CHARACTERIZATION OF PARALLEL SCIENTIFIC SIMULATIONS

Master Project Thesis in Computer Engineering
High Performance Computing

Borja Miñano Maldonado

January 2017
Presentation outline

- Introduction
- Objectives
- Scientific simulations
- Metric and tool overview
- Result from
  - Characterization
  - Modelization
- Conclusions
Introduction

- High Performance Computing (HPC) for
  - Engineering
  - Scientific research
  - Business analytics
  - Weather forecast

- Performance through
  - Hardware improvement
  - Programming skills
Objectives

- Analysis and characterization of parallel scientific simulations
- Bottleneck detection
- Improvement proposals
Scientific simulations

- Single Program, Multiple Data (SPMD)
- Different resource demands
  - CPU
  - Memory
  - I/O requests
- 3 phases
  - Initial setup
  - Initialization – Memory allocation
  - Evolution – High computing and I/O demand
Memory-bound - Advection

- Advection equation
  - Fluid transport

- Simulation objective
  - Numerical discretization test

- Simple algorithm + Huge size

\[
\frac{\partial \psi}{\partial t} + u \cdot \nabla \psi = 0
\]
CPU-bound - Euler

- Euler equations
  - Movement of incompressible flow

- Simulation objective
  - Numerical discretization test

- Complex algorithm + Regular size

\[
\frac{\partial p}{\partial t} + \rho \ast \nabla u = 0
\]
\[
\frac{\partial u}{\partial t} + \nabla \frac{p}{\rho} = g
\]
\[
\frac{\partial e}{\partial t} + \frac{p}{\rho} \ast \nabla u
\]
I/O-bound – Cash and Goods

- Cash and goods Agent Based Model
  - Complex system on trading
- Simulation objective
  - Find unexpected behaviour
- Extremely simple algorithm + High output frequency
Metric overview

- CPU metrics
  - Usage detail
  - Arithmetic operations
  - Function calls
- I/O metrics
  - Read
  - Write
- Memory metrics
  - Read
  - Write
  - Page & cache faults
- Network metrics
  - Sending
  - Receiving

Summary and temporal statistics
Profiler tool selection

- Sar (temporal prof.)
  - CPU usage
  - Paging and cache
  - I/O usage
  - Network usage
- Valgrind Massif (temporal prof.)
  - Memory consuption
- Perf (summary prof.)
  - FLOPs
  - Cache rates
  - Memory access
Performance characterization Advection

Function elapsed time:
- Initialization: 51%
- Evolution: 15%
- Communication: 27%
- Others: 7%

CPU usage:
- user
- nice
- system
- iowait
- steal
- idle

Memory usage:
- MB vs. t

I/O rate:
- read/s
- write/s
Performance modeling - Advection

I/O usage

I/O scalability

Current speedup
Ideal speedup
Conclusions - Advection

- Memory-bound -> I/O-bound
- Output size was too demanding
- Bottleneck: unique hard disk drive

Suggestions:
- Higher rate HDD (hardware)
- Distributed architecture, multiple HDDs
- Less frequent or lighter output
## Performance Characterization

### Euler

<table>
<thead>
<tr>
<th>Event</th>
<th>Count</th>
<th>Events/ins</th>
<th>Events/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache-references</td>
<td>3,522,270.172</td>
<td>0.36 %</td>
<td>5.38 M</td>
</tr>
<tr>
<td>cache-misses</td>
<td>1,237,187.524</td>
<td>0.13 %</td>
<td>1.89 M</td>
</tr>
<tr>
<td>mem-stores</td>
<td>110,211,636.380</td>
<td>11.22 %</td>
<td>168.34 M</td>
</tr>
<tr>
<td>L1-dcache-loads</td>
<td>308,795,201.413</td>
<td>31.43 %</td>
<td>471.65 M</td>
</tr>
<tr>
<td>L1-dcache-load-misses</td>
<td>25,636,359.337</td>
<td>2.61 %</td>
<td>39.16 M</td>
</tr>
<tr>
<td>FLOPS</td>
<td>409,563,438.375</td>
<td>41.69 %</td>
<td>625.56 M</td>
</tr>
<tr>
<td>instructions</td>
<td>982,516,218.875</td>
<td>1.59 %</td>
<td>1,500.6 M</td>
</tr>
</tbody>
</table>

### Function elapsed time

- **Initialization**: 1%
- **Evolution**: 0%
- **Communication**: 0%
- **Others**: 99%
Performance modeling - Euler

![Graphs showing scalability and efficiency for strong and weak scalability.](image)
Conclusions - Euler

- CPU-bound
- Good scalability in physical processors
  - Ideal simulation
- Bottleneck: CPU
- Suggestions:
  - Quicker CPUs
  - More CPUs (not always recommended)
Performance characterization
Cash and goods
Performance modeling
Cash and goods - Memory

Memory usage, variable memory size

Memory scalability, variable memory size
Performance modeling
Cash and goods – I/O

Memory usage, variable I/O freq.

Memory scalability, variable I/O freq.
Conclusions – Cash and goods

- I/O-bound -> Memory-bound
- Unbalanced behaviour for one process
- Bottleneck: Memory access

Suggestions:
- Quicker caches/memory access (hardware)
- Distributed memory architecture
- Balance output algorithm
- Metrics to cover all resources
- Multiple tools:
  - Summary metrics
  - Temporal metrics
  - Consolidate conclusions
- Metric granularity for parallel processes
Project conclusions – Simulations

- Unexpected bottlenecks
- Modelization -> Scalability -> Speedup
Thank you for listening

Questions? / Comments?

Universitat Oberta de Catalunya
www.uoc.edu