Performance POP up

EU H2020 Center of Excellence (CoE)

Performance Engineering for HPC: Implementation, Processes & Case Studies
ISC 2017, Frankfurt, June 22nd 2017
• A Center of Excellence
  • On Performance Optimization and Productivity
  • Promoting best practices in performance analysis and parallel programming

• Providing Services
  • Precise understanding of application and system behavior
  • Suggestion/support on how to refactor code in the most productive way

• Horizontal
  • Transversal across application areas, platforms, scales

• For academic AND industrial codes and users
Partners

• Who?
  • BSC (coordinator), ES
  • HLRS, DE
  • JSC, DE
  • NAG, UK
  • RWTH Aachen, IT Center, DE
  • TERATEC, FR

A team with

• Excellence in performance tools and tuning
• Excellence in programming models and practices
• Research and development background AND proven commitment in application to real academic and industrial use cases
Motivation

Why?

• Complexity of machines and codes
  → Frequent lack of quantified understanding of actual behavior
  → Not clear most productive direction of code refactoring

• Important to maximize efficiency (performance, power) of compute intensive applications and the productivity of the development efforts

Target

• Parallel programs, mainly MPI /OpenMP ... although can also look at CUDA, OpenCL, Python, ...
3 levels of services

Application Performance Audit
- Primary service
- Identify performance issues of customer code (at customer site)
- Small Effort (< 1 month)

Application Performance Plan
- Follow-up on the service
- Identifies the root causes of the issues found and qualifies and quantifies approaches to address the issues
- Longer effort (1-3 months)

Proof-of-Concept
- Experiments and mock-up tests for customer codes
- Kernel extraction, parallelization, mini-apps experiments to show effect of proposed optimizations
- 6 months effort

Apply @ http://www.pop-coe.eu
Target customers

• **Code developers**
  • Assessment of detailed actual behavior
  • Suggestion of more productive directions to refactor code

• **Users**
  • Assessment of achieved performance on specific production conditions
  • Possible improvements modifying environment setup
  • Evidences to interact with code provider

• **Infrastructure operators**
  • Assessment of achieved performance in production conditions
  • Possible improvements modifying environment setup
  • Information for allocation processes
  • Training of support staff

• **Vendors**
  • Benchmarking
  • Customer support
  • System dimensioning/design
Activities (June 2017)

- **Services**
  - Completed/reporting: 80
  - Codes being analyzed: 21
  - Waiting user / New: 22
  - Cancelled: 10

- **By type**
  - Audits: 95
  - Plan: 15
  - Proof of concept: 13

  + 5 training workshops

- **Reports**
  - 5 - 15 pages
Fundamental performance factors

- Factors modeling parallel efficiency
  - Load balance (LB)
  - Communication
    - Serialization (or Micro load balance)
    - Transfer

- Factors describing serial behavior
  - Computational complexity: \#instr
  - Performance: IPC
  - Core frequency
  - Actual values, scaling behavior, impact on parallel efficiency factors

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

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel Efficiency</strong></td>
<td>0.9834</td>
<td>0.9436</td>
<td>0.8980</td>
<td>0.8478</td>
</tr>
<tr>
<td>Load Balance</td>
<td>0.9871</td>
<td>0.9687</td>
<td>0.9099</td>
<td>0.9177</td>
</tr>
<tr>
<td>Serialization efficiency</td>
<td>0.9975</td>
<td>0.9770</td>
<td>0.9938</td>
<td>0.9395</td>
</tr>
<tr>
<td>Transfer Efficiency</td>
<td>0.9988</td>
<td>0.9970</td>
<td>0.9931</td>
<td>0.9833</td>
</tr>
<tr>
<td><strong>Computation Efficiency</strong></td>
<td>1.000</td>
<td>0.9590</td>
<td>0.8680</td>
<td>0.6953</td>
</tr>
<tr>
<td><strong>Global efficiency</strong></td>
<td>0.9834</td>
<td>0.9049</td>
<td>0.7795</td>
<td>0.5894</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IPC Scaling Efficiency</strong></td>
<td>1.000</td>
<td>0.9932</td>
<td>0.9591</td>
<td>0.8421</td>
</tr>
<tr>
<td><strong>Instruction Scaling Efficiency</strong></td>
<td>1.000</td>
<td>0.9721</td>
<td>0.9393</td>
<td>0.9075</td>
</tr>
<tr>
<td><strong>Core frequency efficiency</strong></td>
<td>1.000</td>
<td>0.9932</td>
<td>0.9635</td>
<td>0.9098</td>
</tr>
</tbody>
</table>
Audit characterization

