Understanding the Performance of Ultrascale Systems with Performance Counters

Jeffrey Vetter

Future Technologies Team Leader

Oak Ridge National Laboratory

Presented to

SIAM PP MS32

26 February 2004

San Francisco, CA, USA

Highlights

Massively parallel computing is here

Traditional techniques for performance analysis of hardware counter activity do not scale very well

- Many levels of concurrency

Microprocessor performance counters provide rich information about application performance

- How do we interpret all that performance counter data?

We propose a new solution

- Apply multivariate statistical methods

Our experiments reveal evidence that these techniques allow

- Easier understanding of counter metrics
- Reduce data management problems

Ultrascale Platforms will Rely on Extreme Concurrency

Over the past 15 years, supercomputing has moved toward massively parallel systems

- Scalar processors
- Vector processors
- SoC

This trend will continue

- Red Storm
- BlueGene/L
- Cray X2
- Commodity clusters

Cray X1/X2:
Leadership class computer for science

IBM Power4:
8th in the world

IBM Power3:
DOE-SC's first terascale system

Intel Paragon:
World's fastest computer

With this Extreme Concurrency Comes the Challenge of Efficient Application Execution

Performance analysis will become increasingly important as processor counts grow

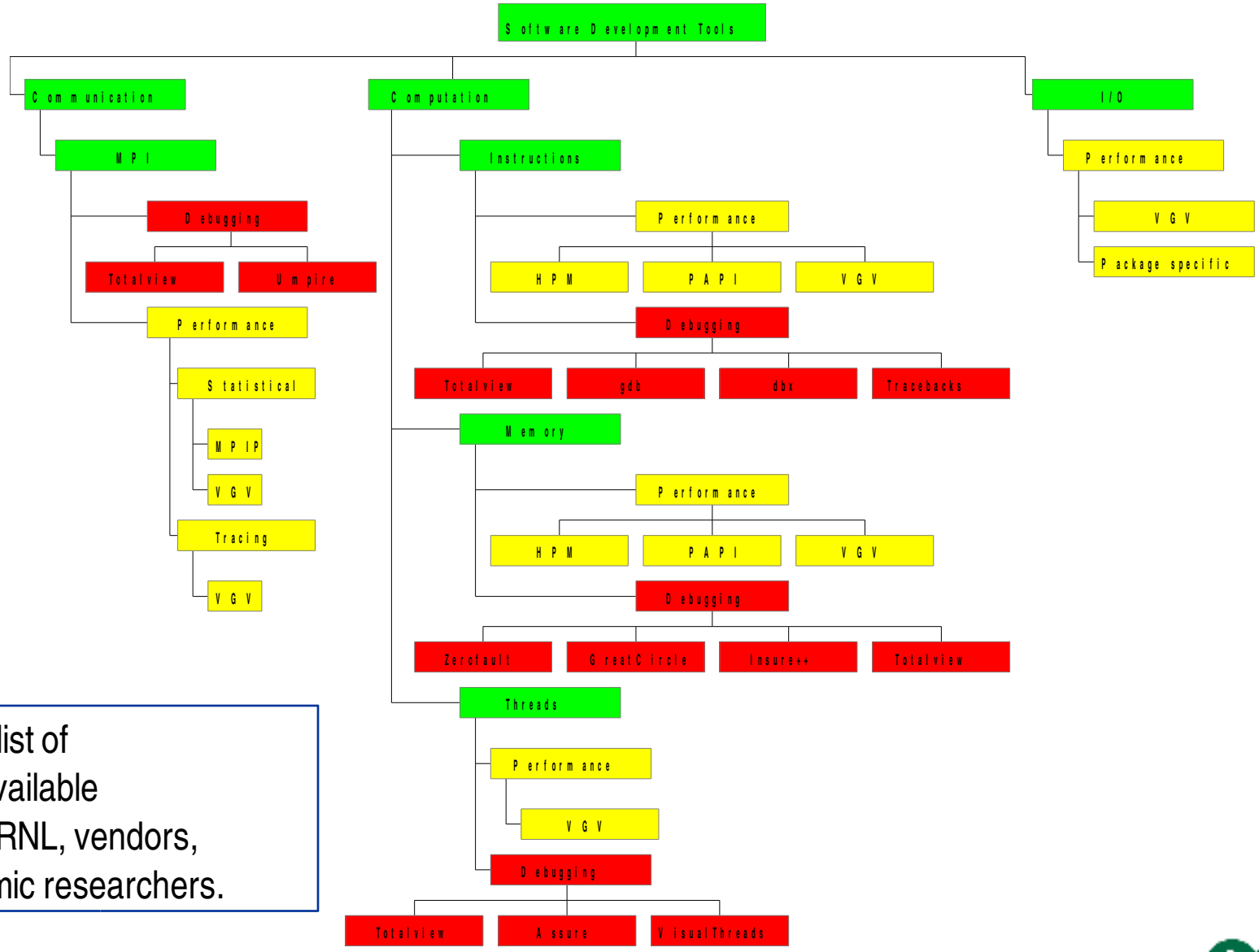
- Even small inefficiencies can be amplified

Understanding performance at this level of concurrency can be impractical and burdensome

