Power/Energy consumption in Fault Tolerance protocols

M. Diouri, O. Gluck, L. Lefevre, F. Cappello

Rennes (France), June 13th, 2012

mehdi.diouri@ens-lyon.fr

7th Workshop of the Joint Laboratory for Petascale Computing.
An important growth of performance: a factor of 1000/10 years.

The most powerful supercomputer is K-Computer (Top500): more than 700,000 cores and able to perform 10 PFlop/s.

A wide range of scientific applications:

IESP (USA), EESI (Europe): roadmaps to Exascale in 2018.
Context and Motivations

The issues addressed at the Exascale:

- Power and energy consumption
  - Most energy efficient: IBM BlueGene/Q\(^1\): 2 GFlops/W.
  - DARPA target: 20 MW for a 1 EFlop: 50 GFlops/W.

- Fault tolerance
  An exascale system = millions of cores.
  Faults many times per day.
  Fault tolerance is mandatory.

\(^1\)Green 500: www.green500.org
Current Fault tolerance protocols

Three main categories of protocols: uncoordinated, coordinated, hierarchical protocols.

Rely on checkpointing/restart:

- with message logging in uncoordinated protocols
- with process synchronization in coordinated protocols.

In hierarchical protocols: processes organized in clusters.

- process synchronization inside a same cluster.
- message logging between clusters.
Motivations

Both the issues of fault tolerance and power consumption are interrelated.

What are the power and energy consumption of current fault tolerance protocols?

What is the best fault tolerance protocol in terms of power/energy consumption?
Methodology

Both the issues of fault tolerance and power consumption are interrelated.

What are the power and energy consumption of current fault tolerance protocols?

===> Experiments: benchmarks to study the energy behavior of the fault tolerance protocols.

===> 3 operations: Checkpointing, Message logging, Process coordination.

What is the best fault tolerance protocol in terms of power/energy consumption?

===> Comparison of the energy consumption of fault tolerance operations during real applications (NAS).
Outline

1. Introduction
2. Experimental infrastructure
3. Energy in fault tolerance protocols
4. Energy-aware choice of fault tolerance protocols
5. Conclusion
Experimental infrastructure

Experiments on the Lyon site of Grid5000: a French scientific platform geographically distributed over 10 sites in France.

The Lyon site offers 64 available identical nodes Sun Fire V20z.

Each node gathers:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2 AMD Opteron</td>
</tr>
<tr>
<td></td>
<td>2.4 GHz, 1 core each</td>
</tr>
<tr>
<td>Memory</td>
<td>2 GB</td>
</tr>
<tr>
<td>Network</td>
<td>Gigabit Ethernet</td>
</tr>
<tr>
<td>HDD</td>
<td>SCSI, 73 GB</td>
</tr>
</tbody>
</table>
An energy-sensing infrastructure of external power meters from Omegawatt \(^2\).

- instantaneous consumption in Watts;
- at each second for each monitored node;
- with a precision of 0.125 Watts.

We used only one core per node in all our experiments.
We ran each experiment 20 times.
We computed the mean value over the 20 values.

\(^2\)http://www.omegawatt.fr/gb/index.php
Energy in fault tolerance protocols - Checkpointing

Checkpointing: storing a snapshot image of the current application state.

From the Berkeley Lab Checkpoint/Restart library (BLCR)\(^3\):

Available in MPICH2 \(^4\).

A benchmark with one process and a 1GByte to checkpoint.

\(^3\)BLCR: https://ftg.lbl.gov/projects/CheckpointRestart/

\(^4\)MPICH2: http://www.mcs.anl.gov/research/projects/mpich2/
Energy in fault tolerance protocols - Checkpointing

Figure: Power consumption due to a 1GByte checkpointing
Energy in fault tolerance protocols - Checkpointing

Figure: Power consumption due to a 1GByte checkpointing for the less/more/median consuming nodes
Energy in fault tolerance protocols - Checkpointing

Figure: Extra power cost due to checkpointing
Energy in fault tolerance protocols - Checkpointing

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Checkpointing consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>the less consuming</td>
<td>2520</td>
</tr>
<tr>
<td>the median consuming</td>
<td>2875</td>
</tr>
<tr>
<td>the more consuming</td>
<td>3570</td>
</tr>
</tbody>
</table>

The large difference between the less and the more consuming nodes is mainly due to:

- the difference in the idle power consumption for about 70 %
- the difference in the checkpointing duration for about 30 %.
Message logging from Guermouche et al. The sender process logs all the messages that are sent to other processes.

The logging function used each time a process sends a message.

We log 100,000 messages of 100 KBytes to get a total volume of 10 GBytes.

