Performance Databases and Multivariate Statistical Analysis

Shirley Moore
shirley@cs.utk.edu

Scalability and Interoperability Issues
- Need to organize and archive large amounts of performance data
- Need to preserve "metadata" – e.g., machine and compiler configuration, test case, application info
  - Important for repeatability
- Need for automated analysis and correlation of large amounts of data for different metrics and from different sources and tools
- Need for multi-experiment analyses

TAU Performance Database Management Framework (PerfDMF)
- Performance analysis programs
- Performance analysis and query toolkit
- Profile data only
- XML representation
- Project / experiment / trial

Other tools
MySQL Oracle

ParaProf Architecture
- Portable, extensible, scalable tool for profile analysis
- Upload performance data from file system, database, and/or at runtime

Visualizations & analysis
Feedback
- Multiple code versions
- Feedback
- Performance modeling and optimization strategies
- Creation of new optimization strategies
- Scalability analysis

Event traces
Automatic transformation of event traces into high-level profile
SvPablo
CUBE
ParaProf
Vampir
Cray X1 is the fastest to solution in all 3 tests.
FFT (nl2) improves time for B3-gtc only.
TeraGrid (peach) faster than p690 (red/yellow) for B1-std?

1. By experiment, total runtime: IBM p690 (red) has strange dip?
2. By event for one experiment: Coll_tr (blue) is a significant cause.
3. By experiment for one event: shows how Coll_tr behaves for all experiments.
1. By experiment, total runtime: X1 (blue) has smallest speedup
2. By event for one experiment: NL_tr (orange) is significant
3. By experiment, one event: shows how NL_tr behaves for all experiments

IBM SP3 has the highest fraction of time spent in communication for all three benchmarks
Cray X1 is lowest

Both Coll_tr (blue) and NL_tr (orange) scale poorly...
I/O (green) is insignificant

Breakdown by phase shows variability from beginning of application to final solution
- Relative Efficiency
- Runtime Breakdown
Iteration 6 (cyan) has big drop in efficiency for 128 processors
Greater variability at higher processor counts

Focus attention on important metrics and show the distribution of those metrics across parallel tasks and code regions
Statistical reductions can be used as filters to reduce the total amount of performance data, either at the time the data are generated or at analysis time.
Principal Components Analysis

- More than one variable may be measuring the same driving principle.
- Often there are only a few driving forces for dozens of system variables.
- Principal Components Analysis (PCA) generates a new set of variables, called principal components.
  - Each principal component is a linear combination of the original variables.
  - Principal components are orthogonal so that there is no redundant information.

Cluster Analysis

- Creates groups of objects, called clusters, or equivalence classes, such that objects in the same cluster are similar and objects in different clusters are distinct.
- Hierarchical (e.g., dendogram) and non-hierarchical (e.g., K-means clustering) methods

Combined PCA and Cluster Analysis of GYRO Profile Data

```
>> R = buildMatrix( data, 11 );
>> [counters, newdataset, percent_explained] = findPrincipalCounters( R , 3 );
>> [indices, q] = findClusters( newdataset , 2 );
>> plotClusters( newdataset , indices )
>> xlabel('FPU0 produced a result');
>> ylabel('Processor cycles');
>> zlabel('FPU executed FDIV instruction');
```