Large Scale Performance Analysis

Judit Gimenez, Jesus Labarta , Harald Servat

judit@cepba.upc.edu

Technology Transfer Research Training Mobility of Researchers

User Support Education HPC Facilities Parallel Expertise

Where we live

Traceland … **… aiming at detailed analyses and flexibility in the tools.**

- **Importance of details**
  - Variance is important
    - ✓ Along time, across processors
  - Highly non linear systems
    - ✓ Microscopic effects may have large macroscopic impact

Paraver (www.cepba.upc.edu/paraver)

(1992-)

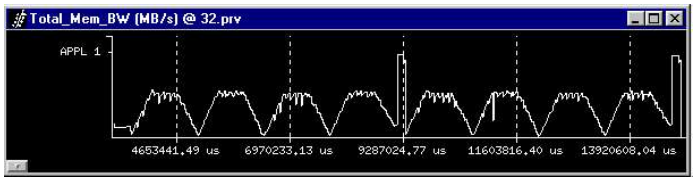
- **Performance visualization tool**
  - Off line analysis of traces
- **Qualitative / Quantitative information**
- **Comparative analyses**
- **Powerful: Flexibility !!!**
  - Performance analysis ≡ search on a huge and fuzzy space
  - One would better
    - ✓ Be equipped with flexible tools ...
      - – No semantics in the tool
    - ✓ ...supporting quantitative analysis ...
    - ✓ and be ready for surprises

Paraver: Performance Data browser

tunable 

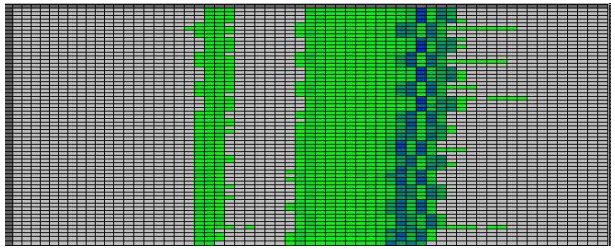
Performance index : s(t) (piecewise constant)

- Identifier of function
- Hardware counts
- Miss ratios
- Performance (IPC, Mflops,…)
- Routine duration
- • •



