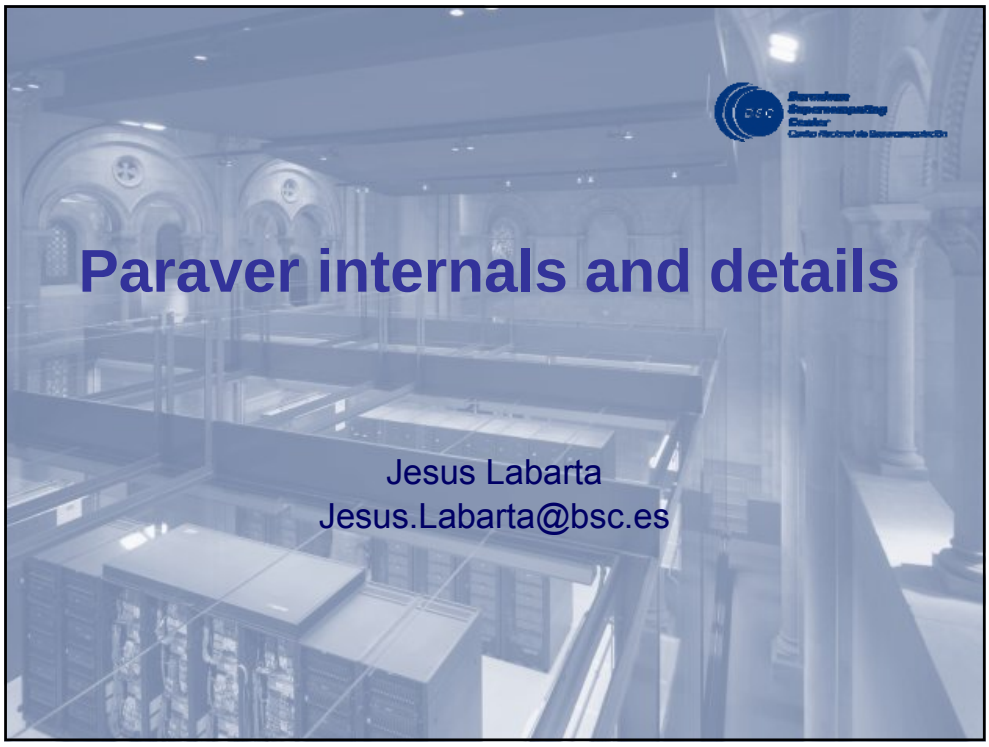


Paraver internals and details

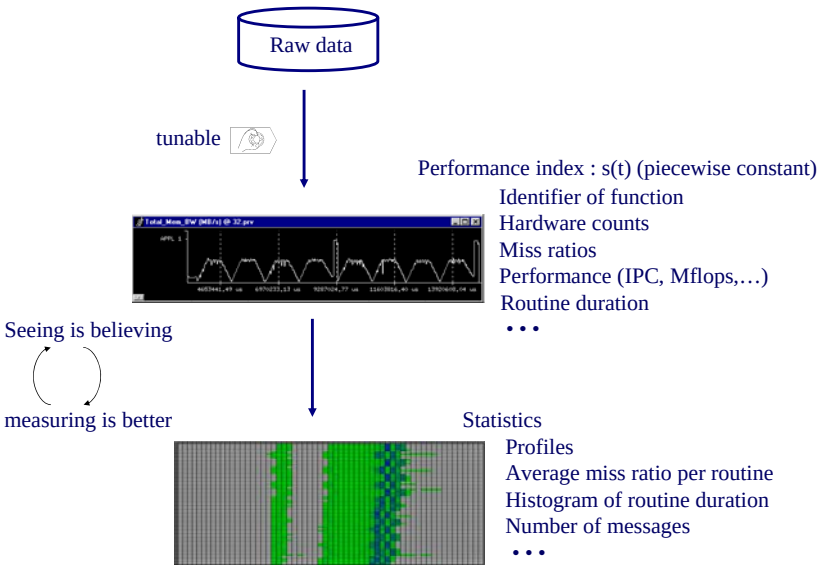
Jesus Labarta
Jesus.Labarta@bsc.es

Index

- Overview
- Semantic module
- Analysis module

overview

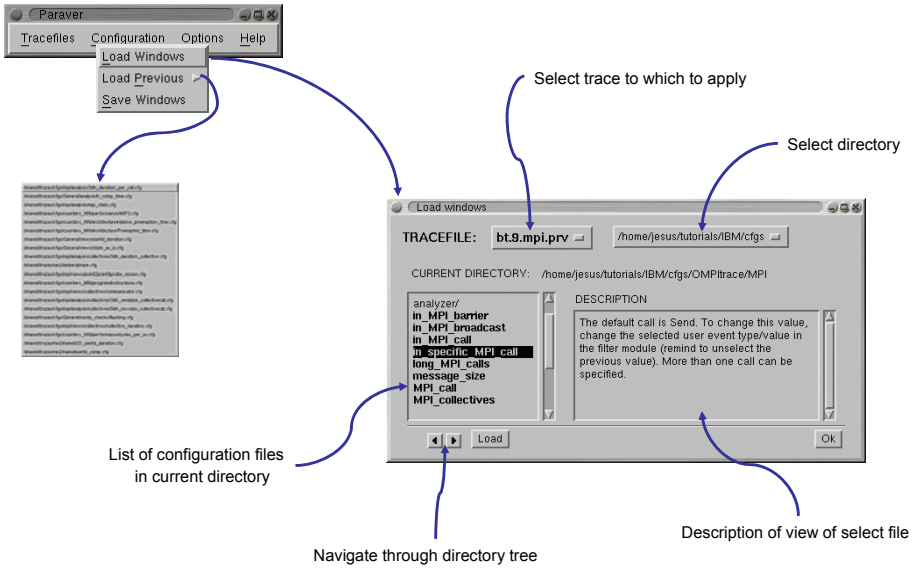
Paraver: Performance Data browser



Performance views

- Few types of views
  - Timelines
  - Textual
  - Statistics
    - Profiles
    - Histograms
- Configuration files
  - Capture knowledge on how to generate view from raw data
  - Loading configuration file pops up display or analysis window
  - Distribution
    - Set of basic view and stats provided with the tool.
  - Also useful for
    - Checkpointing of studies
    - Cooperative work
    - Bug reporting
    - Recording your own favorite views
    - Get used with the tool and to learn on the internals

Configuration files



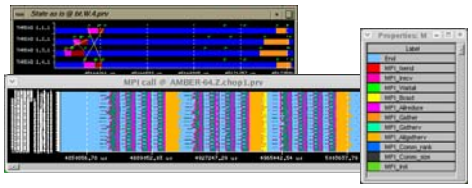
TImelines

- Each window displays one view
  - **Piecewise constant** function of time $S(t) = S_i, i \in [t_i, t_{i+1})$
  - One such function of time per object:
    - Thread, process, application, workload, CPU, node
- Types of functions
  - Categorical
    - State, user function, outlined routine $S_i \in [0, \mathrm{n}] \subset N, \quad n <$
  - Logical
    - In specific user function, In MPI call, In long MPI call $S_i \in \{0, 1\}$
  - Numerical
    - IPC, L2 miss ratio, Duration of MPI call, duration of computation burst $S_i \in R$

