Scalability of Trace Based Tools

Jesus Labarta
Barcelona Supercomputing Center

What is Scalability?
Index

- General view
- Scalability of instrumentation and preprocessing
- Scalability of display
- Scalability, models and automatic analysis
- Summary

Scalability

- Amount of data vs information
- Dynamic range

$10^6$
Performance analysis universe

- Acquisition
- Profile land
- Presentation
- Sampling land
- Timeline land
- Instrumentation land

Variance (space & time) is important
Information is in the details
We do not know what we are facing
Capture a lot of data
Use a flexible browser

CEPBA-tools towards scalability

Selection
- On-Off (time, processors, space)
- External control file
- Events/information emitted (i.e., MPI, HWC)
- Limit buffer sizes / duration
- Structure detection (i.e., periodicity)
- Circular buffer (issues: matching, density)
- Min. duration states
- Software counters (MPI_Probe, #MPIs, size)

Parallel merge

Software counters
- Count original events
- Accumulate values (hwc)
- When: periodic, condition

Subset selection
- Time, processors
- Trace size limit
- States/comms/events

Manual filters/GUI
Automatic

Same ideas applicable at instrumentation, postprocessing & analysis

Functionality
- Non linear
- Composition
- Aggregation

Display
- Non linear render
- What & color
- Generic subset of objects

Performance
- Trace loading
- Metric comp. (intervals)
- OpenMP, Distributed
Scalability of instrumentation and preprocessing

<table>
<thead>
<tr>
<th>Scalability of instrumentation / preprocessing</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Selection of what goes into the trace to still capture behavior and variance</td>
</tr>
<tr>
<td>• Dimensions</td>
</tr>
<tr>
<td>• Time/space</td>
</tr>
<tr>
<td>• Events</td>
</tr>
<tr>
<td>• Satisfying a certain property (i.e. long computation bursts)</td>
</tr>
<tr>
<td>• Specification</td>
</tr>
<tr>
<td>• Manually specified (i.e. trace control xml)</td>
</tr>
<tr>
<td>• Automatic structure detection (i.e. signal processing, clustering)</td>
</tr>
<tr>
<td>• Summarization of non selected events: Software counters</td>
</tr>
<tr>
<td>• Aggregation of values (hwc)</td>
</tr>
<tr>
<td>• Counts (#MPI calls)</td>
</tr>
<tr>
<td>• Distributed implementation</td>
</tr>
</tbody>
</table>
Distributed trace control

- MRNET based mechanism
  - Local instrumentation on a circular buffer
  - Periodic MRNet front-end initiation of collection process
- Local algorithm
- Reduction on tree
- Selection at root propagated
- Locally emit trace events

Algorithm
- Collective duration threshold

Scalability of display
Scalability of Presentation

- **Aggregation**
  - Functional rather than scalability motivation

- **Display**
  - Non linear render
    - Value for pixel
    - Colors

- **Objects**
  - Any subset

Scalability of display

- **Non Linear Render**

CG.C
1024 CPUs
MPI call
Display and navigation

Linpack @ 10000 CPUs
3000 seconds
500MB trace

Impact of contention?
Display and navigation

- Multiple single metric views. Visual correlation
- Flexibility to generate metrics. Non linear
- Zooming and synchronization capabilities
- Non linear 2D rendering
- Non linear coloring. “few” levels. Scale tuning

Interoperation between analysis and display

- AMBER @ 512 procs.
- Which is the longest MPI call? Why?
  - Histogram of duration of MPI calls
  - Duration of longest calls
  - Who sends to 125 in selected area
  - Useful duration
  - Dynamic range = being able to find needles in a haystack
Scalability, models and automatic analysis

Models are key for performance analysis
- Reference for observed metrics
- Identify key factors that explain behavior

Common bad practices
- We very seldom use them
- We use them one way:
  - estimate/prediction
  - Seldom for parameter fitting
- Obsessed by detail instead of modeling just key factors.
- Obsessed by accuracy of prediction instead of properly capturing trends

Models:
- Bounds: nominal/microbenchmarks (MFLOPS, MIPS, IPC, MB/s,...)
- Analytic: parameters obtained from microbenchmarks and (manual) complexity analysis
- Simulators of different levels of detail to estimate reasonably achievable performance

Hierarchical convolution

\[ L2\_miss\_latency = \frac{\#cycles\_instr}{\#L2misses \times idealIPC} \]
Automatic structure detection

- Automatizable through signal processing techniques
- Basis for analysis techniques
## Methodology: Automatic analysis

- **Automatizable through signal processing techniques**
- **Basis for analysis techniques**

### Clustering techniques

![Clustering techniques diagram](image)