Seeing is believing

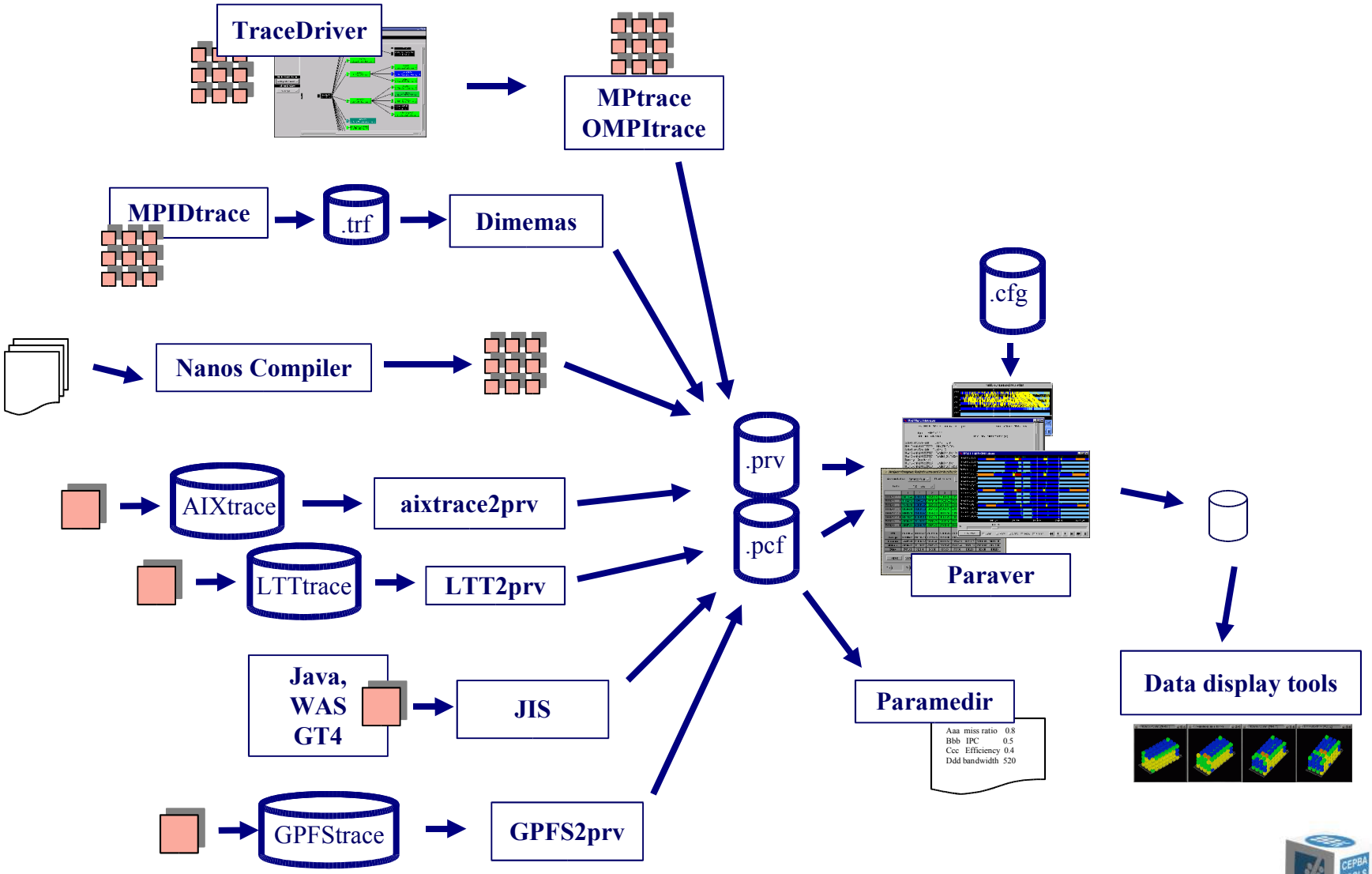
measuring is better



Statistics

- Profiles
- Average miss ratio per routine
- Histogram of routine duration
- Number of messages
- • •

CEPBA-Tools


TraceDriver
MPtrace
OMPItrace
MPIDtrace
.trf
Dimemas
.cfg
Nanos Compiler
.prv
.pcf
AIXtrace
aixtrace2prv
Paraver
LTTtrace
LTT2prv
Java,
WAS
GT4
JIS
Paramedir
Aaa miss ratio 0.8
Bbb IPC 0.5
Ccc Efficiency 0.4
Ddd bandwidth 520
Data display tools
GPFStrace
GPFS2prv

Index

- **Scalability**
- **Where?**
  - Instrumentation
  - Visualization
- **CEPBA-tools evolution**
- **Conclusions**

Work partially supported by EC (HPC_Europa), IBM, MEC

Scalability

- **Our questions**
  - What is scalability?
  - How scalable is an analyst?
  - How far can a trace based approach go?

Scalability

- **What is scalability?**
  - Ability to handle large # processors, time, event frequencies, …
  - Ability to detect and focus on the relevant information at the appropriate level of detail
  - Ability to perform a deep analysis of large applications without looking at the source code
  - Applicable in many areas

- **Key issue**
  - size vs. dynamic range !!!

Index

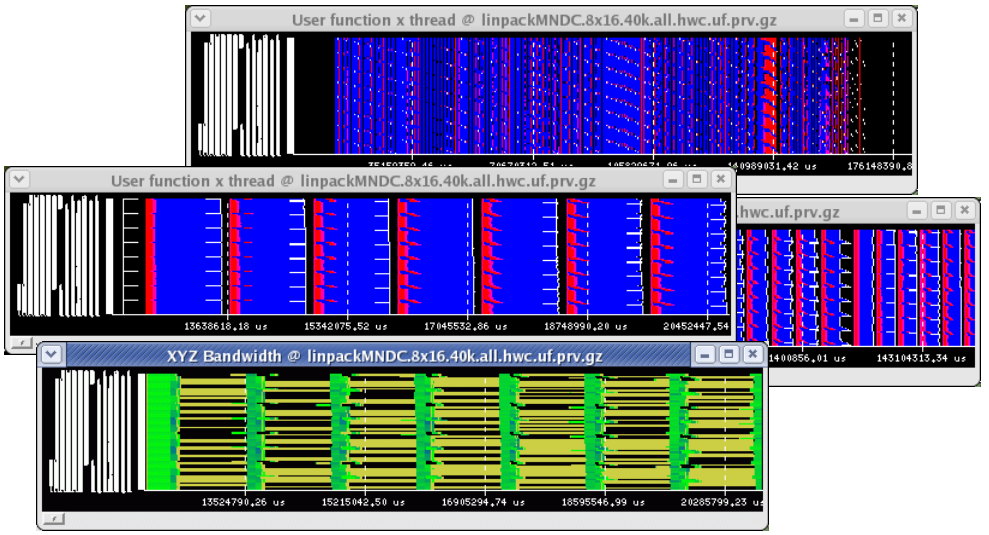
- Scalability
- **Where?**
  - Instrumentation
  - Visualization
- **CEPBA-tools evolution**
- **Conclusions**