Several areas of performance analysis technology don't scale well

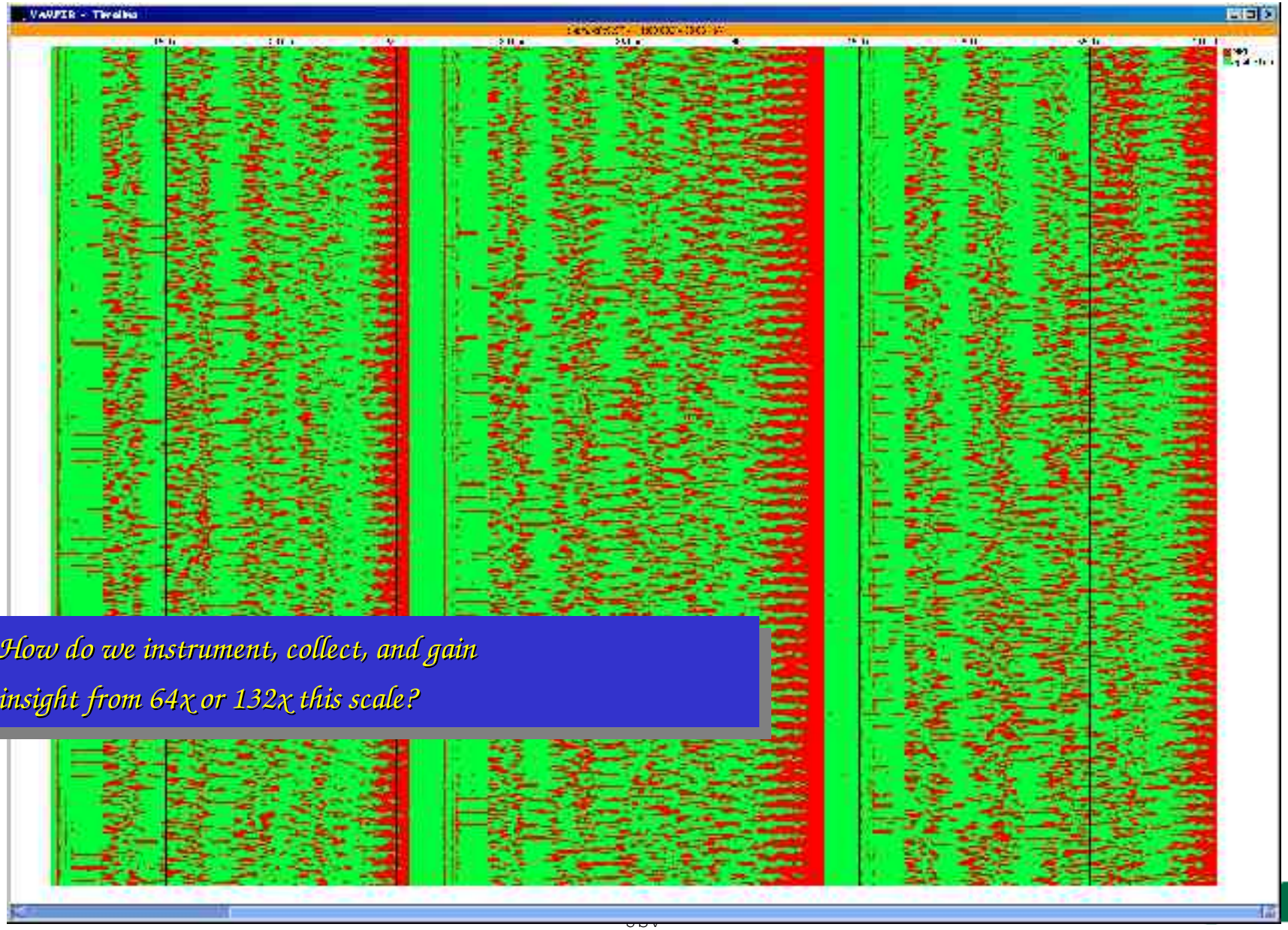
Concurrency	O(100)	O(1,000)	O(10,000)	O(100,000)
Instrumentation	OK	OK	OK	OK
Instrumentation management	OK	Hurdle	Barrier	Barrier
Data management	OK	Hurdle	Barrier	Barrier
Data interpretation	Hurdle	Barrier	Barrier	Barrier

Assortment of Tools to Address Different Topics at Varying Levels of Detail



Partial list of tools available from ORNL, vendors, academic researchers.

Scalability Remains a Major Challenge on Today's Platforms: sPPM at 1024 tasks



How do we instrument, collect, and gain insight from 64x or 132x this scale?

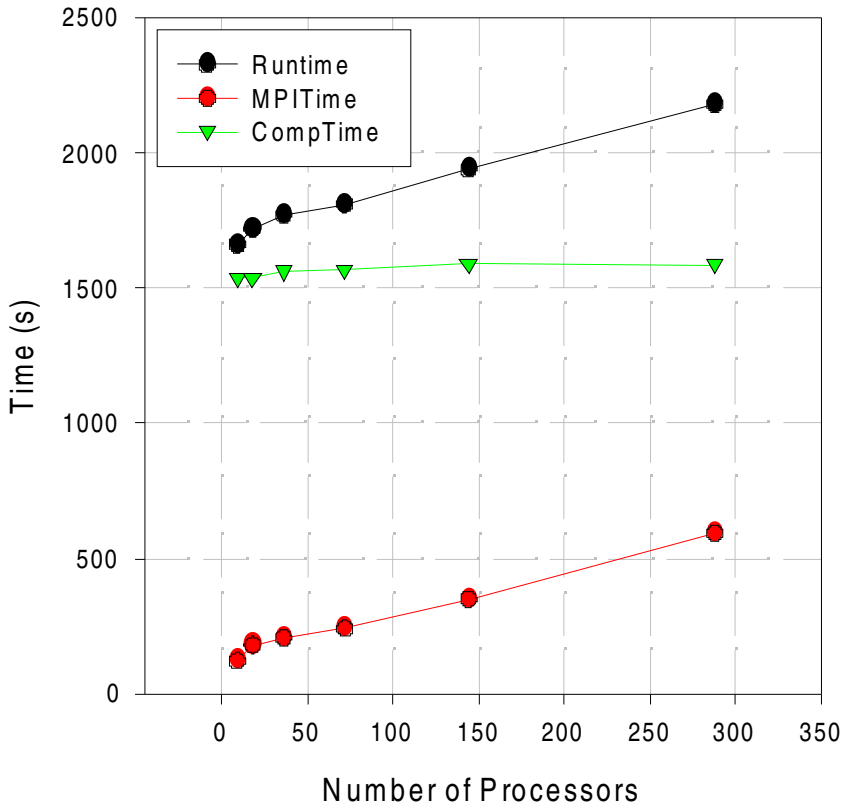
MPIP Provides Scalable MPI Performance Data

MPIP provides lightweight, scalable timing performance data

- Does NOT collect a trace

Tested up to 4096 processors

Gives statistical timing and payload information about MPI callsites



MPIP release 2.5 planned for March. See www.ccs.ornl.gov/~vetter

J.S. Vetter and M.O. McCracken, "Statistical Scalability Analysis of Communication Operations in Distributed Applications," Proc. ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPOPP), 2001.

Hardware Performance Counters are Invaluable for Understanding Computation Performance

Hardware counters measure empirical data that can help a user optimize and understand their code

- Low perturbation
- Real values (cf. static analysis, analytical modeling)

Despite some of their shortcomings, these counters are one of the few practical tools available