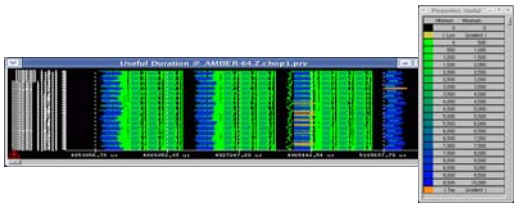
Timelines

- Representation
  - Function of time

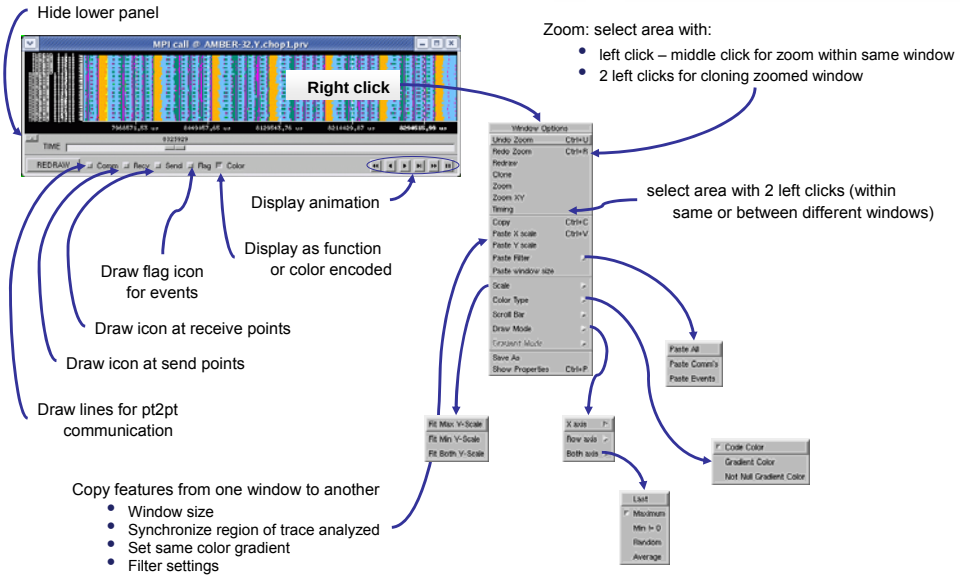
- Color encoding



- Gradient color
  - Light green →Dark blue
- Not null gradient
  - Black for zero value
  - Light green → Dark blue



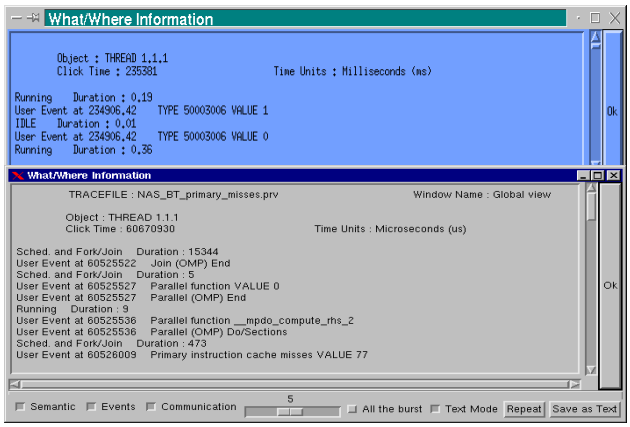
Navigation



9

Textual

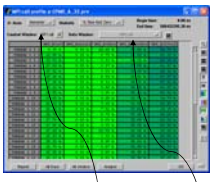
- Appears when clicking on a timeline
- Textual display of
  - Semantic value
  - Events
  - Communications



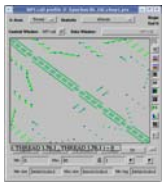
10

Views: Statistics

- 2D
  - Profiles
  - Histograms
  - Correlations
  - Communication Patterns
- 3D



