Improving Peer-to-Peer Resource Usage Through Idle Cycle Prediction

Elton Nicoletti Mathias, Andrea Schwertner Charão, Marcelo Pasin

LSC - Laboratório de Sistemas de Computação
UFSM - Universidade Federal de Santa Maria

Table Of Contents

- Goals
- Motivation
- Related Work
- The Idleness Detection Model
- Preliminary Results
- Further Benefits
- Conclusion and Future Work

Goals

- Dynamic mechanism to exploit idle resources
- Efficient use of resources
  - take the most of the resources
  - low disturbance

Motivation

- Large fraction of workstations are unused for a large fraction of time
- The computational power of these workstation is increasing steadily
- By the exploitation of such idle resources, a great computational power can be gathered for parallel processing
- ProActive offers a peer-to-peer infrastructure that can take profit of idle CPU cycles, but their availability must be expressed statically in configuration files.

Motivation

- Large fraction of workstations are unused for a large fraction of time
- The computational power of these workstation is increasing steadily
- By the exploitation of such idle resources, a great computational power can be gathered for parallel processing
- ProActive offers a peer-to-peer infrastructure that can take profit of idle CPU cycles, but their availability must be expressed statically in configuration files.

Motivation

- Large fraction of workstations are unused for a large fraction of time
- The computational power of these workstation is increasing steadily
- By the exploitation of such idle resources, a great computational power can be gathered for parallel processing
- ProActive offers a peer-to-peer infrastructure that can take profit of idle CPU cycles, but their availability must be expressed statically in configuration files.

Motivation

- Large fraction of workstations are unused for a large fraction of time
- The computational power of these workstation is increasing steadily
- By the exploitation of such idle resources, a great computational power can be gathered for parallel processing
- ProActive offers a peer-to-peer infrastructure that can take profit of idle CPU cycles, but their availability must be expressed statically in configuration files.

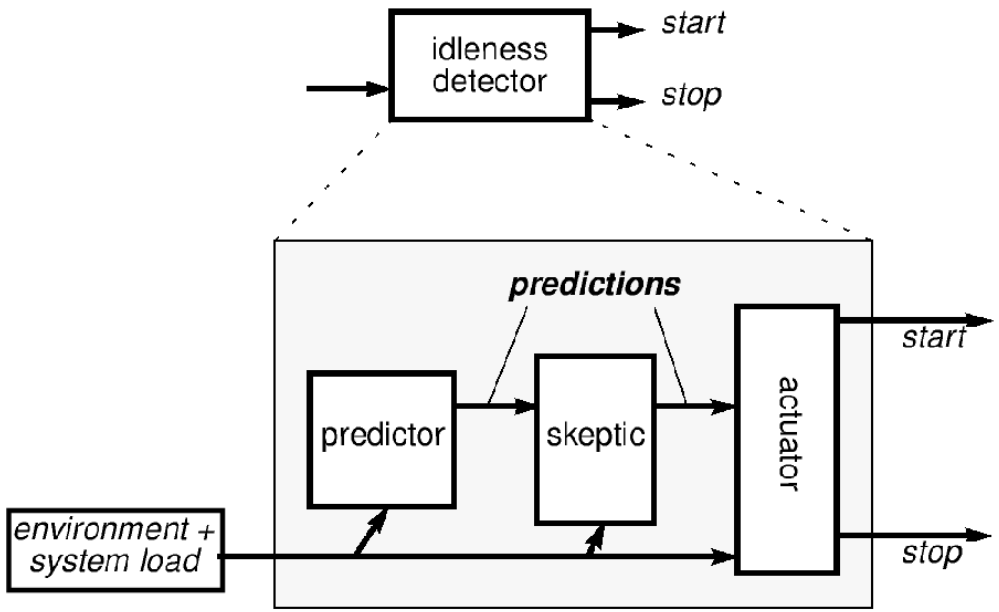
Related Work

- Seti@home (BOINC)
  - Static Prediction
  - Based on keyboard and mouse interruptions
  - Completely independent tasks
- Condor (and XTremWeb)
  - Adaptive prediction
  - Based on CPU utilization and load averages
  - Fully integrated on job scheduler
  - Related to job deadlines
  - Batch processing approach