- Ill-defined
- Inconsistencies across platforms

Some systems have many counters and events

- POWER4 – over 200 events
- IA-64 – over 900 events

High dimensionality is compounded by extreme concurrency

Traditional Techniques (Sweep3d 256 mpi tasks) Yield Huge Number of Data Points

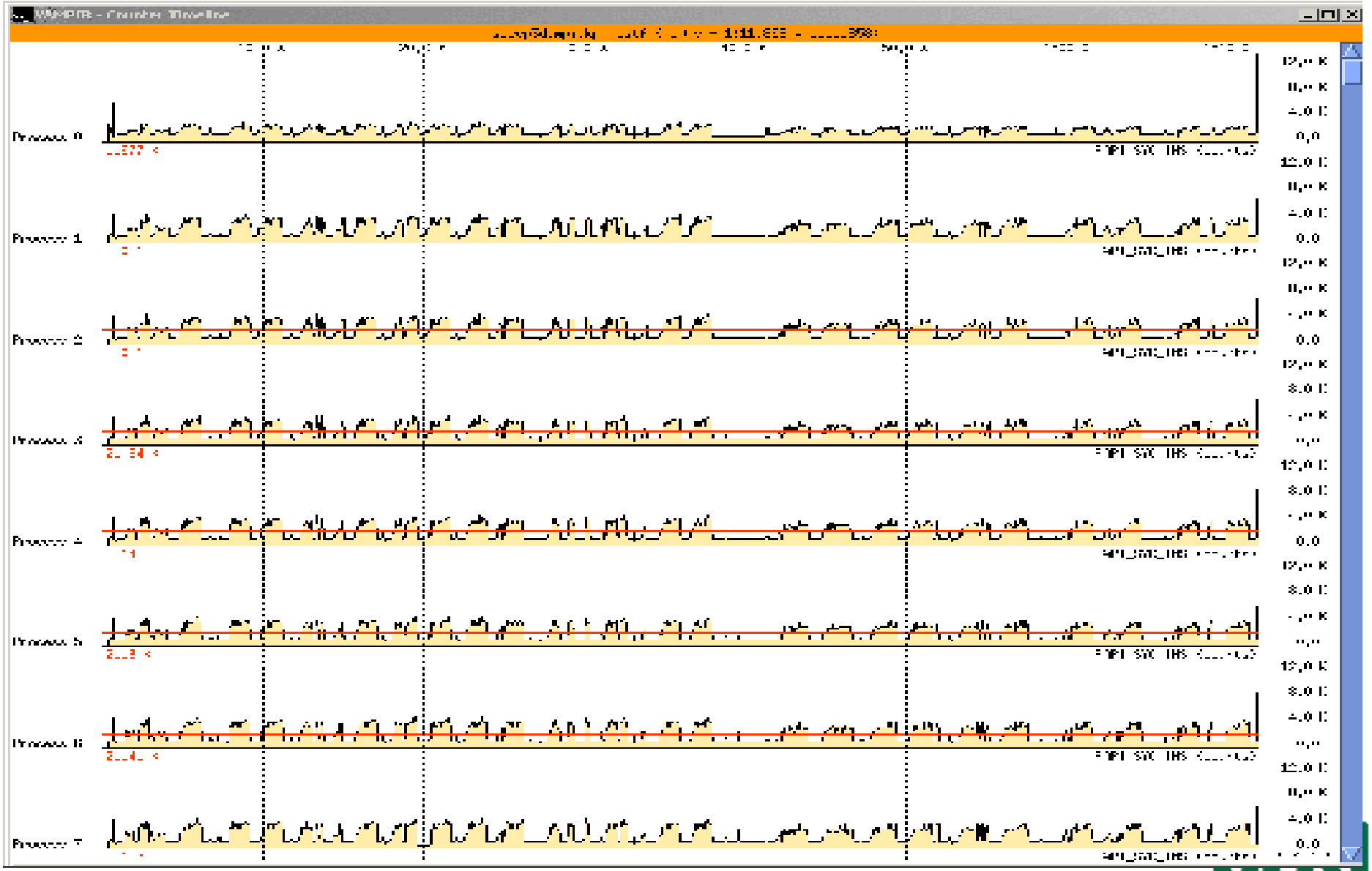
G: Instrumentation ID	P: MPI Task	S: Instance								
1	1	1	83057605	77956513	22653498	1468948	7299392	37442673	21232357	1253921
1	1	2	82331147 04	75877135 87	22574422 17	8987587 8	7281691 3	36129321 04	21007529 84	1260163 843
1	2	1	81973603 00	77017507 98	22330703 95	1469595	7342519 9	37369569 16	20757228 13	1237231 691
1	2	2	81351386 63	75937600 65	22074563 47	9172755 9	7369905 7	35903746 14	20603110 24	1230463 534
2	1	1	83263293 68	75591985 51	24011955 35	1458386	7238297 5	37173268 84	20786530 42	1233604 969
2	1	2	82917911 04	74212484 64	23346289 95	8509892 9	7207491 2	35405216 69	20600234 81	1230670 083
2	2	1	84051067 10	76450556 63	24153969 52	1465589	7278521 8	37087982 98	21047398 98	1248538 879
2	2	2	83810619 57	75237537 89	23772760 92	8606055 6	7260832 4	35532887 29	20844958 01	1256915 508
			56	02	28		9	76	57	516

<num events> X <num sections> X <num instances> X <num tasks>

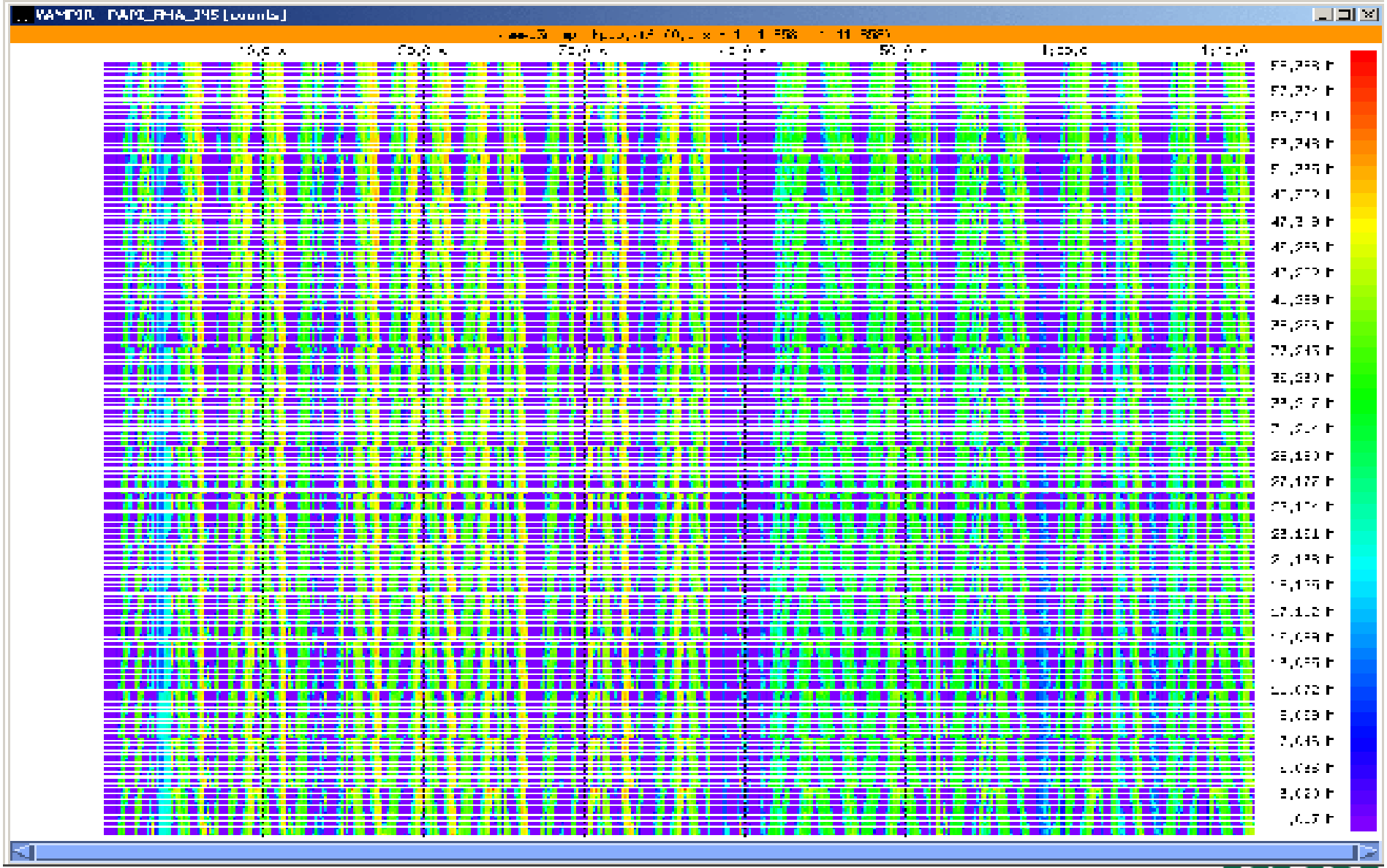
= 23 events X 5 sections X (1~12 instances) X 256