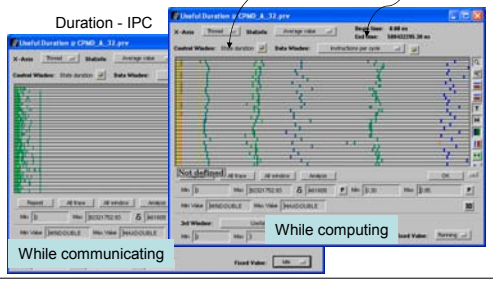
MPI calls profile



Control window

Data window

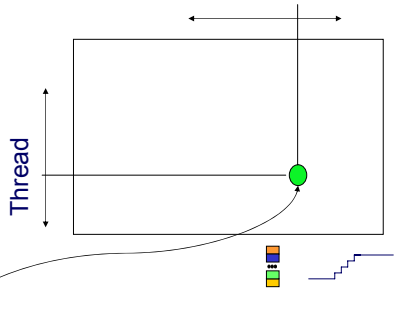
Duration - IPC



How to read profiles

Columns determined by categorical **Control window**

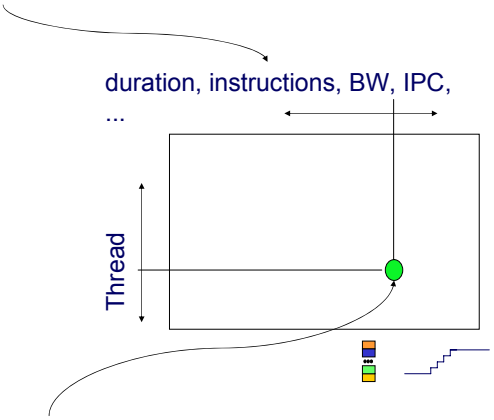
MPI call, user function,…



Value is a statistic (i.e. #occurrences, avg,) computed on **Data window**

How to read histograms

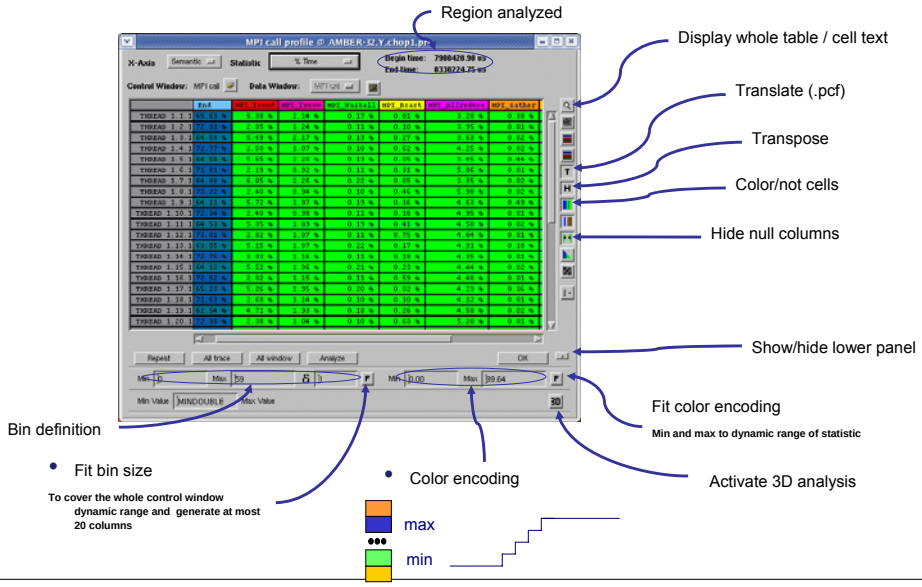
Columns correspond to bins of values of a numeric **Control window**