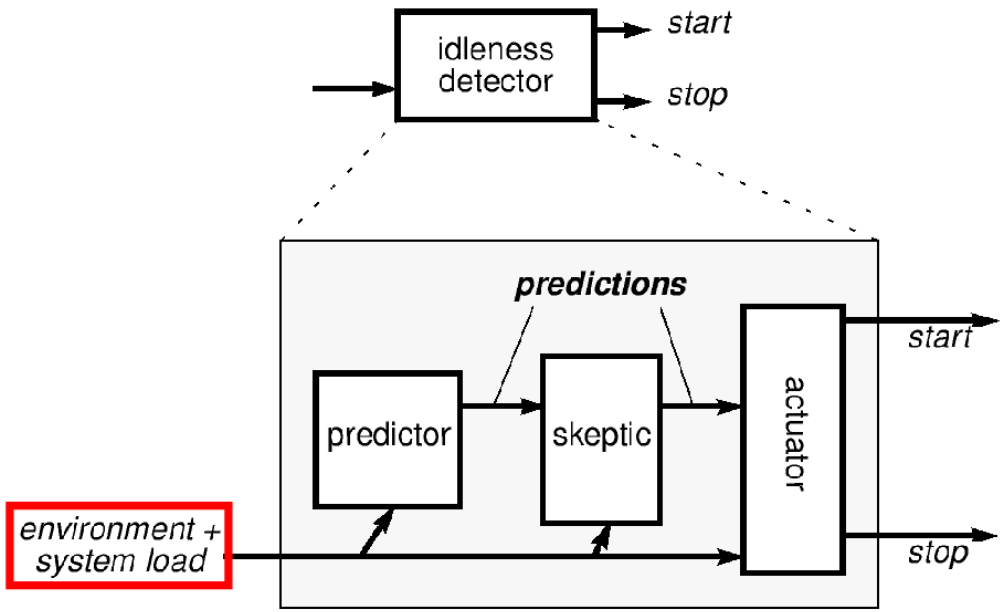
Related Work

- Seti@home (BOINC)
  - Static Prediction
  - Based on keyboard and mouse interruptions
  - Completely independent tasks
- Condor (and XTremWeb)
  - Adaptive prediction
  - Based on CPU utilization and load averages
  - Fully integrated on job scheduler
  - Related to job deadlines
  - Batch processing approach

