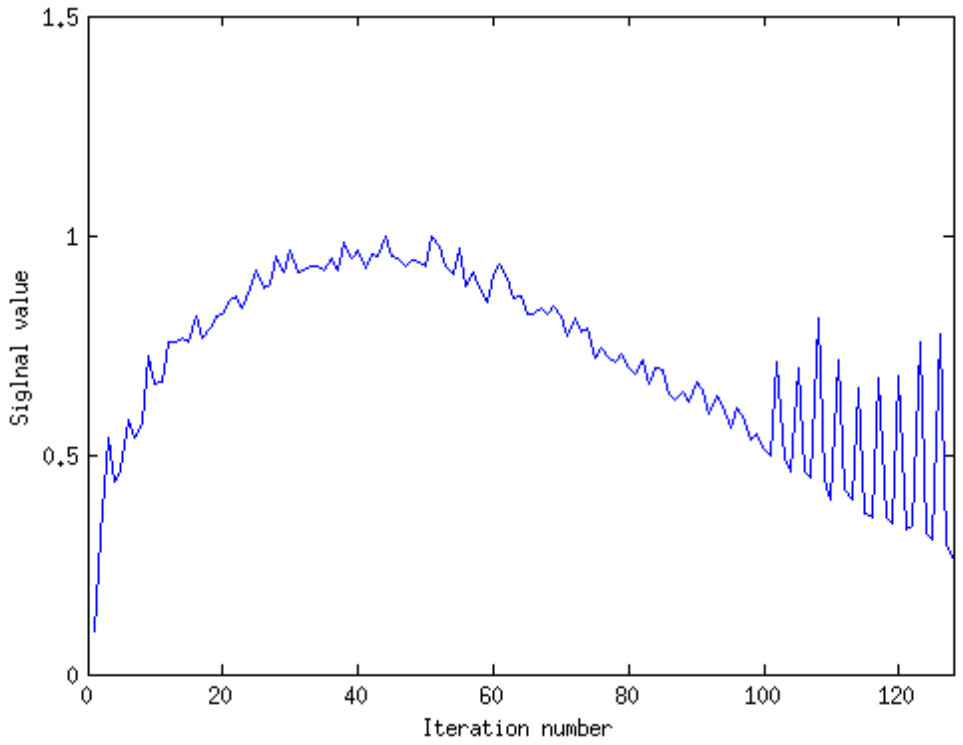
Periscope turn: Example 2

- Short-term variability concentrated in the intervals [4,7], [48,55], [68,71], [116,121]



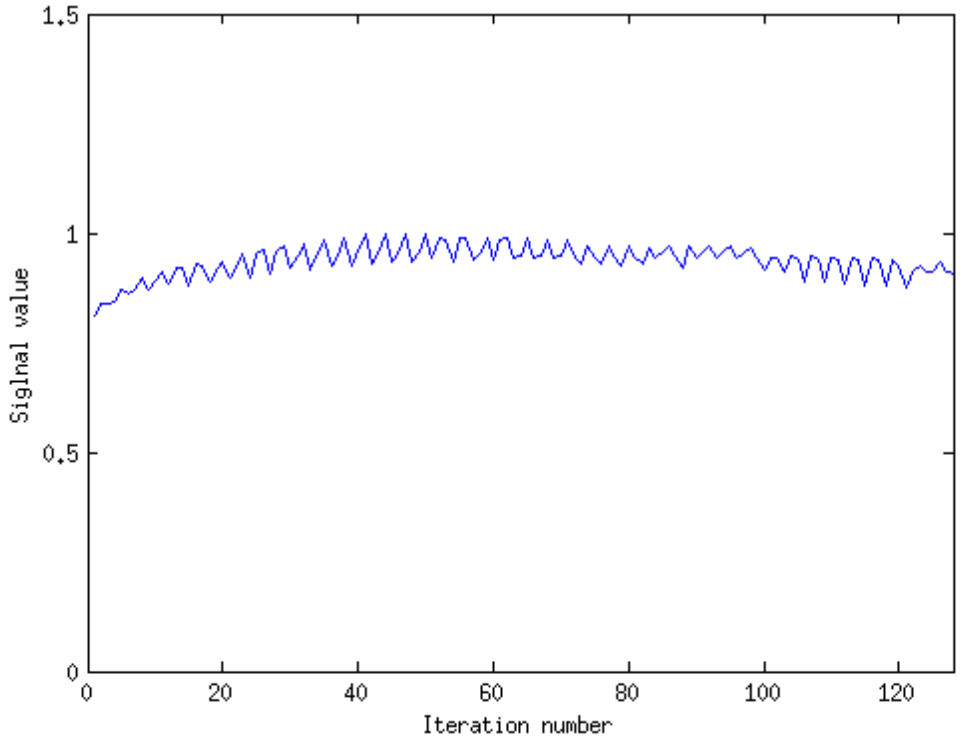
Periscope turn: Example 3

- Performance degradation trend in the interval [0,41]
- Short-term variability concentrated in intervals: [0,7] and [104,127]



Periscope turn: Example 4

- No dynamic behavior
- There is a repetitive short-term variability
- Performance degradation trend in the interval [0,60]



Summary

- Performance is **not always** static
- Manual investigation tedious
- First step towards **automatic analysis** of performance dynamics
- Online profile-based analysis
- Thanks to signal processing techniques
- Performance dynamics properties detail
  - Location
  - Severity
  - High-level qualitative description of the dynamics

References

1. Szebenyi, Zoltán, Brian JN Wylie, and Felix Wolf. "Scalasca parallel performance analyses of PEPC." In Euro-Par 2008 Workshops-Parallel Processing, pp. 305-314. Springer Berlin Heidelberg, 2009.
2. A. D. Malony, S. S. Shende, and A. Morris. Phase-Based Parallel Performance Profiling. In E. Zapata G.R. Joubert, W.E. Nagel, F.J. Peters, O. Plata, P. Tirado, editor, Proceedings of the International Conference ParCo 2005, volume 33, pages 203–210. John von Neumann Institute for Computing, Julich, NIC Series, 2006.
3. Knupfer, Andreas, and Wolfgang E. Nagel. "Construction and compression of complete call graphs for post-mortem program trace analysis." In Parallel Processing, 2005. ICPP 2005. International Conference on, pp. 165-172. IEEE, 2005.
4. Gamblin, Todd, Bronis R. De Supinski, Martin Schulz, Rob Fowler, and Daniel A. Reed. "Scalable load-balance measurement for SPMD codes." In High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for, pp. 1-12. IEEE, 2008.
5. Zoltán Szebenyi: Capturing Parallel Performance Dynamics. PhD thesis, RWTH Aachen University, volume 12 of IAS Series, Forschungszentrum Jülich, 2012, ISBN 978-3-89336-798-6.
6. Dagstuhl Perspectives Workshop 14022, “Connecting Performance Analysis and Visualization to Advance Extreme Scale Computing”, http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=14022

Thank you for your contribution and attention!

Questions?