Value is a statistic (i.e. #occurrences, avg,) computed on **Data window**

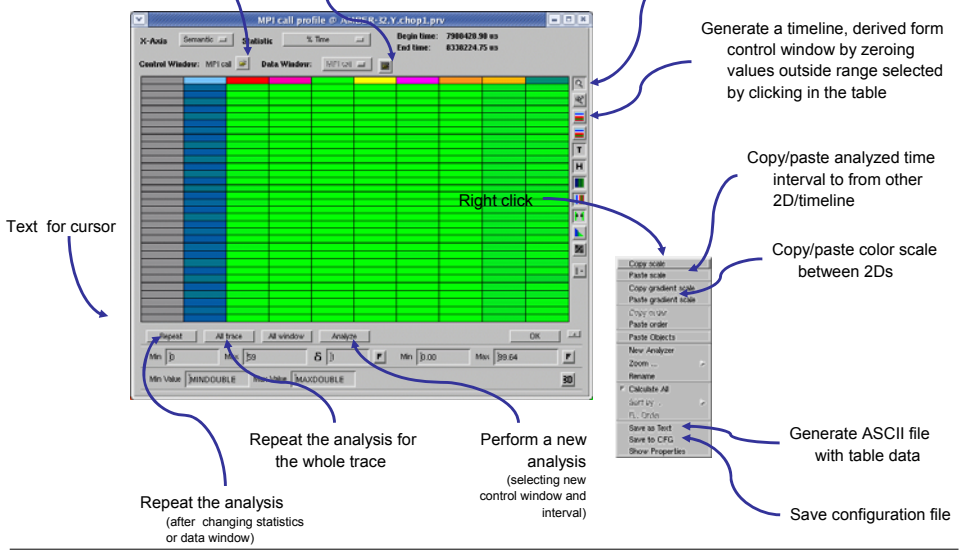
13

2D analysis module



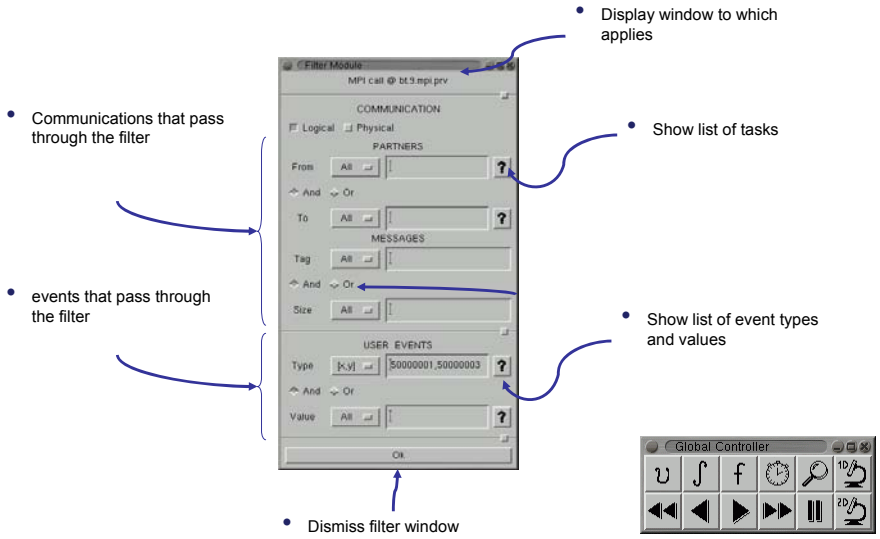
14

2D analysis module



Semantic module internals

Filter module



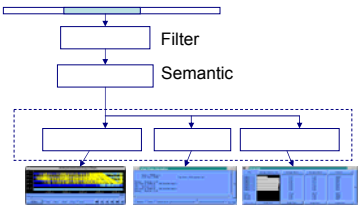
17

Basic functions of time

- The filter module presents a subset of the trace to the semantic module. Each thread is described by
  - A sequence of events $Ev_i, i \in N$, states $St_i, i \in N$ and communications $C_i, i \in N$
  - For each event let $T(Ev_i)$ be its time and $V(Ev_i)$ its value
  - For each state let $T_s(St_i)$ be its start time $T_e(St_i)$ its stop time and $V(St_i)$ its value
  - For each Communication let $T(C_i)$ be its time, $Sz(C_i)$ its size and $Dir(C_i) \in \{send, recv\}$
- Semantic module builds

$$s(t) = S(i), t \in [t_i, t_{i+1}), i \in \mathrm{N}$$

Function of time — Series of values



18

From events to functions

- Different possibilities
  - Last event value $S(i) = V(Ev_i)$
  - Next event value $S(i) = V(Ev_{i+1})$
  - Average Next Event Value $S(i) = \frac{V(Ev_{i+1})}{T(Ev_{i+1}) - T(Ev_i)}$
  - Interval btw. Events $S(i) = T(Ev_{i+1}) - T(Ev_{i+1})$
  - Incoming bytes $S_i = S_{i-1} + InComm_i.size$
  - Bytes btw. events
- Bytes sent/received $S(i) = \sum_j Sz(C_j),\ j \mid T(C_j) \in [T(Ev_i), T(Ev_{i+1}))$