= 353,280 values !!

Try Mapping One or Two Dimensions to Timeline? Scalable?

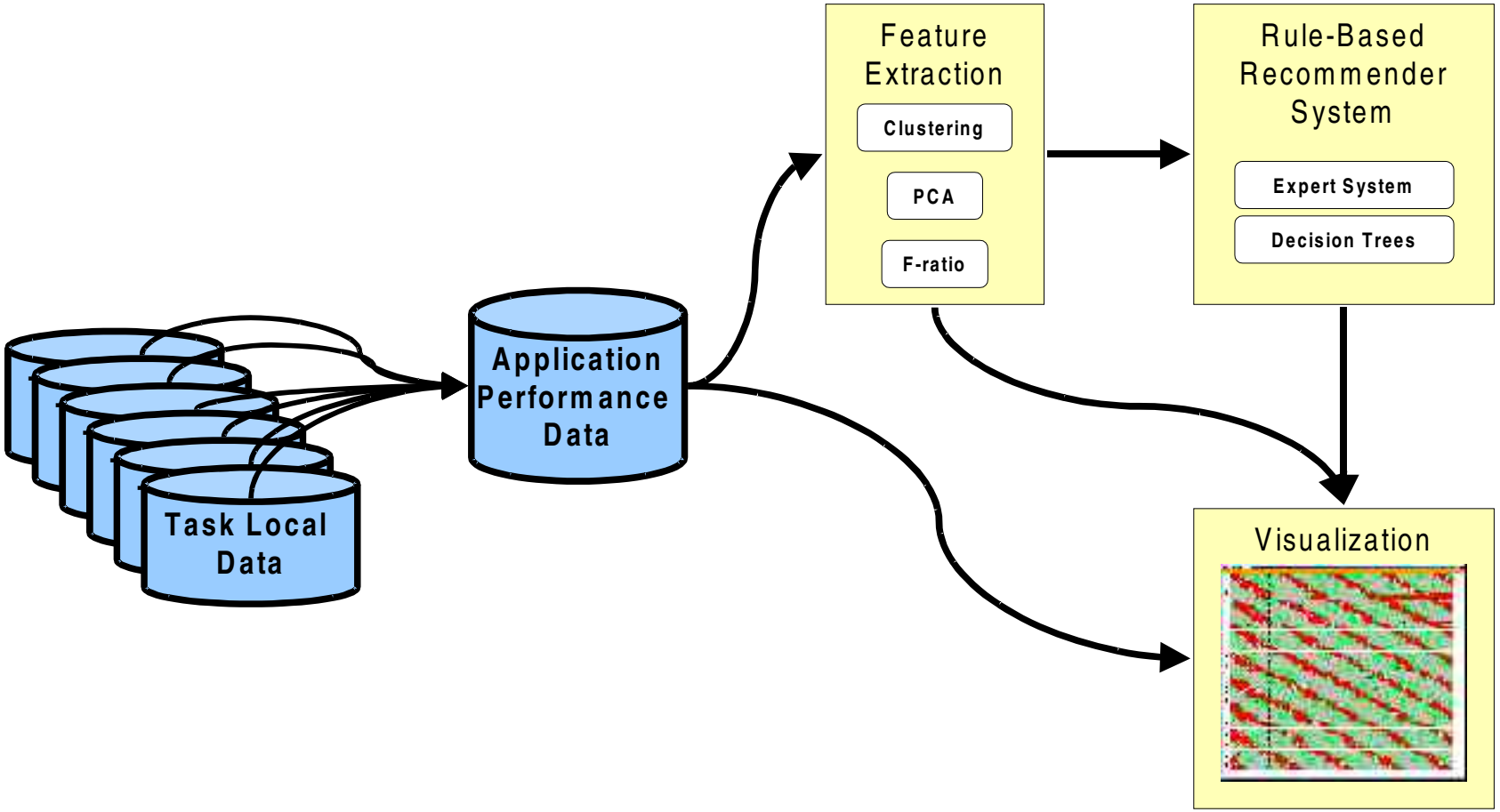


Try Mapping One or Two Dimensions to Timeline w/ Color-coding? Scalable?