Scalability of tracing

- **The issue:**
  - Sufficient information / sufficiently detailed
  - Usable by presentation tool
- **At run time**
  - What, when and who to instrument
  - User driven / automatic / intelligence
- **At merge time**
  - Filter/summarize/discard information
  - Scalability of the merge process itself

Scalability of tracing

- **Techniques**
  - User specified on/off
    - ✓ Static in source code
      - – Time
      - – Processes
    - ✓ Dynamic at run time
      - – Environment variables
  - Limit file size
    - ✓ Stop tracing when reached
    - ✓ Circular buffer. Dump on request (signal)
  - Drop some information
    - ✓ f.i. not instrument MPI



Linpack 128 procs @ BGL, whole run 157s

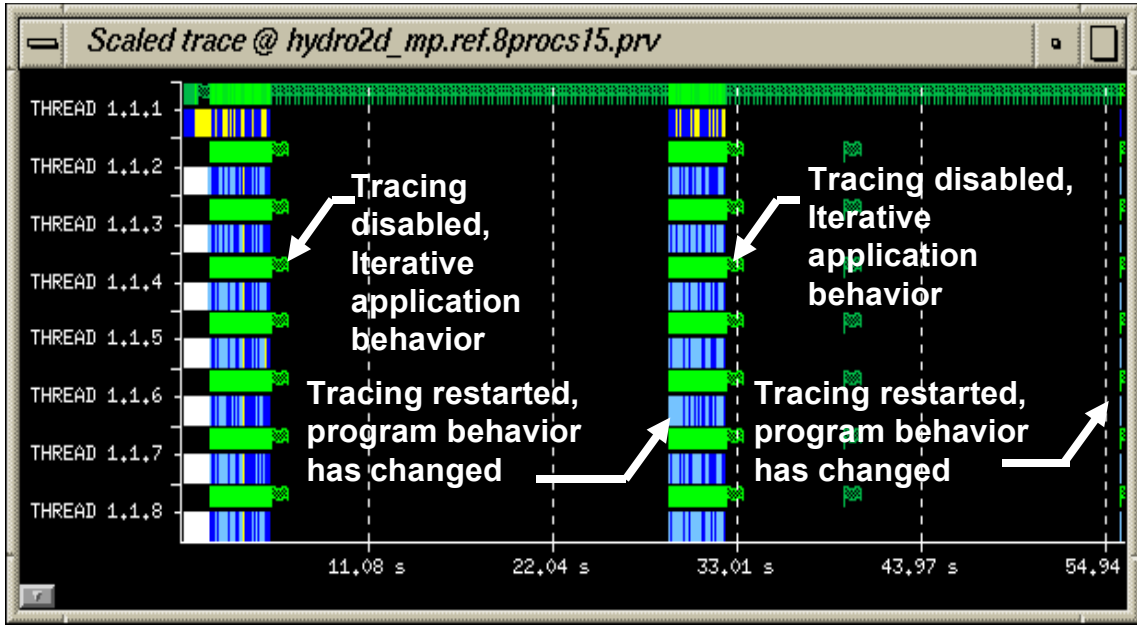
Scalability of tracing

- **Techniques (cont.)**
  - Summarization
    - ✓ What/how
      - – Software counters
    - ✓ When
      - – Periodic samples
      - – Correlated to other events (i.e. function entry/exit)
      - – Threshold frequency
  - *Sampling?*

BT.A.16 Net protocol = IP AIXtrace (164 processes @ 16way node) counter = syscalls	period (ms)	threshold (#events)	size (MB)
	not used		59.9
	10	100	31.2
	100	200	27.1
	10	50	15.6

