A user perspective on energy profiling tools in HPC environments

Fahimeh Alizadeh Moghaddam1,2, Thomas Geenen3, Patricia Lago2, Paola Grosso1
1 System and Network engineering group, University of Amsterdam
2 Software and Services group, VU University Amsterdam
3 SURFsara, Amsterdam, The Netherlands

Introduction

In order to manage the power consumption in HPC environments, the running applications can be profiled using specific tools. The choice for a tool depends on the final purpose of users for the information retrieved. We categorize two types of users:
• The generic user: interested in summary results
• The software developer: interested in behavioral-related energy information.

Research Question

• Under which circumstances should a user choose for an energy profiling tool?
• What will be the consequences in terms of accuracy and overhead of this choice?

Infrastructure Setup

• We use the Cartesius system at SURFsara.
• We run HPC Challenge benchmark as our experimental application.
• SLURM and Score-P are two available tools in the infrastructure.
• The PAPI/RAPI software power model is supported by both tools.

Conclusion

Both tools provide required information to both types of users but they differ in granularity and accuracy:

The generic user:
+ SLURM provides accurate summary info
+ fine-grained summary data from Score-P
- very coarse-grained data from SLURM
- Inaccurate reported data by Score-P

The software developer:
+ time series of power consumption using SLURM
+ total power consumption of function calls using Score-P
- No time series provided by the visualization tool of Score-P (CUBE)

Experiment 1: Collecting power measurements directly by the PAPI library using the rapl_plot application.

- 4 sample rates are considered, from every 1 second (top plot) to every 1000 microseconds (bottom plot)
- As the sample rate decreases, more details are missing from the plots.

Experiment 2: Collecting energy data through SLURM

- The measurements from Rapl_plot and SLURM are almost identical.
- SLURM shows small delays to the sudden changes in power consumption.

Experiment 3: Collecting energy data through Score-P

<table>
<thead>
<tr>
<th>#</th>
<th>Test</th>
<th>Power Consumption Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HPL</td>
<td>6%</td>
</tr>
<tr>
<td>2</td>
<td>DGEMM</td>
<td>Single: 22.6% Star: 47.3%</td>
</tr>
<tr>
<td>3</td>
<td>FFT</td>
<td>MPI: 13.2% Star: 31.7%</td>
</tr>
<tr>
<td>4</td>
<td>LatencyBandwidth</td>
<td>Single: 34.9% Star: 28.9%</td>
</tr>
<tr>
<td>5</td>
<td>PTRANS</td>
<td>29.5%</td>
</tr>
<tr>
<td>6</td>
<td>RandomAccess</td>
<td>MPI: 1.7% Star: 59.0%</td>
</tr>
<tr>
<td>7</td>
<td>RandomAccess.LCG</td>
<td>MPI: 1.9% Star: 46.1%</td>
</tr>
<tr>
<td>8</td>
<td>STREAM</td>
<td>Single: -23.3% Star: 49.2%</td>
</tr>
</tbody>
</table>

- Score-P introduces varying amount of overhead for different application runs.
- MPI code path performs with the least measurement overhead compared to Single and Star variants.

This work has been sponsored by the European Fund for Regional Development under project MRA Cluster Green Software.