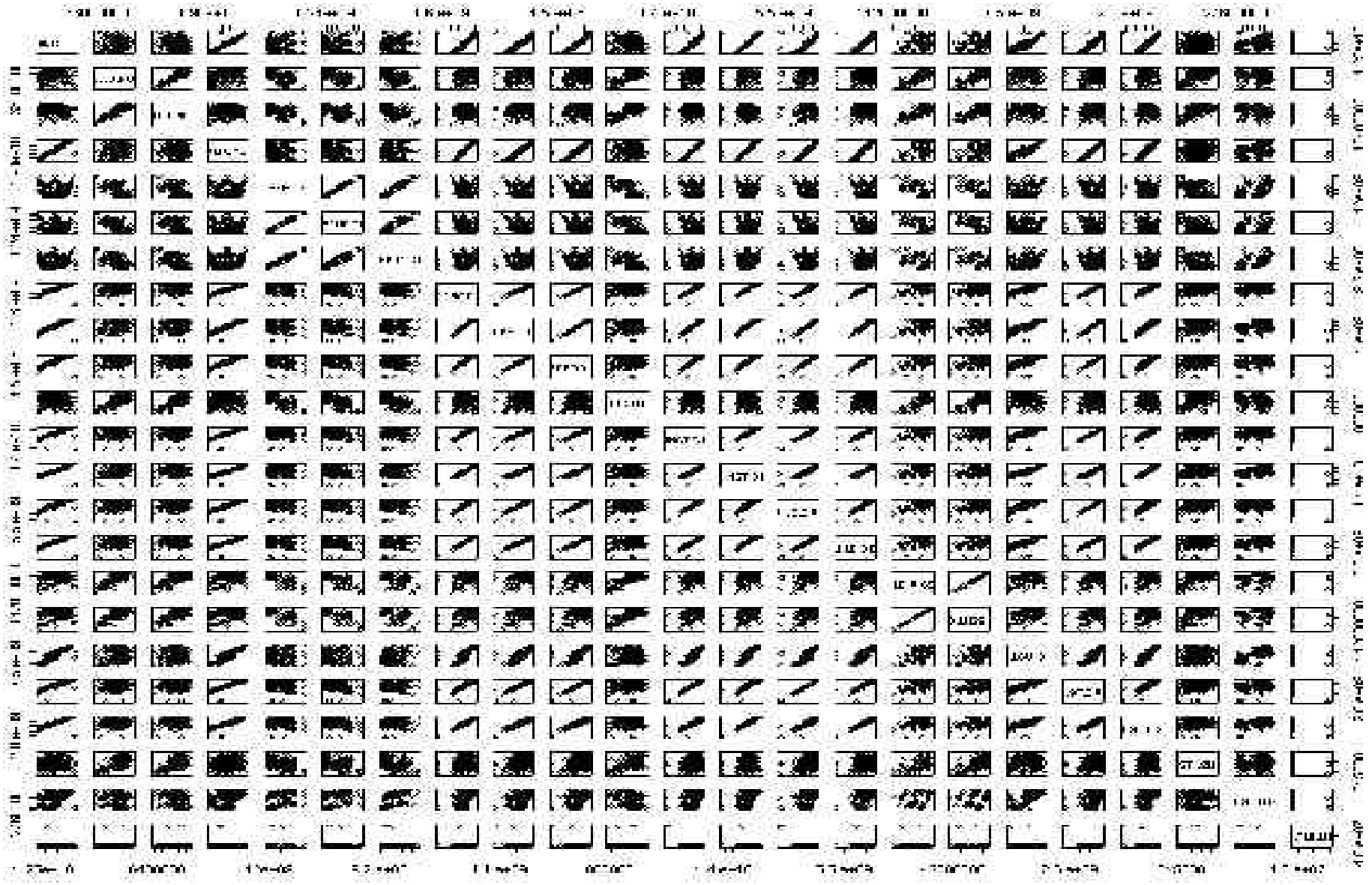
other tasks?

New Solution: Use Multivariate Statistics to Augment Analysis



Joint work with D.H. Ahn.

Sweep3D Scatterplot Matrix Illustrates Correlations between Counter Values



Many Facets of Multivariate Statistical Analysis are Useful

MSA provides a means to find relationships among variables (hardware metrics dimension) and individuals (i.e. mpi tasks or omp threads dimension)

Cluster Analysis groups together the processing elements (i.e.mpi tasks) that behave statistically similarly in hardware counter data space

Factor Analysis allows us to group together those metrics that explains the same underlying factor

Combining CA/FA and using their parameters (F-ratio, factor loadings) provides a variety of viewpoints on hardware counter dataset in more meaning ways

One Example of MSA: Cluster Analysis

Techniques

- Hierarchical
- K-means

Hierarchical method provides a general idea about cluster structure on dataset by building a hierarchical tree

Both methods work well to group together similar processing elements, based on dissimilarity matrix

Once datapoints are clustered, F-ratio (between-cluster variability/within-cluster variability) can identify important events that yield the particular cluster structure

Sweep3D Performance Experiment

Description

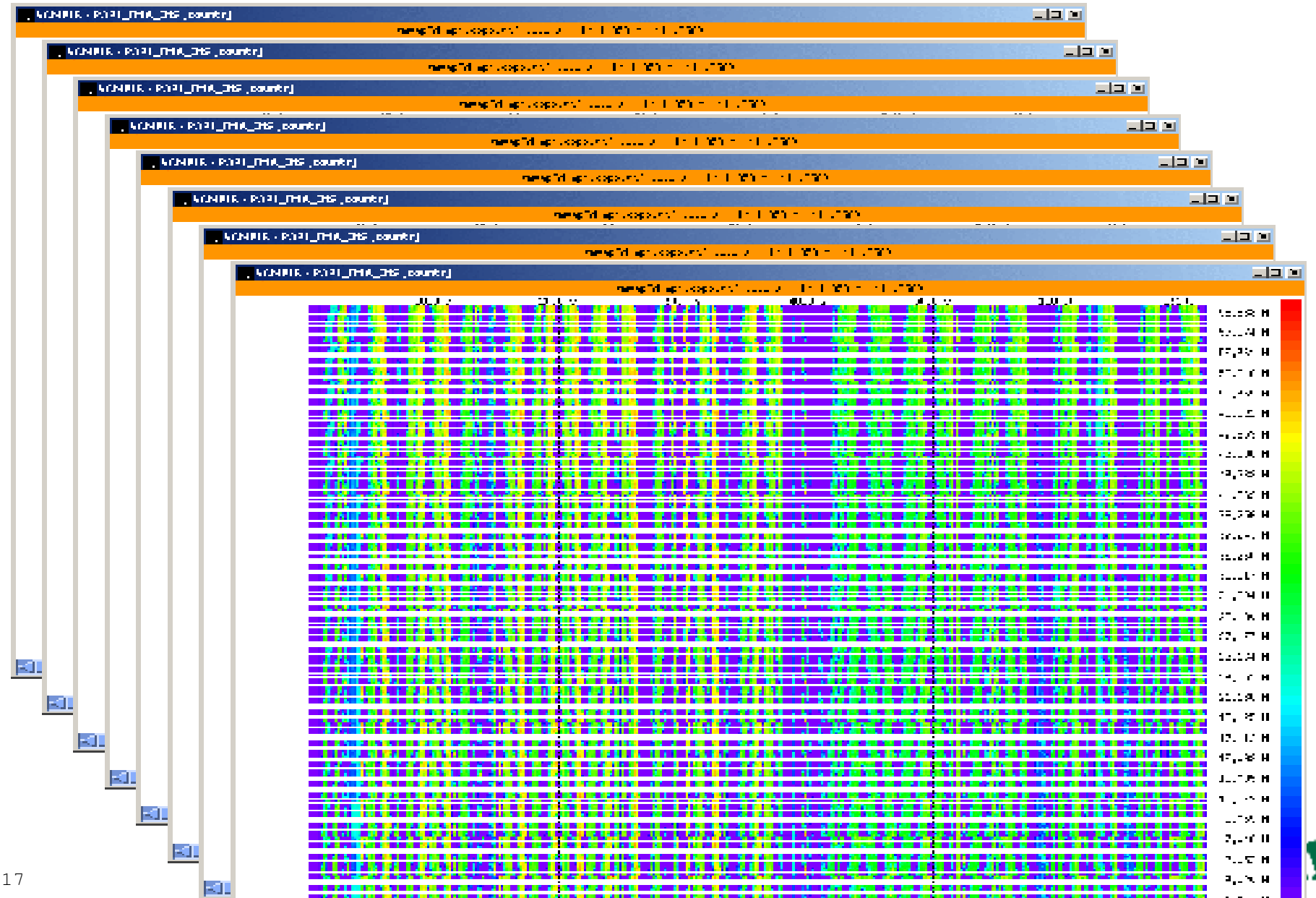
- A solver for the 3-D, time-independent, particle transport equation on an orthogonal mesh. Sweep3D uses wavefront algorithm for discrete ordinates deterministic particle transport simulation.
- Sweep3D exchanges messages between processors as wavefronts propagate diagonally across the 3-D space. Sweep section is the core of this algorithm. About 63% of its aggregate time spent in this section.