Proposed Model



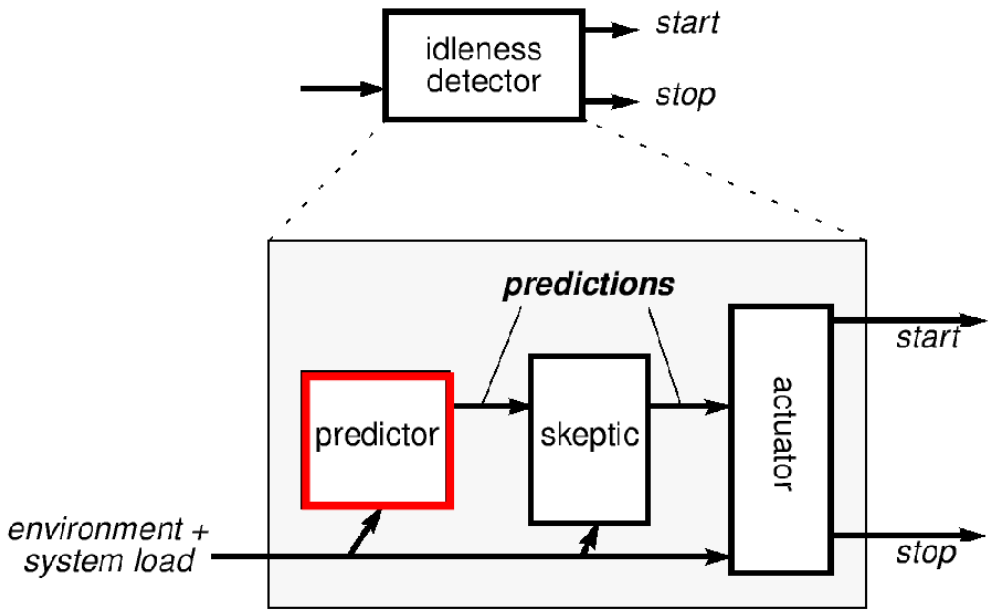
Metric Collection



Metric Collection

- Use of native library
  - Keep Java portability
  - Low Overhead
- Currently Suported:
  - OSs: Linux, Solaris, FreeBSD, AIX, IRIX, HPUX, MacOS X and Windows NT/XP/2000
  - archs: i386, ia64, sparc, powerpc, s390
- Interfaced with common-use tools, such as:
  - Ganglia, Performance Co-Pilot, Parmon and SCMS (SNMP support is also being implemented)

Predicting Values



Predicting Load Values

Important concepts:

- Interval of idleness
- Horizon of prediction

Approach Used:

- Larger horizon: Usage Pattern Analysis through Wavelet Compression
- Smaller horizon: Time Series Models (Moving Averages, ...)

Predicting Load Values

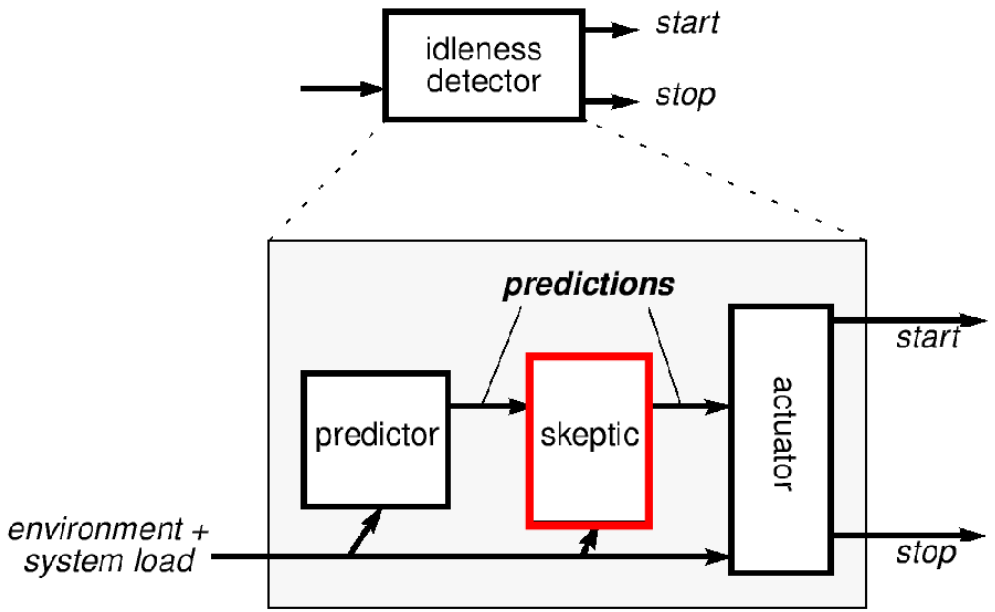
Important concepts:

- Interval of idleness
- Horizon of prediction

Approach Used:

- Larger horizon: Usage Pattern Analysis through Wavelet Compression
- Smaller horizon: Time Series Models (Moving Averages, ...)