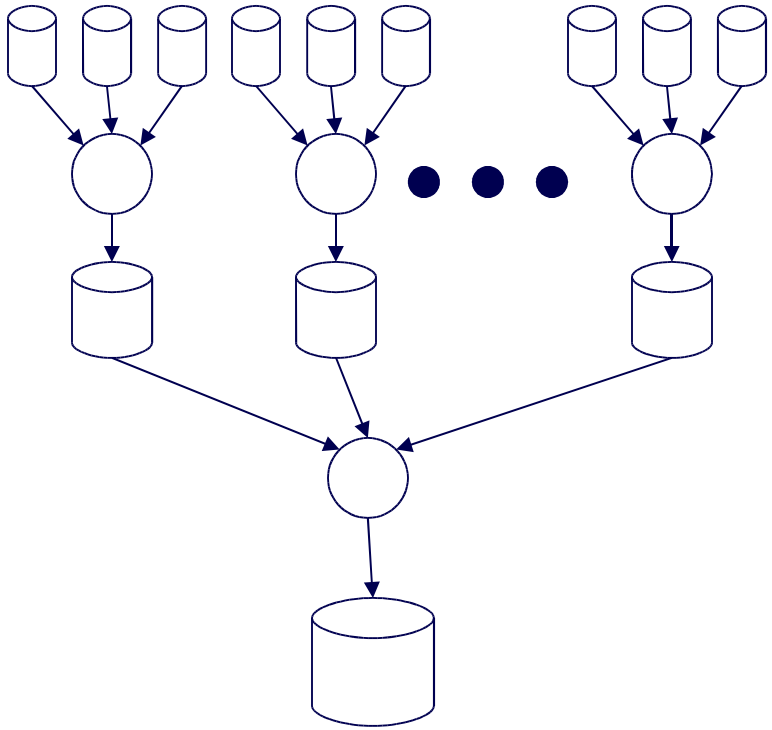
Scalability of tracing

- **Techniques (cont.)**
  - Automatic on/off (2001)
    - ✓ Periodicity detection
    - ✓ Application structure identification



Scalability of tracing

- **Scalable merge**



One file per process

Partially resolved communications

Paraver trace

Index

- Scalability
- **Where?**
  - Instrumentation
  - Visualization
- **CEPBA-tools evolution**
- **Conclusions**

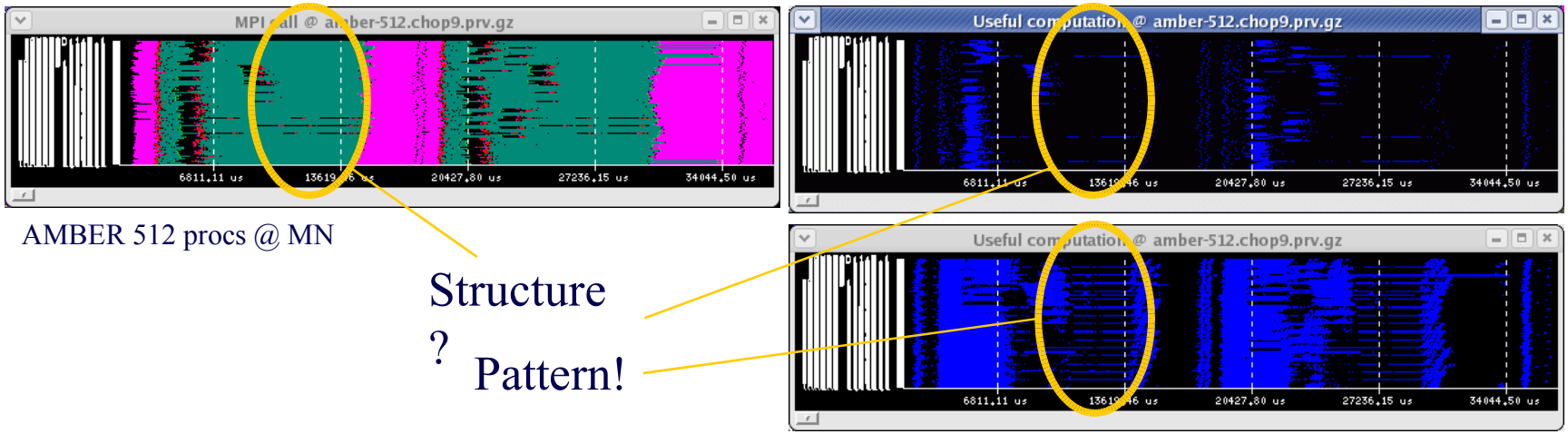
Scalability of visualization

- **The problem**
  - Can we squeeze large amounts of data to obtain information and convey it to the analyst?
    - ✓ Both qualitative presentation and precise/accurate statistics
    - ✓ Support large dynamic range
    - ✓ Actual information often
      - In very small data sets.
      - Scattered
      - In the variance
      - Within perturbed/noisy data

Finding needles in haystacks (select the right KB out of 200MB)

Scalability of visualization

- **Issues**
  - Non Linear Rendering
    - ✓ #pixels << Space x Time
    - ✓ Displayed value:
      - minimum, maximum, average, last, random selection, ….