Experiment

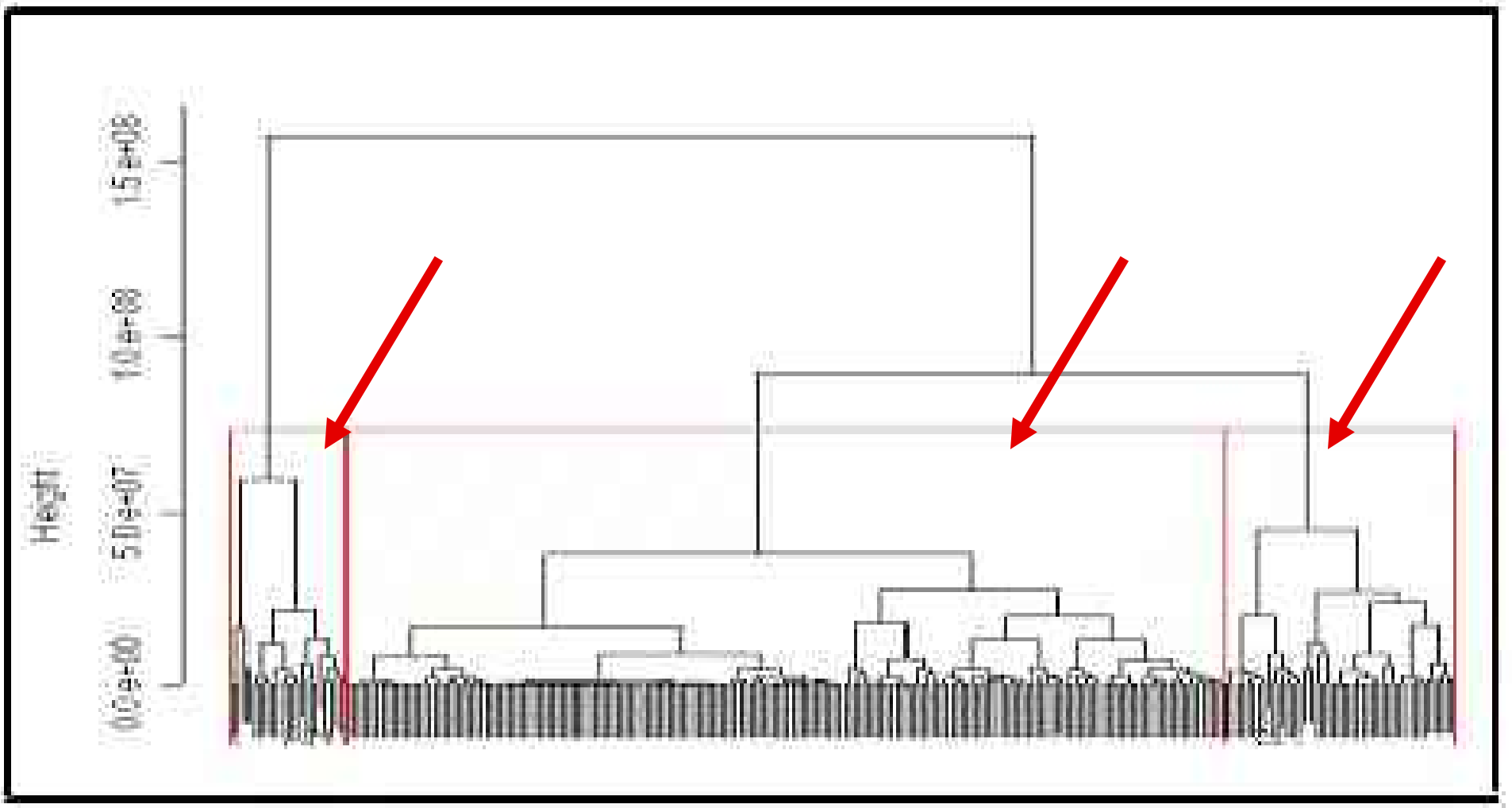
- Problem size : 512 X 512 X 150
- Num of tasks : 256 MPI tasks (16 X 16)

Capturing 8 counter events for one code section for each MPI task

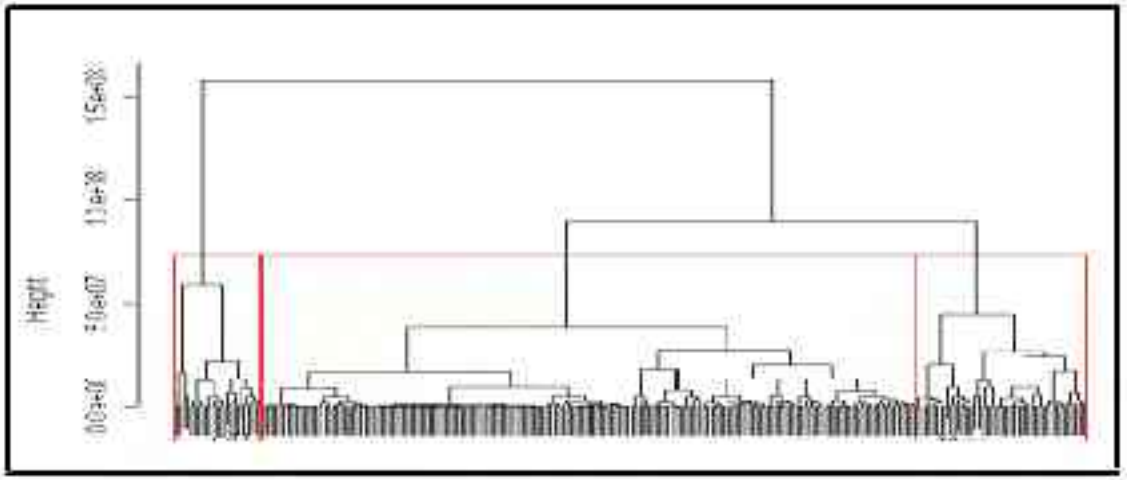
Curse of Dimensionality: Each Counter Generates a Separate Dimension of Values over Time

