Understanding applications with Paraver

Judit Gimenez
judit@bsc.es

NERSC - Berkeley, August 2016
Humans are visual creatures

Films or books?
- Two hours vs. days (months)

Memorizing a deck of playing cards
- Each card translated to an image (person, action, location)

Our brain loves pattern recognition
- What do you see on the pictures?
Our Tools

- Since 1991
- Based on traces
- Open Source
  - [http://www.bsc.es/paraver](http://www.bsc.es/paraver)

Core tools:
- Paraver (paramedir) – offline trace analysis
- Dimemas – message passing simulator
- Extrae – instrumentation

Focus
- Detail, variability, flexibility
- Behavioral structure vs. syntactic structure
- Intelligence: Performance Analytics
**Paraver – Performance data browser**

**Trace visualization/analysis**

+ trace manipulation

**Timelines**

**2/3D tables**

(Statistics)

**Goal = Flexibility**

No semantics
Programmable

**Comparative analyses**

Multiple traces
Synchronize scales
Timelines

Each window displays one view

- **Piecewise constant** function of time

Types of functions

- Categorical
  - State, user function, outlined routine

- Logical
  - In specific user function, In MPI call, In long MPI call

- Numerical
  - IPC, L2 miss ratio, Duration of MPI call, duration of computation burst

\[ s(t) = S_i, i \in [t_i, t_{i+1}) \]

\[ S_i \in [0, n] \subset N, \quad n < \]

\[ S_i \in \{0, 1\} \]

\[ S_i \in R \]
From timelines to tables

- MPI calls
- Useful Duration
- Histogram Useful Duration
Analyzing variability through histograms and timelines

Useful Duration

IPC

Instructions

L2 miss ratio
By the way: six months later ....
Variability … is everywhere

- CESM: 16 processes, 2 simulated days

- Histogram useful computation duration shows high variability

- How is it distributed?

- Dynamic imbalance
  - In space and time
  - Day and night.
  - Season ? 😊
Trace manipulation

Data handling/summarization capability

- Filtering
  - Subset of records in original trace
  - By duration, type, value,…
  - Filtered trace IS a paraver trace and can be analysed with the same cfgs (as long as needed data kept)

- Cutting
  - All records in a given time interval
  - Only some processes

- Software counters
  - Summarized values computed from those in the original trace emitted as new even types
  - #MPI calls, total hardware count,…
Dimemas: Coarse grain, Trace driven simulation

- **Simulation:** Highly non linear model
  - MPI protocols, resource contention...

- **Parametric sweeps**
  - On abstract architectures
  - On application computational regions

- **What if analysis**
  - Ideal machine (instantaneous network)
  - Estimating impact of ports to MPI+OpenMP/CUDA/…
  - Should I use asynchronous communications?
  - Are all parts equally sensitive to network?

- **MPI sanity check**
  - Modeling nominal

- **Paraver – Dimemas tandem**
  - Analysis and prediction
  - What-if from selected time window

Detailed feedback on simulation (trace)
Network sensitivity

MPIRE 32 tasks, no network contention

$L = 5\mu s - BW = 1 \text{ GB/s}$

$L = 1000\mu s - BW = 1 \text{ GB/s}$

$L = 5\mu s - BW = 100\text{MB/s}$

All windows same scale
Would I benefit from asynchronous communications?

**SPECFEM3D**

Real

Ideal

Prediction

MN

Prediction

100MB/s

Prediction

10MB/s

Prediction

5MB/s

Prediction

1MB/s

Courtesy Dimitri Komatitsch
Ideal machine

The impossible machine: \( \text{BW} = \infty, \quad L = 0 \)

Actually describes/characterizes Intrinsic application behavior

- Load balance problems?
- Dependence problems?

Impact on practical machines?
Ideal speeding up ALL the computation bursts by the CPU ratio factor

- The more processes the less speedup (higher impact of bandwidth limitations)!!

![Graphs showing impact of architectural parameters on speedup](image-url)
Hybrid parallelization

Hybrid/accelerator parallelization

- Speed-up SELECTED regions by the CPU ratio factor

(Previous slide: speedups up to 100x)
Models and Extrapolation
Parallel efficiency model

Parallel efficiency = LB eff * Comm eff
Parallel efficiency refinement: $\text{LB} \cdot \mu\text{LB} \cdot \text{Transfer}$

**Serializations / dependences ($\mu\text{LB}$)**

**Dimemas ideal network $\rightarrow$ Transfer (efficiency) = 1**
Why scaling?

\[ \eta_{\parallel} = LB \times Ser \times Trf \]

CG-POP mpi2s1D - 180x120

Good scalability !! Should we be happy?

\[ \eta_{\parallel} = \eta_{\parallel} \times \eta_{\text{instr}} \times \eta_{\text{IPC}} \]
AVBP (strong scale)

Input: runs 512 – 4096

\[ Amdahl_{fit} = \frac{\text{metric}_0}{f_{\text{metric}} + (1 - f_{\text{metric}}) \times P} \]

Strong scale → small computations between MPI calls become more and more important
TURBORB (strong scale)

**Input:** runs 512 – 4096

Network contention can be reduced balancing / limiting the random selection within a node.

