Performance analysis with BSC-Tools

Judit Gimenez, Harald Servat
BSC

Paraver details and internals
Loading configuration files

Basic mode cfgs
When saving analyst specified subset of options that will be made available to users when loading and text describing them.

Detailed mode cfgs
No subset was specified. Full detail available when loading configuration

Always possible to switch from basic to detailed mode

Navigation

Shortcuts:
- Drag and move (D&M) – Zoom all threads
- Control D&M – Zoom XY (a subset of objects and time)
- Shift D&M – Timing
### Table Information and Control

#### Create a New Table

#### Display Whole Table / Cell Text

#### Color/Not Cells

#### Transpose

#### Hide Null Columns

#### Region Analyzed

#### Bin Definition

#### Statistic (Cell)

#### Change Data Window

#### Activate 3D Analysis

#### Color Encoding

<table>
<thead>
<tr>
<th>max</th>
<th>min</th>
<th>NULL entry</th>
</tr>
</thead>
</table>

#### Table Information and Control

#### Open Data Window

#### Open Control Window

#### Open 3D Window

#### Generate a Timeline, Derived from Control Window with the Range of Values Selected Clicking in the Table (Zoom Mode Only)

#### Right Click

#### Generate ASCII File with Table Data

#### Shortcuts (Zoom Mode Only):
- Drag and Move (D&M) – Zoom
- Control D&M – Zoom XY
**3D tables**

One additional dimension
One plane per value of a 3D control window
Useful to categorize histograms
i.e. histogram of duration of specific user function

![3D control window: determines planes](image1)

![Actual Plane on display](image2)

**Handling very large traces**

Paraver data handling utilities
If trying to load a very large trace, Paraver will ask if you want to filter it

Three steps:
Filter original trace discarding most of the records only keeping most relevant information (typically computation bursts longer than a given lower bound)

Analyze coarse grain structure of trace. Typically useful_duration.cfg

Cut original trace to obtain a fully detailed trace for the time interval considered representative or of interest
Filtering very large traces

Trace to which it will be applied
A trace with basename.filter1.prv will be generated

Setting can be saved/restored to/from an xml file

Select filtering option

Discard events and communications

Keep only Running bursts ...

A clean/representative interval can be identified longer than 3000 ns

Analyze coarse grain structure

Filtered trace IS a Paraver trace

Can be analyzed with standard cfgs as long as the information they require is still in the trace
A typical view that shows a lot of the structure of a trace is useful_duration.cfg
Repetitive structure is often apparent
Perturbations can also be typically identified
A clean/representative interval can be identified
Cutting very large traces

Load a filtered trace and use the scissors tool

- **Scissors tool**
- **Browse to select file from which the cut will be obtained**
- **Select cutter**
- **Click to select region**
- **Select time interval by clicking left and right limits in a window of the filtered trace previously loaded**
- **Recommended cuts within long computation bursts**

Filter module

- **Communications that pass through the filter**
- **Events that pass through the filter**
- **Show list of event types**
Semantic module: Control

**Derived windows**

**Point wise operation**

\[ S = \alpha \cdot S^a \cdot <op> \cdot \beta \cdot S^b \]

\(<op> : +, -, \ast, /, \ldots>\)

**L2 Line Loads**

\[ \text{x100} \]

**Loads**

**Stores**

**Mem Ops**

**L2 miss ratio**
Semantic module

Derived windows

Point wise operation

\[ S = \alpha \cdot S^a \wedge \beta \cdot S^b \]

<op> : +, -, *, /, ...

Interval between MPI events

In MPI call

MPI call duration

Semantic module: derived windows

How to build expression

Derived metric

Multiplying factor

Derived metric

.add product

.multiply
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, last event value, next event value, average next event value, interval between events, ...

Semantic module: resource view

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

- 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
**Dimemas details**

**Dimemas trace generation**

**Paraver → Dimemas trace Generation**

prv2trf original.prv dimemas.trf

Default: duration of each computation region taken from .prv computation duration

```
Usage:
prv2trf -i <iprobe_miss_threshold> -b <hw_counter_type>,<factor> <paraver_trace> <dimemas_trace>

-h                           This help
-i <iprobe_miss_threshold>   No generate initial idle states
-b <hw_counter_type>,<factor> Maximum MPI_Iprobe misses to discard Iprobe area CPU burst

```

Force synchronized start of all threads

Computation region duration derived from hardware counters assuming/modeling a given performance (<factor>)
Collective Communication Model

- Per call model
- Model factor
  - Lin
  - Log
  - Const
- Size of message
  - Min over all processes
  - Mean over all processes
  - Max over all processes
- Specified in input file

2D analysis module

\[ Time_{th,j,k} = \int_{t_{start}}^{t_{end}} \delta_{h,j,k}(t) dt \]
\[ \%Time_{th,j,k} = \frac{\int_{t_{start}}^{t_{end}} \delta_{h,j,k}(t) dt}{t_{end} - t_{start}} \]
\[ \%TimeNotZero_{th,j,k} = \frac{\sum_{i=1}^{N_{burst}} \delta_{h,j,k}(t) dt}{t_{end} - t_{start}} \]
\[ NumBursts_{th,j,k} = t_{end} - t_{start} + 1 \]
\[ t_{start} = \min(t_i) \]
\[ t_{end} = \max(t_i) \]
\[ Max_{th,j,k} = \max(S_{h,j,k}(t))^\prime \]
\[ Min_{th,j,k} = \min(S_{h,j,k}(t))^\prime \]
\[ Integral_{th,j,k} = \sum_{i=1}^{N_{burst}} S_{h,j,k}(t) \delta_{h,j,k}(t) dt \]
\[ Time_{th,j,k} = \frac{\sum_{i=1}^{N_{burst}} S_{h,j,k}(t) \delta_{h,j,k}(t) dt}{\sum_{i=1}^{N_{burst}} \delta_{h,j,k}(t) dt} \]