We ran the same benchmark for the 64 nodes.
Energy in fault tolerance protocols - RAM logging

Figure: Power consumption of 10 GBytes of RAM Logging
Figure: Power consumption of 10 GBytes of HDD logging
Energy in fault tolerance protocols - HDD logging

**Figure:** Power consumption of 10 GBytes of HDD logging
Energy in fault tolerance protocols - Message logging

**Figure:** Extra power cost due to the message logging

RAM logging consume more power than HDD logging.
Energy in fault tolerance protocols - Message logging

Table: Energy ratio of Message Logging in Joules/GByte

<table>
<thead>
<tr>
<th>Nodes</th>
<th>RAM logging energy consumption</th>
<th>HDD logging energy consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>the less consuming</td>
<td>128</td>
<td>2550</td>
</tr>
<tr>
<td>the median consuming</td>
<td>137</td>
<td>2900</td>
</tr>
<tr>
<td>the more consuming</td>
<td>155</td>
<td>3600</td>
</tr>
</tbody>
</table>

We consume more energy by logging on HDD. Values for message logging on HDD are close to those of checkpointing on disk.
With a power capping point of view, users could decide to promote logging operation on HDD.

It is more energy efficient to log on RAM rather than on HDD. This is mainly due to the logging time:

- on HDD = more than 140 seconds for 10 GBytes
- on RAM = 7 seconds for 10 GBytes.
The process coordination implemented in MPICH2: a synchronization barrier.

A barrier in MPICH2 = an infinite loop that stops once the processes are synchronized.

In our testbed, an infinite synchronization barrier between the 64 processes of the 64 nodes

63 processes are running a barrier and 1 process is finalizing the MPI program.

We stop the infinite barrier after 30 seconds.
Energy in fault tolerance protocols - Process coordination

**Figure:** Power consumption of 64 nodes coordination
What is important is how long the coordination lasts. Eq. how long processes stay waiting each others.

We should minimize this waiting time: slowing down the fastest processes (DVFS).
Existing benchmarks that use resources (CPU, ...) intensively.

30 seconds of burnK6: an intermediate CPUburn \(^5\).

30 seconds of HDparm \(^6\).

30 seconds of STREAM \(^7\).

\(^5\)http://packages.debian.org/stable/cpuburn
\(^6\)http://doc.ubuntu-fr.org/hdparm
\(^7\)http://www.cs.virginia.edu/stream/
Results analysis - Comparison with intensive-using resources

**Figure:** Power consumption for the most/less/median consuming nodes
Results analysis - Comparison with intensive-using resources

**Table:** Extra power cost comparison

<table>
<thead>
<tr>
<th></th>
<th>HDparm between 7W and 10W</th>
<th>HDD Checkpointing and Logging 6W and 8W</th>
</tr>
</thead>
<tbody>
<tr>
<td>STREAM</td>
<td>18W</td>
<td>RAM Logging 18W</td>
</tr>
<tr>
<td>burnK6</td>
<td>23W</td>
<td>Process coordination 20W</td>
</tr>
</tbody>
</table>
Experiments with 4 NAS in class C (BT, CG, LU, and SP).

RAM logging compared to Process coordinations.

Table: Overall extra energy consumption (in kJ) of RAM logging and coordinations in NAS benchmarks with 64 processes

<table>
<thead>
<tr>
<th></th>
<th>BT</th>
<th>CG</th>
<th>LU</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM logging</td>
<td>16.06</td>
<td>14.44</td>
<td>5.85</td>
<td>25.65</td>
</tr>
<tr>
<td>Coordinations</td>
<td>20.32</td>
<td>15.14</td>
<td>13.18</td>
<td>16.52</td>
</tr>
</tbody>
</table>

http://www.nas.nasa.gov/publications/npb.html
Conclusion

Energy evaluation for fault tolerance protocols:

3 operations: checkpointing, message logging and coordination.

Process coordination and RAM logging consume more power than checkpointing and HDD logging.

For identical nodes performing the same operation, the extra power cost due to this operation is the same.

Power consumption of a node during a given operation remains constant during a operation.
Conclusion

More power to store data on RAM.
HDD logging is more energy consuming than RAM logging.

We obtain the same extra power consumption for existing benchmarks that use intensively the same resources.

Proposed how to make an energy-aware choice of fault tolerance protocols:

Message logging should be preferred for applications involving small volumes of data exchanged.
Current and Future works

Experiments on many infrastructures.

Energy estimation tool for fault tolerance protocols: For checkpointing (submitted) and restart (in progress).

Goal: enable to predict which protocol is the most energy efficient.

Investigate energy efficient solutions for fault tolerance protocols by applying some green leverages.
Conclusion

Thank you for your attention.