19

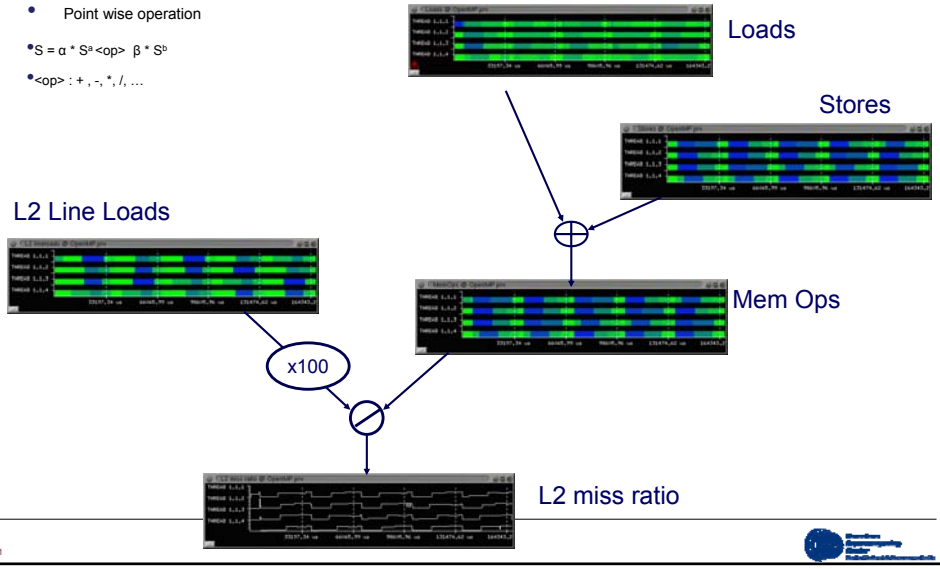
The power of maths: Composition

- S'(t) = f(S(t)) S' = f ° S
  - Sign $S'(t) = sign(S(t))$
  - 1-sign $S'(t) = 1 - sign(S(t))$
  - Select range $S'(t) = S(t) \in [a,b] ? S(t) : 0$
  - Sign ° Is equal $S'(t) = sign(S(t) = a ? S(t) : 0)$
  - Delta $S'(t) = S_{i+1} - S_i$
  - Stacked value

20

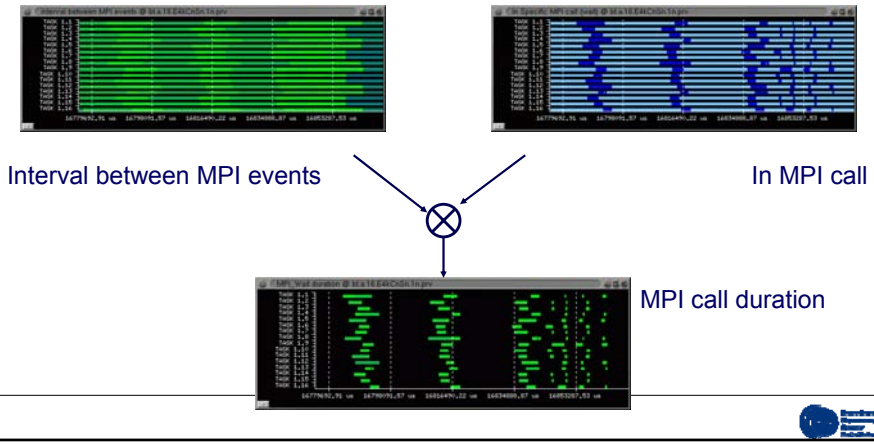
Semantic module

- Derived windows
  - Point wise operation
- S = α * S^a <op> β * S^b
- <op> : + , -, *, /, …



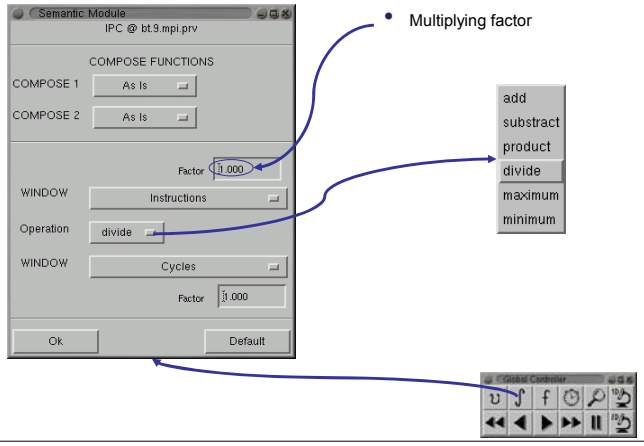
Semantic module

- Derived windows
  - Point wise operation
- S = α * S^a <op> β * S^b
- <op> : + , -, *, /, …

