Writing and Reading a Reproducible Article

Luka Stanisic and Arnaud Legrand

MESCAL team, LIG, Univ. of Grenoble

COMPAS, 22.04.2014
Context

- HPC applications nowadays use both multi-core CPUs and GPUs
- Managing efficiently computation execution and data transfer is extremely complex
- Need for portable performance $\leadsto$ Runtime system
HPC applications nowadays use both multi-core CPUs and GPUs. Managing efficiently computation execution and data transfer is extremely complex. Need for portable performance; Runtime system.

Many configuration parameters:
1. Task granularity
2. Scheduling strategies
3. Application structure

Emerging challenges:
1. Finding optimal combination of parameters for a given machine
2. Evaluate configurations on a wide variety of platforms
3. Quickly identify performance issues (e.g., bottlenecks)

Possible solution: Simulation
Our proposal

**StarPU**

Dynamic runtime for hybrid architectures. StarPU execution consists in scheduling a graph of tasks with data dependencies on the different computing resources.

**Simgrid**

Versatile simulator for distributed systems

Implementation:

- StarPU applications and runtime are **emulated**
- All operations related to thread synchronization, actual computations and data transfer are **simulated**
- Control part of StarPU is modified to dynamically inject computation and communication tasks into the simulator
- StarPU calibration and platform description is used by Simgrid
Results

<table>
<thead>
<tr>
<th></th>
<th>Cholesky</th>
<th>LU</th>
</tr>
</thead>
<tbody>
<tr>
<td>hannibal: 3 QuadroFX5800</td>
<td><img src="image1" alt="Graph" /></td>
<td><img src="image2" alt="Graph" /></td>
</tr>
<tr>
<td>attila: 3 TeslaC2050</td>
<td><img src="image3" alt="Graph" /></td>
<td><img src="image4" alt="Graph" /></td>
</tr>
<tr>
<td>conan: 3 TeslaM2075</td>
<td><img src="image5" alt="Graph" /></td>
<td><img src="image6" alt="Graph" /></td>
</tr>
<tr>
<td>frogkepler: 2 K20</td>
<td><img src="image7" alt="Graph" /></td>
<td><img src="image8" alt="Graph" /></td>
</tr>
</tbody>
</table>

Matrix dimension vs. GFlop/s for different conditions:

Experimental Condition:
- SimGrid
- Native
Results

Comparing traces

Native

CPU0
CPU1
CPU2
CPU3
GPU1
GPU2
GPU3

Simgrid

CPU0
CPU1
CPU2
CPU3
GPU1
GPU2
GPU3

State

- POTRF
- TRSM
- GEMM
- DriverCopy
- FetchingInput

Time [ms]

0
25000
50000
75000
• Works fine *now*, but coming to this point was not easy

• We had to do many iterations of:
  1. Running *complex beta* code on several not always *dedicated* machines
  2. Comparing with simulations, *debugging*, *understanding*, *remodeling* and going back to step 1 until not satisfied

• With good results, we decided to make a reproducible article
  1. From outside it looks like any other pdf paper
  2. From inside . . .

I will try to convince you that our article is not only *reproducible* but also *readable* and *understandable*!

http://dx.doi.org/10.6084/m9.figshare.928338
Reproducible Research
Requires a daily usage of a labbook (org, git/svn, ...)