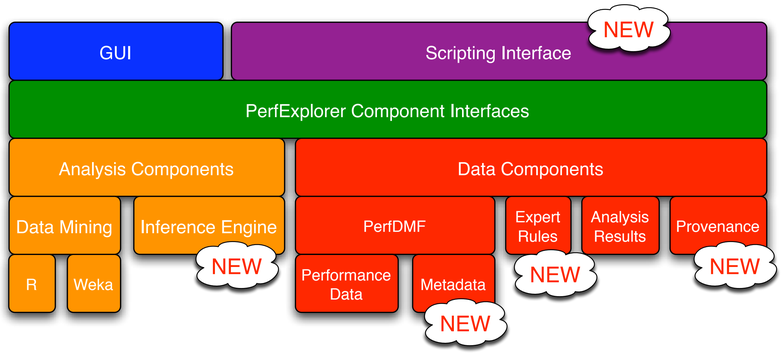
Automated Performance Data Mining with Knowledge Support in PerfExplorer2

Kevin A. Huck

khuck@cs.uoregon.edu - 1201 University of Oregon, Eugene OR 97403 - http://www.cs.uoregon.edu/research/tau/

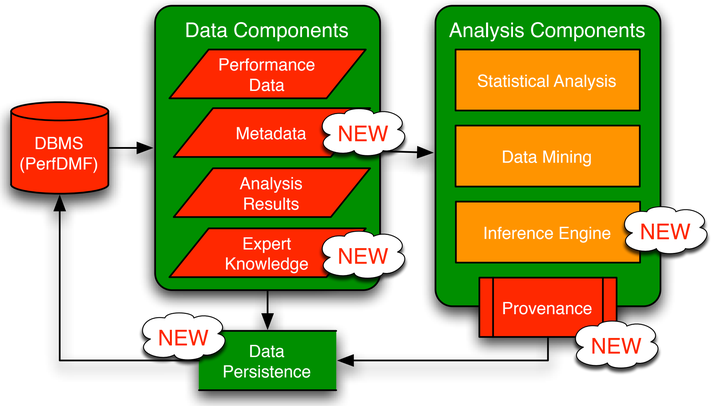
UNIVERSITY OF OREGON

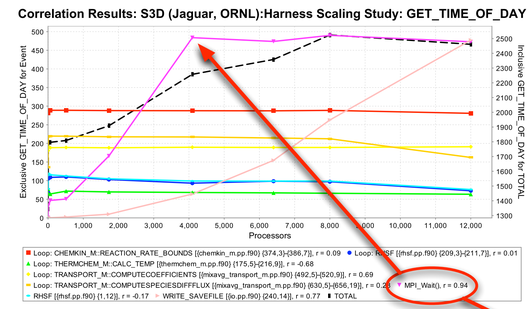
Tuning and Analysis Utilities

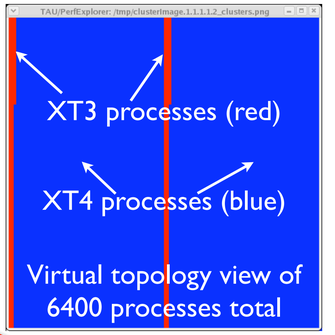


New Design: PerfExplorer2 is a redesigned toolkit for analyzing large parallel performance profiles and for performing comparisons among multiple performance profiles. The new design is component based, with well defined interfaces between layers. External tools are integrated where possible. New components include: a *scripting interface* for automation, *data persistence* for provenance and intermediate results, and an *inference engine* programmed with *expert rules* in order to explain performance results with respect to newly integrated *metadata* values.

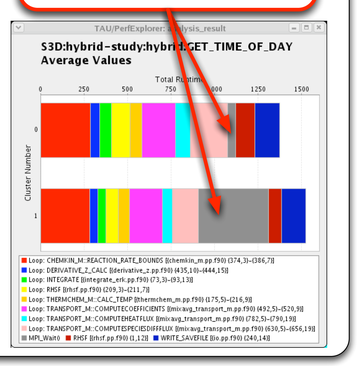
Component Interaction: Analysis processes are defined and controlled through Jython scripting. The scripting is used to input *data components* into *analysis components*, which return new analysis results which can be used as input data components for further analysis. The full *provenance* is stored for each analysis process, including input and output data, and all analysis configuration choices.