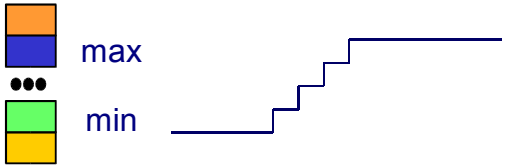
AMBER 512 procs @ MN

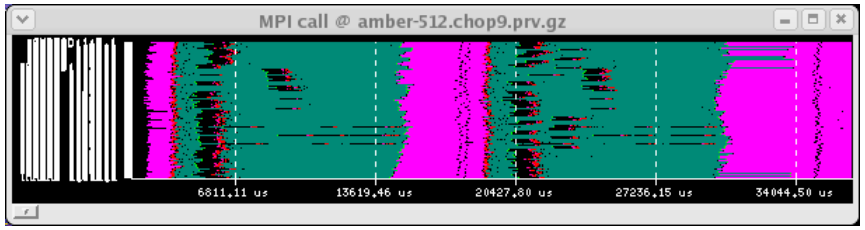
Structure

? Pattern!

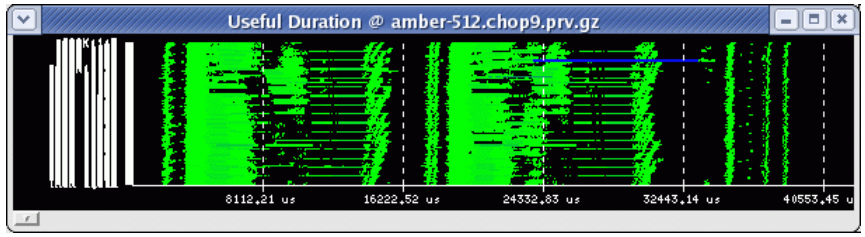
Scalability of visualization

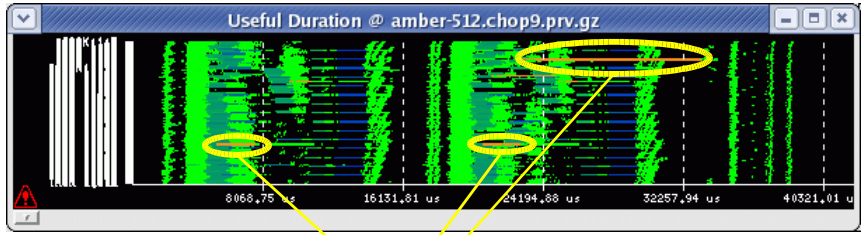
- **Issues (cont.)**
  - Non Linear Rendering (cont.)
    - ✓ Gradient colors
      - – Small #levels