Adjusting Predictions



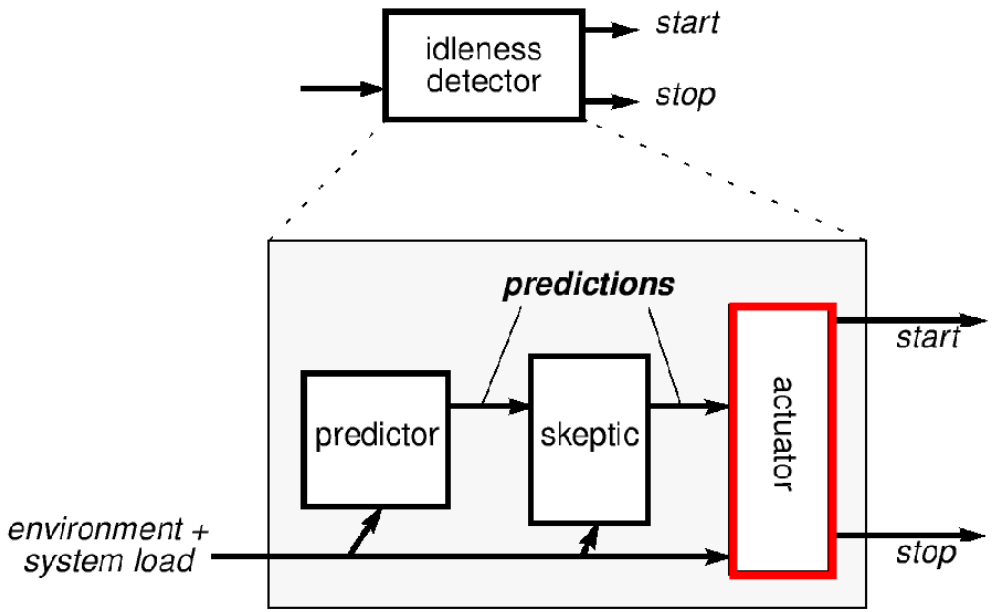
Adjusting Predictions

- Accuracy of predictions depends on environment behaviour
- Past prediction analysis can improve the future ones

How:

- Threshold adjusts
- Prediction algorithm adjusts

Integration with ProActive's P2P



Integration with ProActive's P2P

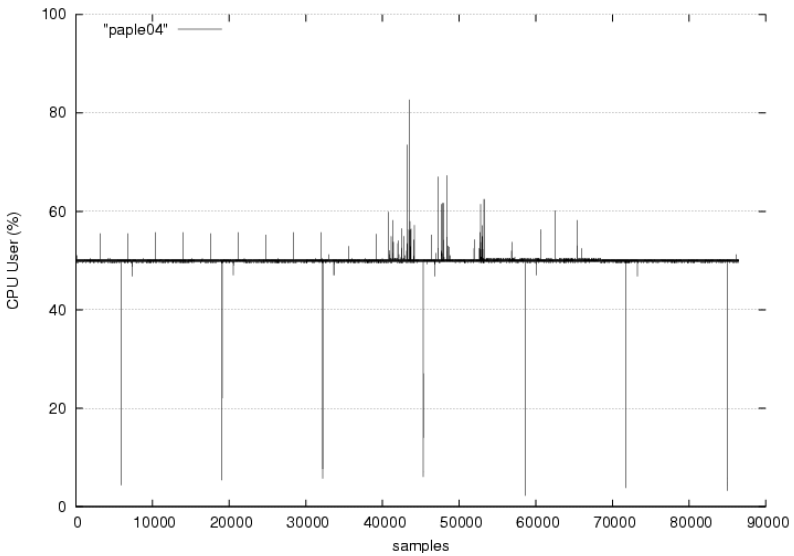
- Availability of P2P Service daemon guided by idleness detector instead of xml file
- Control of keyboard and mouse interruptions on workstations
- So far, no changes on the P2P protocol