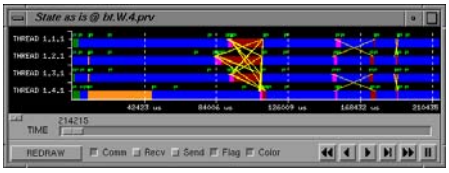


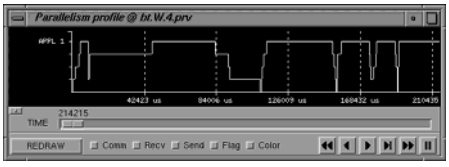
Semantic module: derived windows

- How to build expression

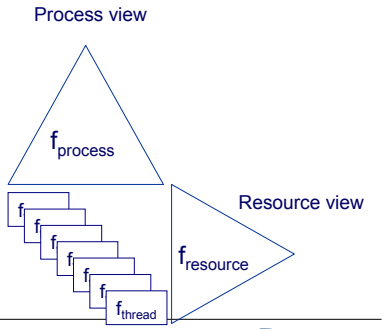


Semantic module





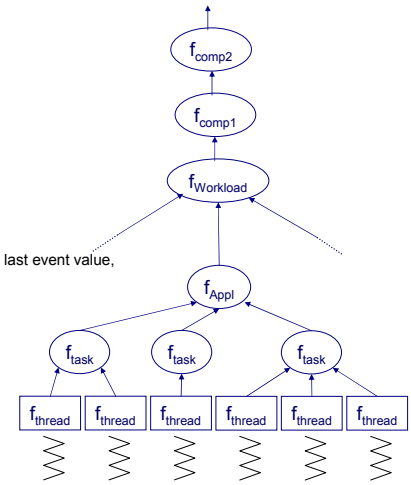
- Angle:
  - Process model
    - Thread, task, application, workload
  - Resource model
    - CPU, node, system



Semantic module: process model view

- Semantic value: S(t)
- $S = f_{comp2} \circ f_{comp1} \circ f_{Workload} \circ f_{Application} \circ f_{task} \circ S_{thread}$
- Semantic functions

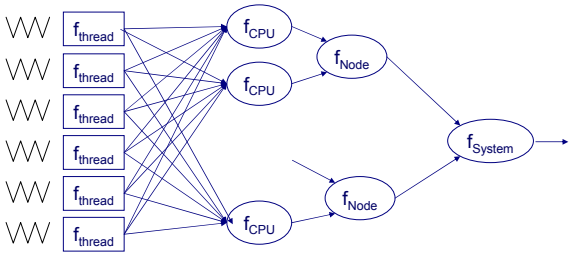
- f_{comp2}, f_{comp1}: sign, mod, div, in range, select range
- $f_{Application}$, $f_{Workload}$: add, average, max, select
- f_{task} : add, average, max, select
- S_{thread}: in state, useful, given state,
-
- next event value,
- average next event value
- interval between events, …



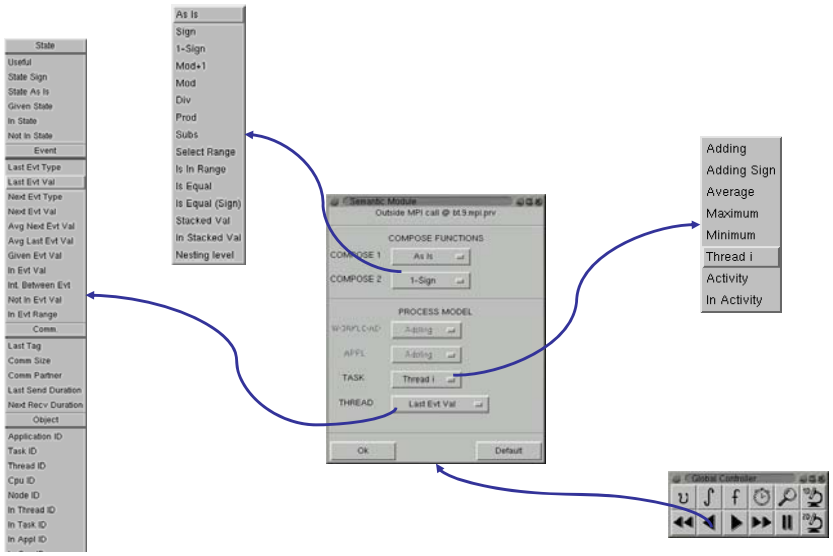
Semantic module: resource view

- $Sf_{resource} = f_{comp2} \circ f_{comp1} \circ f_{System} \circ f_{Node} \circ f_{CPU} \circ S_{thread}$
- Semantic functions

- f_{System} : add, average, max, select
- f_{Node} : add, average, max, select
- f_{CPU} : active thread, select
- S_{thread}: in state, useful, given state, next event value, thread_id



Semantic module: process model view



2D/3D internals

3D/2D analysis module

- Single flexible quantitative analysis mechanism
- Let
  - cw_1 and cw_2 two views we will call control views
  - dw a view we will call data window