AMBER 512 procs @ MN

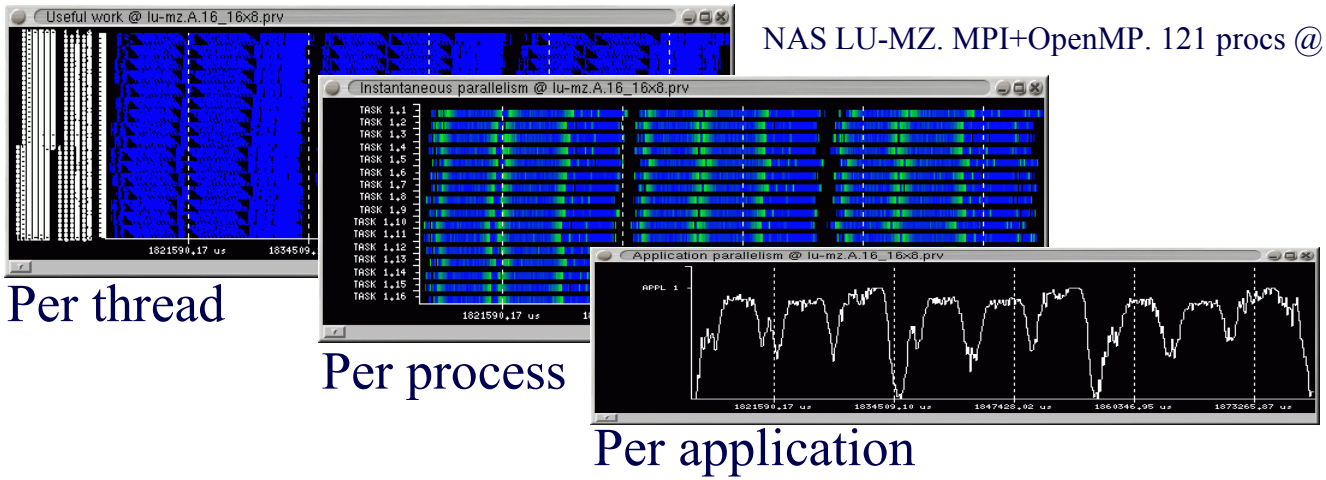




Outliers!

Scalability of Presentation

- **Issues (cont.)**
  - Aggregation - Functional motivation (1992)
    - ✓ Computing an aggregated metric for groups of threads/ processes
      - – Following programming model structure of application
      - – Following resource model



NAS LU-MZ. MPI+OpenMP. 121 procs @ SP

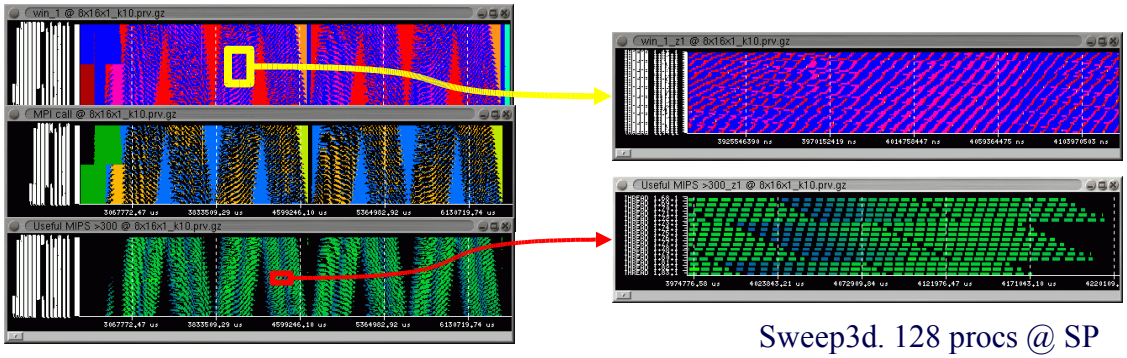
Per thread

Per process

Per application

Scalability of visualization

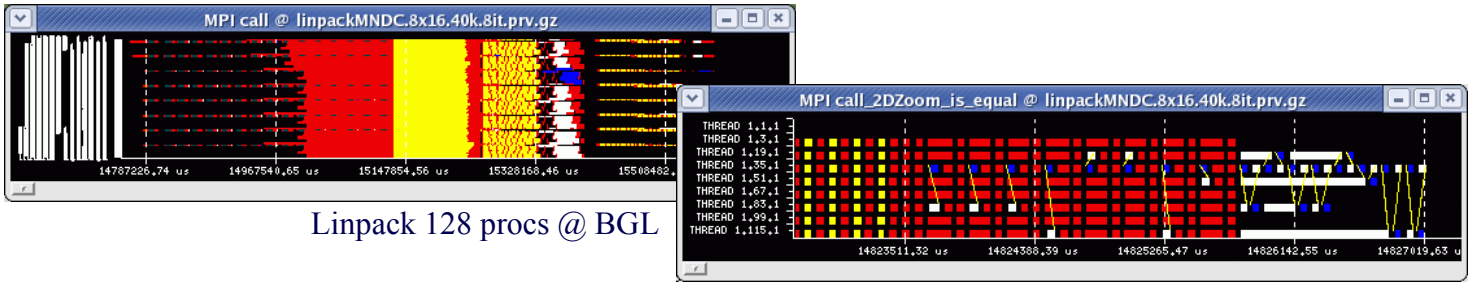
- **Issues (cont.)**
  - Display a subset of processes
    - ✓ Scroll bars, 2D zooms ➔ Limited to contiguous processes