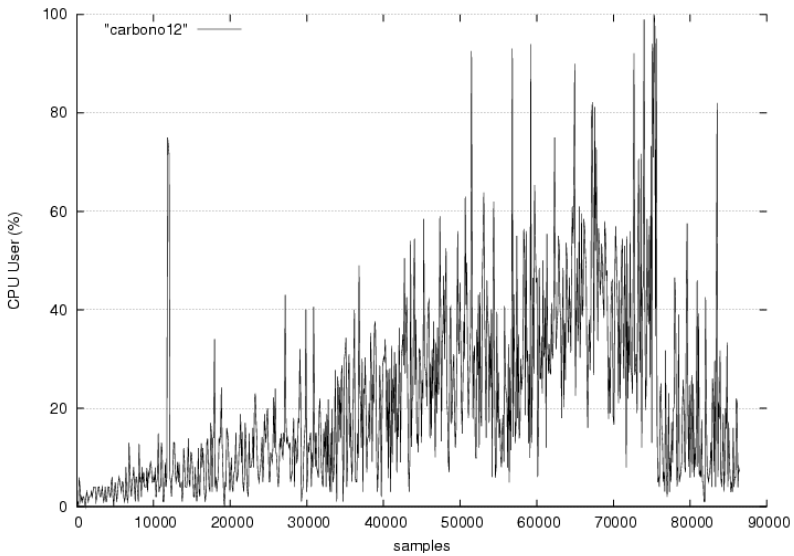
Prediction Accuracy

- Collected values: CPU usage on 30 nodes
- Nodes with different usages: dedicated cluster machines, a frontend and users' workstations
- Interval of reading: 10 seconds
- Hit rates table:

Steps ahead	Wavelet Compression	Time Series	Both
01	97.5 %	98.3%	98.5 %
10	90.8 %	92.1%	93.5 %
30	85.7 %	82.0%	86.3 %

Prediction Accuracy





Steps ahead	Wavelet Compression	Time Series	Both
01	99.5 %	99.4%	99,6 %
10	97,2 %	96.8%	98,5 %
30	96,5 %	92,1%	97,9 %

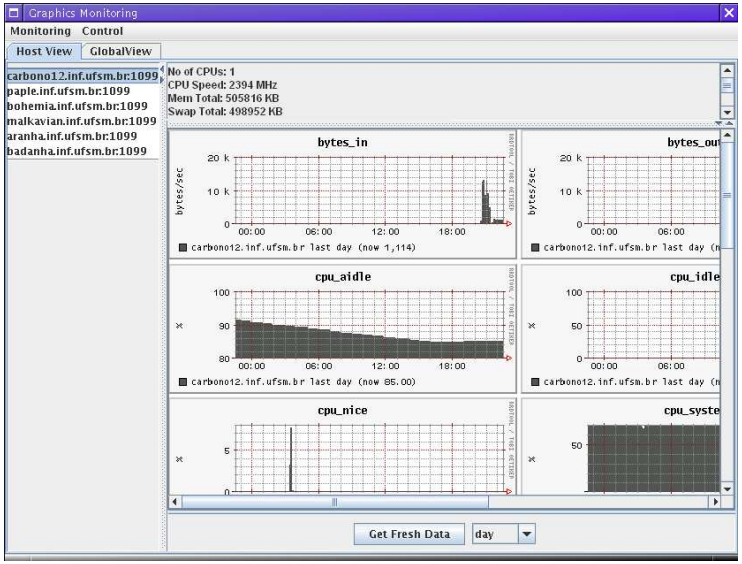
Steps ahead	Wavelet Compression	Time Series	Both
01	84.4 %	92.2%	96,6 %
10	68,8 %	74.1%	78,2 %
30	46,1 %	66,9%	67,5 %