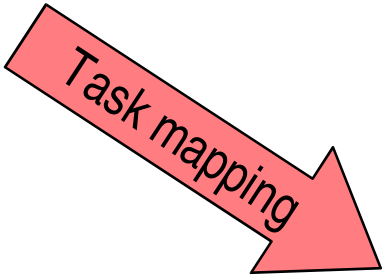


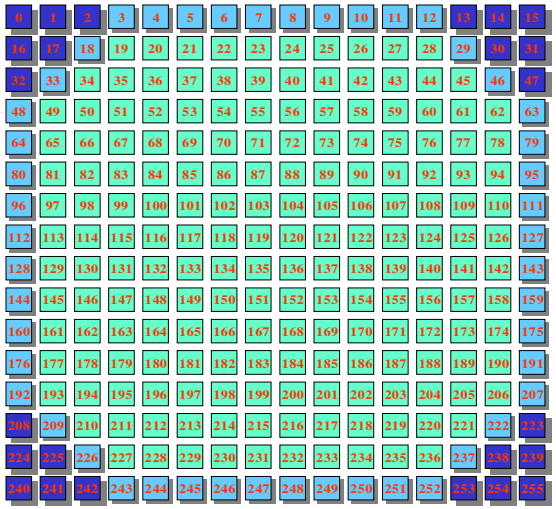
Dendrogram Illustrates the Clusters for the Counter Values across All 256 MPI tasks



Mapping the Clusters to the Application Topology Reveals Interesting Results







Performance optimizations to a representative translate to entire cluster

Easily identify metrics that separate performance clusters

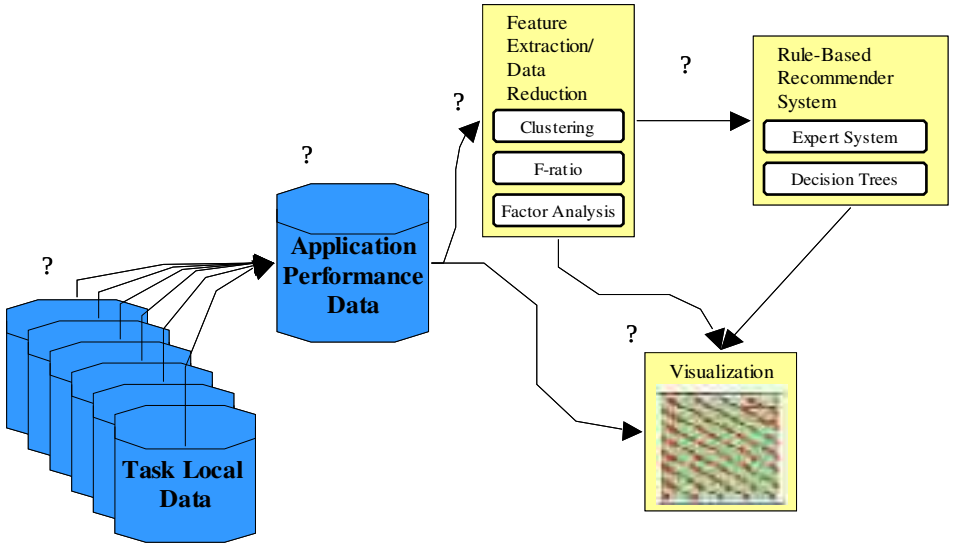
Collapse 256 tasks into 3!

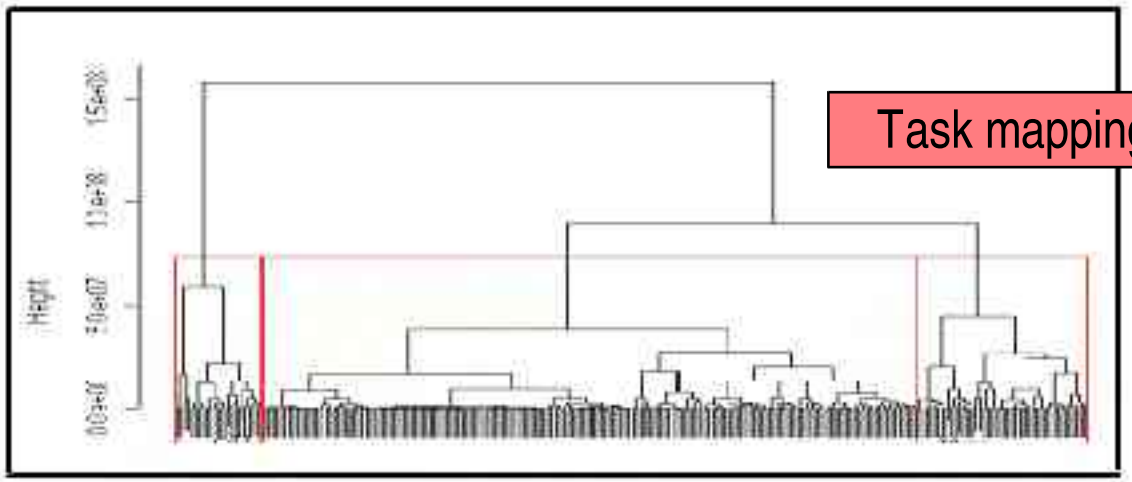
Summary: Multivariate Statistical Analysis of Hardware Counter Data

Hardware counters produce huge amounts of data on large systems

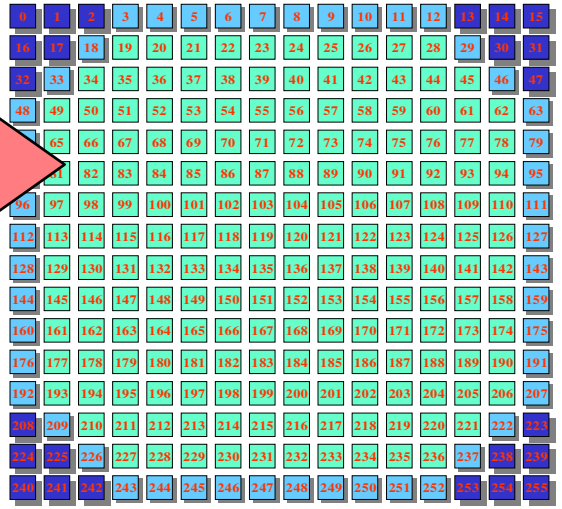
Multivariate statistical techniques help distill important features

Clustering, Factor analysis, PCA





Task mapping



D.H. Ahn and J.S. Vetter, "Scalable Analysis Techniques for Microprocessor Performance Counter Metrics," Proc. SC 2002, 2002.