Sweep3d. 128 procs @ SP

    - ✓ Select any subset

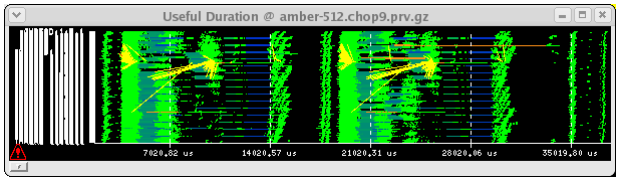
Human eye: a great correlator



Linpack 128 procs @ BGL

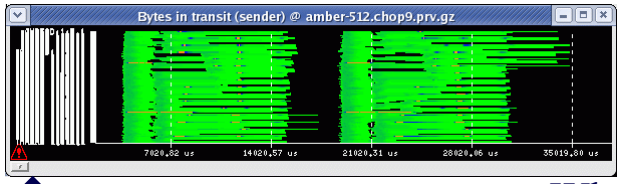
Scalability of visualization

- **Issues (cont.)**
  - Display a subset of processes (cont.)
    - ✓ how to select “relevant” subsets? Integrate analysis and visualization (selection by property) - Functional motivation (2002)



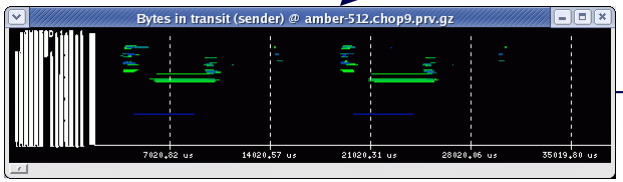
Some proc. delayed

COPY COMM. FILTER

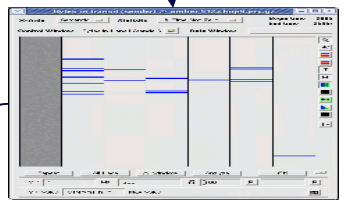


Who sends how much

Who sends how much to 140

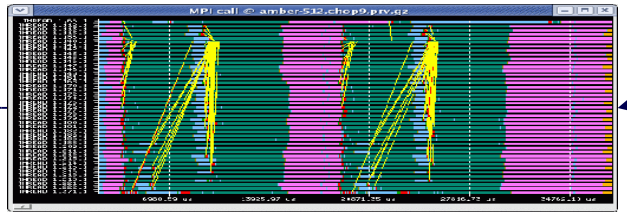


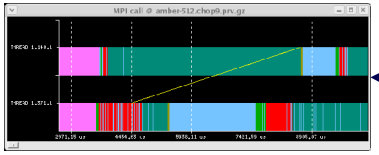
AMBER 512 procs @ MN



Histogram

Selection of senders to 140





MPI impl. does not overlap comp/comm

Index

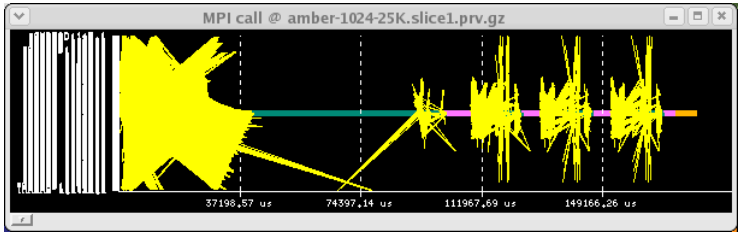
- Scalability
- **Where?**
  - Instrumentation
  - Visualization
  - Trace2trace
- **CEPBA-tools evolution**
- **Conclusions**

Trace2trace

- **The issue**
  - Fit between tracing and visualization
    - ✓ Approach: collect “all” the information and later generate partial “views”
  - Replicate some of the tracing/visualization features
    - ✓ Available information: whatever was emitted to original trace
    - ✓ Batch: more time for computation
  - Minimum loss, maximum compatibility
  - User driven