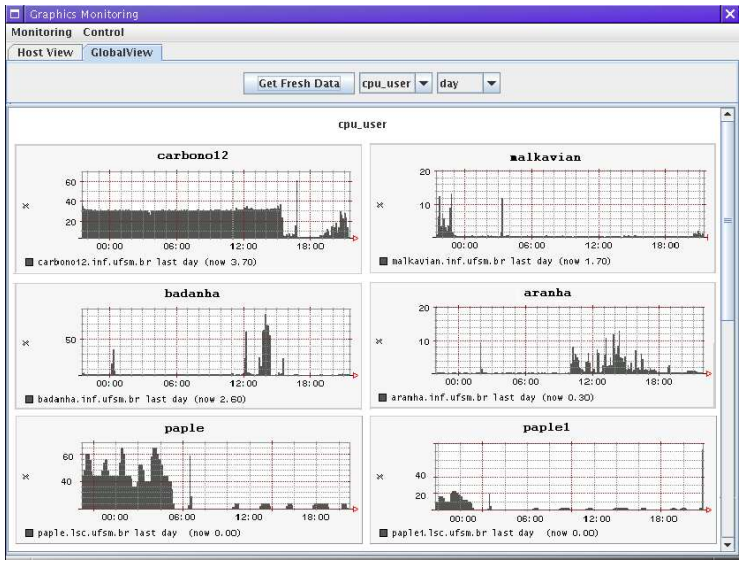
Integration with ProActive

- Intervals of idleness
  - Shorter than 3-5 min: system can become unstable
  - Longer than 15 min: not advisable for non-dedicated environments
- Horizon of prediction
  - Shorter are strong co-related. Longer, not too much
- More tests are required for evaluating the model...

Further Benefits

- Extension to the IC2D
  - Graphical visualization of load history (CPU, memory, network usage, load average, ...)





Further Benefits

- Package for obtaining on demand informations about hardware and machines' load indexes
  - Portability
  - Low Overhead

Conclusions

- It is possible to roughly ”predict” the resource usage for the future based on the past;
- Non-dedicated environments, such as workstations connected by ordinary networks, can offer a great computational power for parallel processing;
- Usage of resources can be improved through the constant use of their idle times;
- It is possible to keep P2P self-organization, by adjusting the horizon of idleness detection

Future Work

- Evaluate the behaviour of the model for larger environments
- Analyse finer grain usage (smaller horizons)
- Measure the impact of addition or subtraction of workstations on applications and look for ways to minimize it
- Asses the worthiness of using non-dedicated environments for parallel processing

References

- Matt W. Mutka, Estimating Capacity for Sharing in a Privately Owned Workstation Environment, IEEE Transactions on Software Engineering, v.18 n.4, p.319-328, April 1992
- A. Acharya, G. Edjlali, and J. Saltz. The utility of exploiting idle workstations for parallel computation. In Proceedings of SIGMETRICS'97, 1997
- R. Golding, P. Bosch, C. Staelin, T. Sullivan, and J. Wilkes. Idleness is not sloth. In Proc. of the USENIX Conf., pages 201–212, New Orleans, LA, Jan. 1995
- M. Litzkow, M. Livny, and M. Mutka. Condor - A Hunter of Idle Workstations. In Proceedings of the 8th International Conference of Distributed Computing Systems, 1988.
- G. Fedak, C. Germain, V. N'eri, and F. Cappello. XtremWeb: A Generic Global Computing System. In IEEE Int. Symp. on Cluster Computing and the Grid, 2001
- David P. Anderson , Jeff Cobb , Eric Korpela , Matt Lebofsky , Dan Werthimer, SETI@home: an experiment in public-resource computing, Communications of the ACM, v.45 n.11, p.56-61, November 2002
- C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. "Spawn: A Distributed Computational Economy,"IEEE Transactions on Software Engineering, February 1992.

Improving Peer-to-Peer Resource Usage Through Idle Cycle Prediction

Elton Nicoletti Mathias, Andrea Schwertner Charão, Marcelo Pasin

LSC - Laboratório de Sistemas de Computação
UFSM - Universidade Federal de Santa Maria