Other Performance Analysis Techniques to Address Scalability

Automatic Classification for MPI Trace Analysis

Use decision tree classification (a supervised learning technique) to classify application's messages automatically

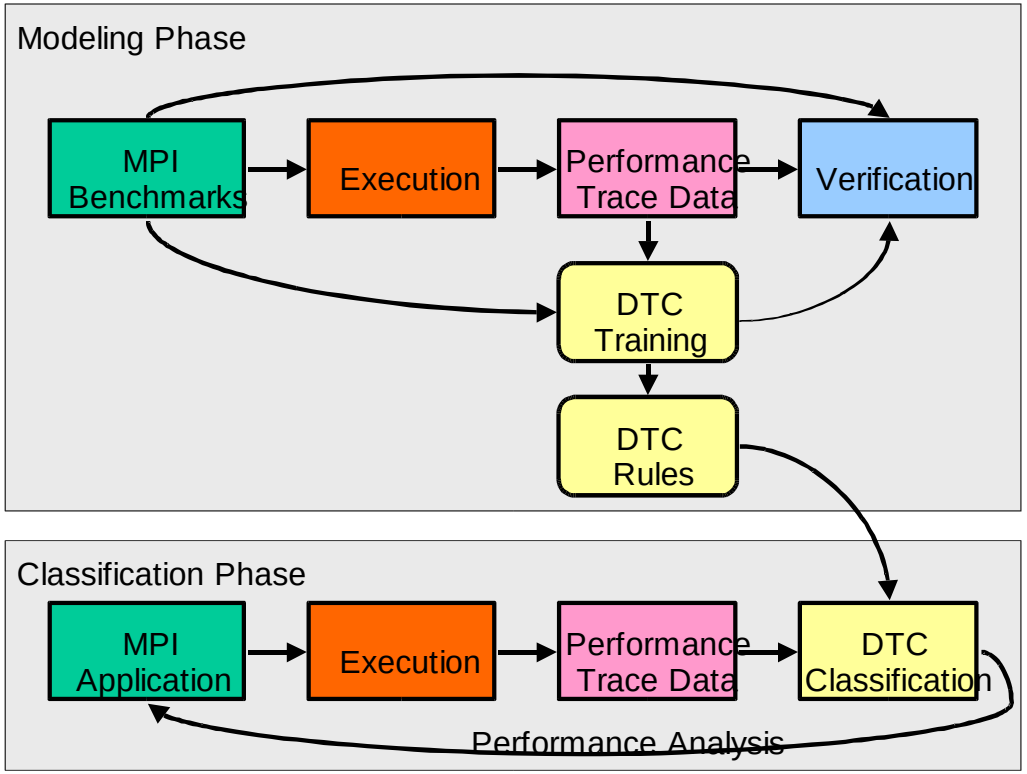
Compare an application's message operations to 'normal' communication for a particular MPI configuration

Modeling Phase (once)

- –Use benchmarks to generate decision tree
- –Both efficient and inefficient

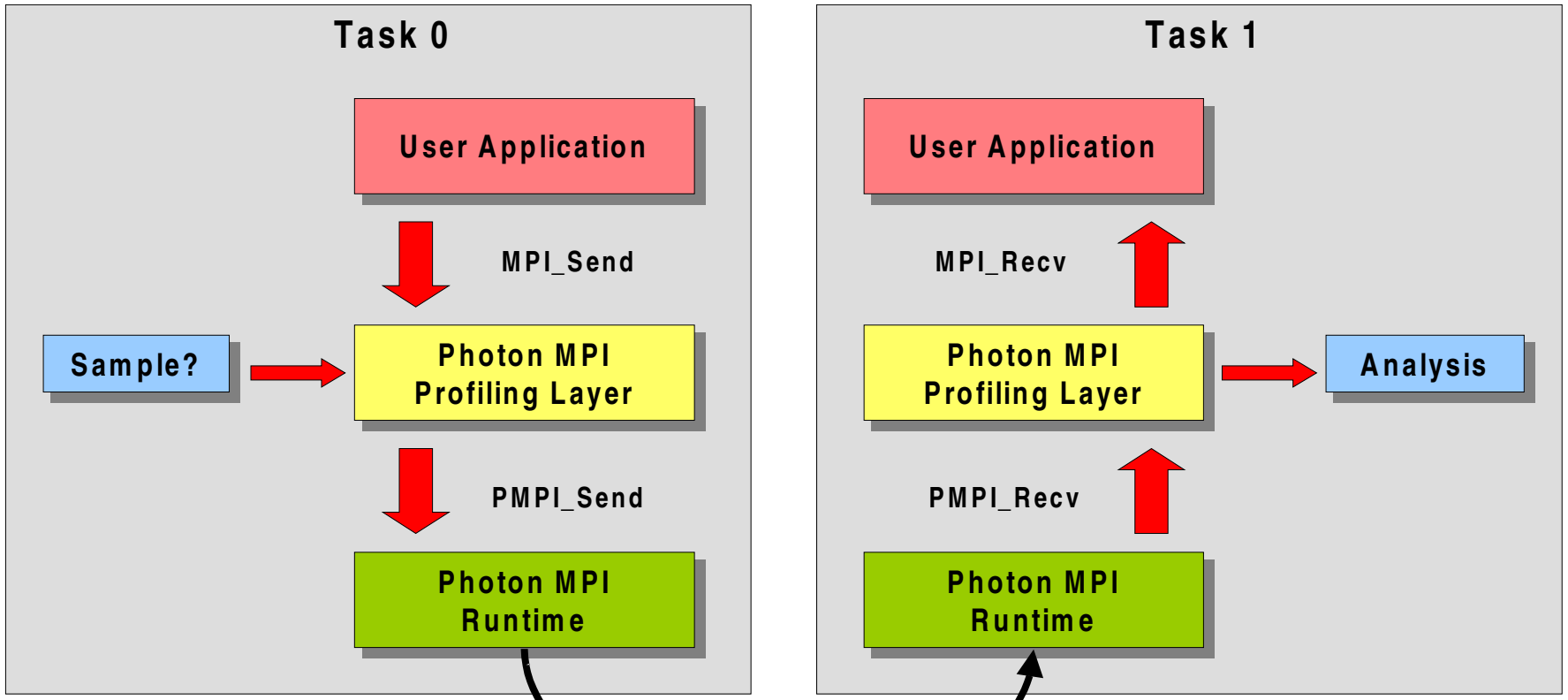
Classification Phase (many)

- –Execute application
- –Analyze application trace with classifier based on decision tree



J.S. Vetter (2000). *Performance Analysis of Distributed Applications using Automatic Classification of Communication Inefficiencies*. Proc. ACM Int'l Conf. Supercomputing (ICS).JSV

Use Message Sampling and Runtime Analysis to Reduce Data and Perturbation



Modified MPI runtime carries tiny amount of data with each message.

Photon Tagged Message

All communicating pairs in the system sample and analyze independently.

J.S. Vetter (2002), *Dynamic Statistical Profiling of Communication Activity in Distributed Applications*. Proc. SIGMETRICS: International Conf. Measurement and Modeling of Computer Systems

Conclusions

Massively parallel computing is here

Microprocessor performance counters provide rich information about application performance

Traditional techniques for performance analysis of hardware counter activity do not scale very well

- How do we interpret all the counter data?

We propose a new solution

- Apply multivariate statistical methods

Experiments reveal evidence that these techniques allow

- Easier understanding of counter metrics
- Reduce data management problems
- Improved productivity

More information at www.ccs.ornl.gov/~vetter

Bonus Slides