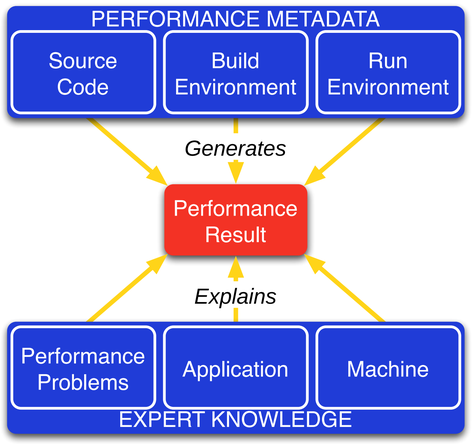




MPI_Wait() is the primary difference between the two clusters of processes, and dominates total execution

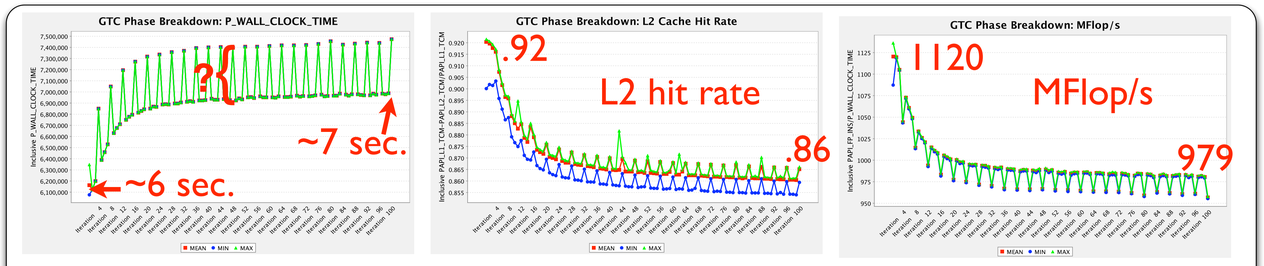


S3D Application Example: Analysis showed poor scalability due to increased time in a communication routine, MPI_Wait. This increased time is primarily due to a load balance issue caused by the hybrid Cray XT3/XT4 architecture. Cluster analysis of a 6400 process run identified two natural clusters of processes, and metadata analysis found that the faster XT4 nodes are waiting on the handful of slower XT3 nodes.



Metadata and Expert Knowledge: One of the key components of the new design is the integration of performance *metadata* and analysis *expert knowledge*. The metadata is collected at build and run time, and is stored with the performance results. Rather than analyzing the performance results alone, the performance results are considered as dependent on the metadata. Expert knowledge is used to explain the analysis results, rather than just providing summaries of the performance data. While this work is in its initial stages, results are promising.

This research is sponsored by contracts (DE-FG02-07ER25826 and DE-FG02-05ER25680) from the MICS program of the U.S. Department of Energy's Office of Science and from the National Science Foundation (grant #CCF0444475). Thanks to Prof. Allen D. Malony and the Performance Research Laboratory team members, particularly Sameer Shende and Alan Morris.



GTC Application Example: The application was instrumented with TAU, with extra instrumentation to capture dynamic phase measurements around the main iteration loop. The data shown is one execution on 64 processors of a Cray XT4/XT3 for 100 iterations. Each successive iteration takes more longer than the previous iteration. In addition, there is a spike in execution time every 4th iteration. The overall increase in execution time is primarily due to a decrease in the L2 hit rate, as memory access become less regular over time. This increase in execution time results in a decrease in the overall MFlop/s rate. The spikes are due to a diagnostic step every 4th iteration (runtime configurable) identified and captured as metadata.