For each window w

$$S_{th}^{w}(t) = S_{th}^{w}(i), t \in \left[t_i^w, t_{i+1}^w\right)$$

- For each control window we define a set of bins

$$bin_j^{cw} = \left[range_j^{cw}, range_{j+1}^{cw}\right) \qquad range_{j+1}^{cw} = range_j^{cw} + delta^{cw}$$

- And the discriminator functions

$$\delta_j^{cw}(t) = ((S^{cw}(t) \in bin_j^{cw})?1:0)$$

$$\delta_{j,k}(t) = \delta_j^{cw_1}(t) * \delta_k^{cw_2}(t)$$

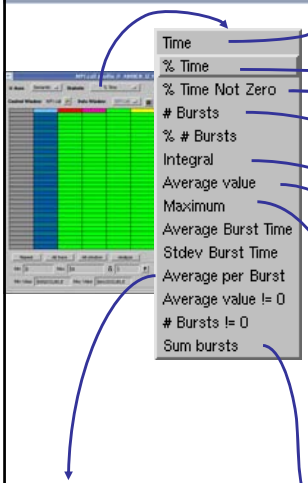
Identify regions with cw's within the (j,k) bin

- The 3D analysis module computes a cube (or plane in the case of 2D) of statistics

$$M(thread, j, k) = statistic(S_{th}^{dw}(t) * \delta_{th,j,k}(t))$$

- Where the statistic can represent the average value, the number of intervals,….

2D analysis module



$$Time(th, j, k) = \int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt$$

$$\%Time(th, j, k) = \frac{\int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt}{t_{end} - t_{start}}$$

$$\%TimeNotZero(th, j, k) = \frac{\int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt}{\sum_j \int_{t_{start}}^{t_{end}} \delta_{tn,j,k}(t)dt}$$

$$NumBurst(th, j, k) = i_{end} - i_{start} + 1$$
$$i_{start} = \min(i) \mid t_i > t_{start}, i_{end} = \max(i) \mid t_i < t_{end}$$

$$Integral(th, j, k) = \int_{t_{start}}^{t_{end}} S_{th}^{dw}(t)\delta_{th,j,k}(t)dt$$

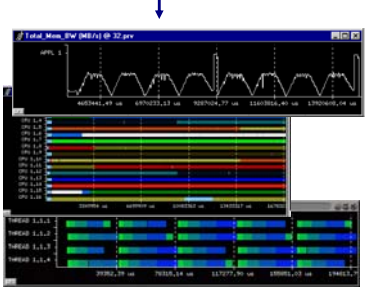
$$Average(th, j, k) = \frac{\int_{t_{start}}^{t_{end}} S_{th}^{dw}(t)\delta_{th,j,k}(t)dt}{\int_{t_{start}}^{t_{end}} \delta_{th,j,k}(t)dt}$$

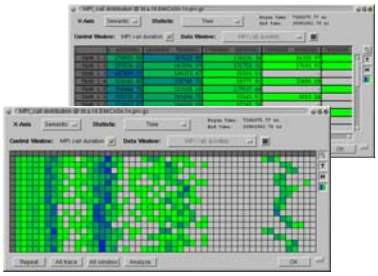
$$Maximum(th, j, k) = \max(S_{th}^{dw}(t)\delta_{th,j,k}(t)), t = \left[t_{start}, t_{end}\right)$$

$$SumBurst(th, j, k) = \sum_{i_{start}}^{i_{end}} S_{th}^{dw}(i)\delta_{th,j,k}(t_i), i_{start} = \min(i) \mid t_i > t_{start}, i_{end} = \max(i) \mid t_i < t_{end}$$

Representation module

Functions of time



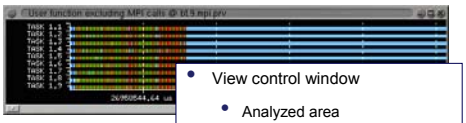


Seeing is believing

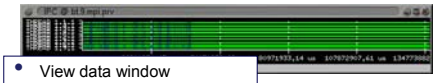
measuring is better

31

2D analysis

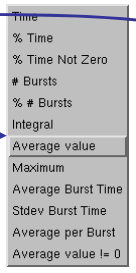


- View control window
  - Analyzed area



- View data window

- Select data window



- Select statistic

- Perform an analysis
  - On same area
    - On whole trace (same CW)
      - On whole CW
        - Selecting new CW

32

2D analysis module