Automatic Performance Analysis for Parallel Applications: Initial Ideas

Flavio Alles Rodrigues
Lucas Mello Schnorr
GPPD/UFRGS
{farodrigues, schnorr}@inf.ufrgs.br

Outline

Outline

[1] Motivation

Outline

[1] Motivation

[2] Automatic Performance Analysis

Outline

[1] Motivation

[2] Automatic Performance Analysis

[3] Survey of Tools

Outline

[1] Motivation

[2] Automatic Performance Analysis

[3] Survey of Tools

[4] Issues

Outline

[1] Motivation

[2] Automatic Performance Analysis

[3] Survey of Tools

[4] Issues

[5] Potential Approaches & Upcoming Work

Motivation

Motivation

Parallel computers are increasing in size...

Motivation

Parallel computers are increasing in size...

...becoming more heterogeneous...

Motivation

Parallel computers are increasing in size...

...becoming more heterogeneous...

and consuming more power.

Motivation

- Programming such a system is challenging

Motivation

- Programming such a system is challenging
- Performance analysis tools are essential

Automatic Performance Analysis

Automatic Performance Analysis

- Detect behavioral patterns and performance issues automatically

Automatic Performance Analysis

- Detect behavioral patterns and performance issues automatically
- Increases the amount of information that can be analysed

Automatic Performance Analysis

- Detect behavioral patterns and performance issues automatically
- Increases the amount of information that can be analysed
- Large-scale parallel applications generate huge amounts of data

Automatic Performance Analysis

- Detect behavioral patterns and performance issues automatically
- Increases the amount of information that can be analysed
- Large-scale parallel applications generate huge amounts of data
- Path to scale analysis to the required levels

Survey of Tools

Survey of Tools

- Automate some aspect of analysis

Survey of Tools

- Automate some aspect of analysis
- Focus is on parallel applications

Survey of Tools

- Automate some aspect of analysis
- Focus is on parallel applications
- Presented to public in peer-reviewed articles

Survey of Tools

- Paradyn [1]
- Periscope [2]
- Scalasca [3]

Paradyn

- Online, profile-based analysis

Paradyn

- Online, profile-based analysis
- W^3 Search Model

Paradyn

- Online, profile-based analysis
- W^3 Search Model
  - *Why* is the application performing poorly?

Paradyn

- Online, profile-based analysis
- W^3 Search Model
  - *Why* is the application performing poorly?
  - *Where* is the bottleneck?

Paradyn

- Online, profile-based analysis
- W^3 Search Model
  - *Why* is the application performing poorly?
  - *Where* is the bottleneck?
  - *When* does the problem occur?

Periscope

- Online, profile-based analysis

Periscope

- Online, profile-based analysis
- Distributed autonomous search for pre-defined bottlenecks (ASL [4])

Periscope

- Online, profile-based analysis
- Distributed autonomous search for pre-defined bottlenecks (ASL [4])
- Two distinct search strategies

Periscope

- Online, profile-based analysis
- Distributed autonomous search for pre-defined bottlenecks (ASL [4])
- Two distinct search strategies
  - Single-node Performance (*i.e.* stalled cycles)

Periscope

- Online, profile-based analysis
- Distributed autonomous search for pre-defined bottlenecks (ASL [4])
- Two distinct search strategies
  - Single-node Performance (*i.e.* stalled cycles)
  - MPI Performance (*i.e. load imbalances*)

Scalasca

- Offline, trace-based analysis

Scalasca

- Offline, trace-based analysis
- Parallel trace analysis: communications replay to search for pre-defined performance properties

Scalasca

- Offline, trace-based analysis
- Parallel trace analysis: communications replay to search for pre-defined performance properties
  - Wait states (forward replay) and its root causes (backwards replay)

Issues

Issues

- Static expectations of automatic analysis

Issues

- Static expectations of automatic analysis
  - Problems that automatic analysis tools diagnose

Issues

- Static expectations of automatic analysis
  - Problems that automatic analysis tools diagnose
  - Assumption that these tools make regarding the machines where the applications are executed

Potential Approaches

Potential Approaches

- Machine Learning

Potential Approaches

- Machine Learning
  - Unsupervised learning techniques

Potential Approaches

- Machine Learning
  - Unsupervised learning techniques
    - Clustering
    - Hidden Markov Models
    - Artificial Neural Networks
      - Self Organizing Maps
      - Adaptive Resonance Theory

Upcoming Work

- Python
- scikit-learn [5]
- Pajé Traces [6]

References

[1] Miller, Barton P., et al. "The Paradyn parallel performance measurement tool." *Computer* 28.11 (1995): 37-46.

[2] Benedict, Shajulin, Ventsislav Petkov, and Michael Gerndt. "PERISCOPE: An online-based distributed performance analysis tool." *Tools for High Performance Computing 2009*. Springer Berlin Heidelberg, 2010. 1-16.

[3] Geimer, Markus, et al. "The Scalasca performance toolset architecture." *Concurrency and Computation: Practice and Experience* 22.6 (2010): 702-719.

[4] Gerndt, Michael, and Karl Fürlinger. "Specification and detection of performance problems with ASL." *Concurrency and Computation: Practice and Experience* 19.11 (2007): 1451-1464.

[5] http://scikit-learn.org

[6] http://paje.sourceforge.net/download/publication/lang-paje.pdf

Automatic Performance Analysis for Parallel Applications: Initial Ideas

Flavio Alles Rodrigues
Lucas Mello Schnorr
GPPD/UFRGS
{farodrigues, schnorr}@inf.ufrgs.br