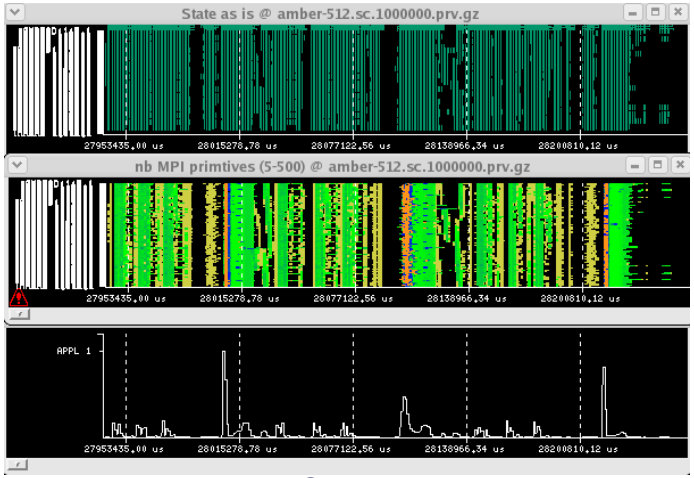
Trace2trace

- **Techniques**
  - Trace size control
    - ✓ On/off: space / time
    - ✓ Limit file size
  - Summarization (software counters)
    - ✓ Used to identify regions of activity, periodicities
  - …



AMBER. 1024 procs (500-530) @ MN

- **Next: Automatic structure identification**
  - Selection of a proper subset of a trace
    - ✓ time / space - MPI calls pattern
  - Point to potential performance problems



AMBER 512 procs @ MN

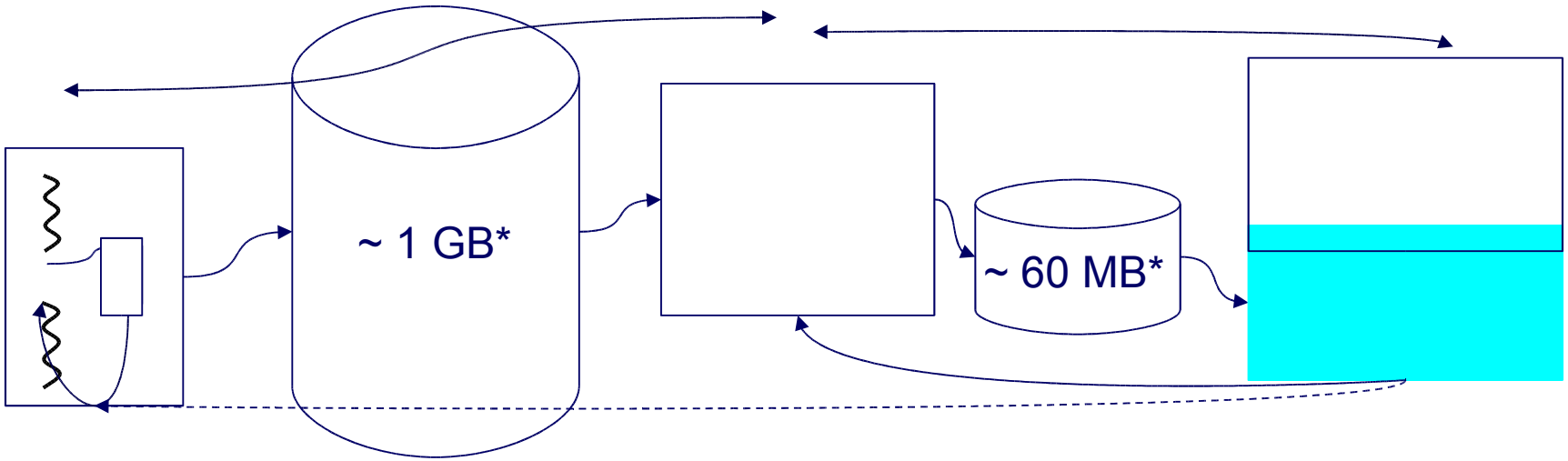
Index

- Scalability
- Where?
  - Instrumentation
  - Visualization
  - Trace2trace
- **CEPBA-tools evolution**
- **Conclusions**

CEPBA-tools towards scalability

Functionality shift?

Functionality shift?



- ✓ Large flat files
- ✓ Scalable merge

- ✓ Simple batch filters
- ✓ Combinations

- ✓ Speed up trace load, metric computation
- ✓ Increase semantic functionality
- ✓ Still memory eager

*** compressed files (gz)**

Conclusion

- **We do believe trace based approaches can be used @ large scale systems**
  - Consider detail of time and space variability
- **We need**
  - Tracing and preprocessing tools
    - ✓ Intelligence/flexibility to select what to trace
  - Analysis tools
    - ✓ Flexible statistics and structure identification algorithms
    - ✓ Flexible selection / focusing mechanisms to expose effects
    - ✓ Loop between stats and selection mechanisms
  - Plus a lot of memory
  - Plus patience, curiosity,….