**Equation:**

\[
Amdahl_{fit} = \frac{\text{metric}_0}{f_{\text{metric}} + (1 - f_{\text{metric}}) \times P^{1/3}}
\]
Clustering
Using Clustering to identify structure

Automatic Detection of Parallel Applications Computation Phases (IPDPS 2009)
What should I improve?

What if ....

... we increase the IPC of Cluster 1?

... we balance Clusters 1 & 2?

13% gain

19% gain
Tracking scalability through clustering

- OpenMX (strong scale from 64 to 512 tasks)
Folding
Folding: Detailed metrics evolution

• Performance of a sequential region = 2000 MIPS

  Is it good enough?

  Is it easy to improve?
Folding: Instantaneous CPI stack

- Trivial fix (loop interchange)
- Easy to locate?
- Next step?
- Availability of CPI stack models for production processors?
  - Provided by manufacturers?
Instantaneous metrics with minimum overhead

- Combine instrumentation and sampling
  - Instrumentation delimits regions (routines, loops, …)
  - Sampling exposes progression within a region
- Captures performance counters and call-stack references
“Blind” optimization

From folded samples of a few levels to timeline structure of “relevant” routines

Recommendation without access to source code

CG-POP multicore MN3 study

- Unbalanced MPI application
  - Same code
  - Different duration
  - Different performance
Address space references

Based on PEBS events

- Loads: address, cost in cycles, level providing the data
- Stores: only address

Modified Stream

```c
for (i = 0; i < NITERS; i++) {
    Extrae_event (BEGIN);
    for (j = 0; j < N; j++)
        c[j] = a[j];
    for (j = 0; j < N/8; j++)
        b[j] = s*c[random()]%j;
    for (j = 0; j < N; j++)
        c[j] = a[j]+b[j];
    a[j] = b[j]+s*c[j];
    Extrae_event (END);
}
```

![Graph showing memory access patterns and timing](image)
Lulesh memory access

Architecture impact

Stalls distribution
Methodology
Performance analysis tools objective

Help generate hypotheses

Help validate hypotheses

Qualitatively

Quantitatively
First steps

Parallel efficiency – percentage of time invested on computation
- Identify sources for “inefficiency”:
  - load balance
  - Communication /synchronization

Serial efficiency – how far from peak performance?
- IPC, correlate with other counters

Scalability – code replication?
- Total #instructions

Behavioral structure? Variability?

Paraver Tutorial: Introduction to Paraver and Dimemas methodology
BSC Tools web site

www.bsc.es/paraver

- downloads
  - Sources / Binaries
  - Linux / windows / MAC

- documentation
  - Training guides
  - Tutorial slides

Getting started

- Start wxparaver
- Help → tutorials and follow instructions
- Follow training guides
  - Paraver introduction (MPI): Navigation and basic understanding of Paraver operation
Thanks!

Use your brain, use visual tools :)  
Look at your codes!

www.bsc.es/paraver
Demo
Some examples of efficiencies

<table>
<thead>
<tr>
<th>Code</th>
<th>Parallel efficiency</th>
<th>Communication efficiency</th>
<th>Load Balance efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gromacs@mt</td>
<td>66.77</td>
<td>75.68</td>
<td>88.22</td>
</tr>
<tr>
<td>BigDFT@altamira</td>
<td>59.64</td>
<td>78.97</td>
<td>75.52</td>
</tr>
<tr>
<td>CG-POP@mt</td>
<td>80.98</td>
<td>98.92</td>
<td>81.86</td>
</tr>
<tr>
<td>ntchem_min@pi</td>
<td>92.56</td>
<td>94.94</td>
<td>97.49</td>
</tr>
<tr>
<td>nicam@pi</td>
<td>87.10</td>
<td>75.97</td>
<td>89.22</td>
</tr>
<tr>
<td>cp2k@jureca</td>
<td>75.34</td>
<td>81.07</td>
<td>92.93</td>
</tr>
<tr>
<td>lulesh@mn3</td>
<td>90.55</td>
<td>99.22</td>
<td>91.26</td>
</tr>
<tr>
<td>lulesh@leftraru</td>
<td>69.15</td>
<td>99.12</td>
<td>69.76</td>
</tr>
<tr>
<td>lulesh@uv2 (mpt)</td>
<td>70.55</td>
<td>96.56</td>
<td>73.06</td>
</tr>
<tr>
<td>lulesh@uv2 (impi)</td>
<td>85.65</td>
<td>95.09</td>
<td>90.07</td>
</tr>
<tr>
<td>lulesh@mt</td>
<td>83.68</td>
<td>95.48</td>
<td>87.64</td>
</tr>
<tr>
<td>lulesh@cori</td>
<td>90.92</td>
<td>98.59</td>
<td>92.20</td>
</tr>
</tbody>
</table>