### Data Analysis

<table>
<thead>
<tr>
<th>Region</th>
<th>IPC</th>
<th>L1D hits per 1000</th>
<th>L1D misses per 1000</th>
<th>L2 D hits per 1000</th>
<th>L2 D misses per 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.57</td>
<td>2.34</td>
<td>0.01</td>
<td>75.55</td>
<td>0.30</td>
</tr>
<tr>
<td>2</td>
<td>0.54</td>
<td>0.48</td>
<td>0.05</td>
<td>52.6</td>
<td>0.06</td>
</tr>
<tr>
<td>3</td>
<td>0.53</td>
<td>1.18</td>
<td>0.14</td>
<td>47.64</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>0.62</td>
<td>0.38</td>
<td>0.04</td>
<td>43.27</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.42</td>
<td>1.56</td>
<td>0.18</td>
<td>43.84</td>
<td>0.20</td>
</tr>
</tbody>
</table>

### System Performance

- **WRF-NMM Peninsula 4km 128 procs**
  - 570 s
  - 2.2 GB
  - MPI, HWC

- **WRF-NMM Peninsula 4km 256 procs**
  - 570 s
  - 5 MB
  - 4.6 s

- **WRF-NMM Peninsula 4km 512 procs**
  - 570 s
  - 36.5 MB
  - 36.5 MB
Scaling model: based on measurements

\[
Sup = \frac{P}{P_0} \cdot \frac{LB}{LB_0} \cdot \frac{CommEff}{CommEff_0} \cdot \frac{IPC}{IPC_0} \cdot \frac{\#instr}{\#instr_0}
\]

\[
T
\]

\[
\sum_{i}^{n} eff_i = \frac{T_i}{T}
\]

\[
CommEff = \max(\text{eff}_i)
\]

\[
LB = \frac{\sum_{i}^{n} eff_i}{P \times \max(\text{eff}_i)}
\]

Scaling model: adding simulation capabilities

\[
Sup = \frac{P}{P_0} \cdot \frac{\text{macroLB}}{\text{macroLB}_0} \cdot \frac{\text{microLB}}{\text{microLB}_0} \cdot \frac{CommEff}{CommEff_0} \cdot \frac{IPC}{IPC_0} \cdot \frac{\#instr}{\#instr_0}
\]

Migrating/local load imbalance

Serialization
Scaling model

<table>
<thead>
<tr>
<th></th>
<th>WRF-NMM-Iberia</th>
<th>WRF-ARW-Iberia</th>
</tr>
</thead>
<tbody>
<tr>
<td>实验结果</td>
<td>128 process</td>
<td>256 process</td>
</tr>
<tr>
<td>RealCommEff</td>
<td>Micro LB</td>
<td>Micro LB</td>
</tr>
<tr>
<td>IPC/IPC_0</td>
<td>1/Complexity</td>
<td>1/Complexity</td>
</tr>
<tr>
<td>Model Speedup</td>
<td>Real speedup</td>
<td>Real speedup</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>WRF-NMM-Europe</th>
<th>WRF-ARW-Europe</th>
</tr>
</thead>
<tbody>
<tr>
<td>实验结果</td>
<td>128 process</td>
<td>256 process</td>
</tr>
<tr>
<td>RealCommEff</td>
<td>Micro LB</td>
<td>Micro LB</td>
</tr>
<tr>
<td>IPC/IPC_0</td>
<td>1/Complexity</td>
<td>1/Complexity</td>
</tr>
<tr>
<td>Model Speedup</td>
<td>Real speedup</td>
<td>Real speedup</td>
</tr>
</tbody>
</table>

Modular, interoperable infrastructure

- Paraver
- DIMEMAS
- FSIM...
- Venus (ZRL)
- PeekPerf
- Data Display Tools
- .prv.
- .pcf.
- .viz.
- .txt.
- .trf.
- Machine description
- Time analysis, filters
- Contention analysis environment
- Instr. Level Simulators
- Stack Can
- OpenDX
- MRNET
- XML control
- Content analysis
- Cube
- Data Display Tools
Summary

Scalability

- Mechanisms
  - Separation of engine and display
  - Distributed implementation
  - Data encoding
  - Subset selection
  - Non linear rendering
    - Logical pixels (2x2, 3x3, M x N?)
    - Software counters

- Algorithms
  - Techniques to process the raw data
  - Signal processing, clustering,…
  - Metrics
    - Should be useful
    - Understandable by “mortals”
    - Lead to right decision making (i.e. Computation vs MPI)
  - Based on models

Emphasis
necessary, not sufficient

Importance
intelligence
automatic
Thesis

“A single instrumented run captures a lot of information that is essentially thrown away in current parallel programming practice”

“It is possible to squeeze the information in the trace”

An analogy

Use of traces

Huge probe effect

Team work

Multidisciplinary

Correlate different sources

Speculate till arriving to consistent theory