**Code**

- **Parallel programming model**
  - 77% MPI or MPI+X
  - 17% pure OpenMP
  - Few from new paradigms

- **Programming language**
  - 64% Fortran (+X) as expected
  - 9.4% Python (+X) not that expected

Histogram showing distribution of programming languages and parallel programming models.
Audit characterization

User profile

• Country
  • 23% requests from countries outside the consortium
  • 33.9% UK, 26.3% DE, 13.2% ES, 3.6% FR

• User institution versus code area
  • Industrial companies provide all cases from new HPC sectors
Audit characterization

**Code**

- **Scientific/technical area**
  - Dominated by Engineering and Physics
  - 90.5% of the requests from traditional HPC sectors
  - But also some requests on Data analytics, Deep learning, Medical, Media film, Text processing

### Area versus parallel programing model
Other activities

• Promotion and dissemination
  • Market and community development
  • Dissemination material and events

• Customer advocacy
  • Gather customers feedback, ensure satisfaction, steer activities

• Sustainability
  • Explore business models

• Training
  • Best practices on the use of the tools and programming models
    • Cooperation with other CoEs (EoCoE)
    • Lot of interest ... customers want to learn how to do it themselves
Audit characterization

Performance Audit results

• Parallel efficiency
  • At least 67% would benefit / require optimizations (acceptable + bad)
  • Most frequent reason for acceptable efficiency is data transfer and for bad efficiency is load balance (+ data transfer)

• Serial performance (IPC)
  • 44% have IPC >1 for all regions
  • Others may benefit from a serial performance improvement
    • 24% general IPC < 1
Case study: FDS Audit

• Customer:
  • SME
  • User of the code

• Code: FDS (Fire dynamics simulation)
  • Simulates fire and smoke development in structures

• Code Area: Engineering

• Performance Audit:
  • Efficiency drop above 200 cores
  • Evaluate efficiency running @ MareNostrum
Spatio-temporal structure

- Initialization
- Iterative phase
Scalability

4 iterations

16

32

64

128

256

Speedup

MPI ranks

- Speedup
- Linear
# Efficiency

<table>
<thead>
<tr>
<th></th>
<th>32</th>
<th>48</th>
<th>64</th>
<th>96</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Efficiency</td>
<td>91.74%</td>
<td>90.56%</td>
<td>88.74%</td>
<td>84.66%</td>
<td>86.41%</td>
<td>78.95%</td>
</tr>
<tr>
<td>Load Balance</td>
<td>94.60%</td>
<td>92.49%</td>
<td>93.40%</td>
<td>85.84%</td>
<td>87.05%</td>
<td>81.32%</td>
</tr>
<tr>
<td>Comm. Efficiency</td>
<td>96.97%</td>
<td>97.92%</td>
<td>95.01%</td>
<td>98.63%</td>
<td>99.26%</td>
<td>97.08%</td>
</tr>
<tr>
<td>Serialization</td>
<td>96.99%</td>
<td>97.95%</td>
<td>95.05%</td>
<td>98.70%</td>
<td>99.37%</td>
<td>97.54%</td>
</tr>
<tr>
<td>Transfer</td>
<td>99.98%</td>
<td>99.97%</td>
<td>99.96%</td>
<td>99.93%</td>
<td>99.89%</td>
<td>99.53%</td>
</tr>
<tr>
<td>Computation Scalability*</td>
<td>100.00%</td>
<td>102.51%</td>
<td>102.60%</td>
<td>103.55%</td>
<td>101.17%</td>
<td>95.64%</td>
</tr>
<tr>
<td>Global Efficiency</td>
<td>91.74%</td>
<td>92.84%</td>
<td>91.05%</td>
<td>87.67%</td>
<td>87.42%</td>
<td>75.50%</td>
</tr>
</tbody>
</table>

Table 1. Time efficiencies for the FOA from executions using 16 to 256 processes.

<table>
<thead>
<tr>
<th></th>
<th>32</th>
<th>48</th>
<th>64</th>
<th>96</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPC Scalability*</td>
<td>100.00%</td>
<td>101.33%</td>
<td>101.33%</td>
<td>101.33%</td>
<td>100.44%</td>
<td>98.22%</td>
</tr>
<tr>
<td>Instructions Scalability*</td>
<td>100.00%</td>
<td>101.34%</td>
<td>102.02%</td>
<td>101.90%</td>
<td>100.85%</td>
<td>97.71%</td>
</tr>
</tbody>
</table>

Table 2. Other efficiencies for the FOA from executions using 16 to 256 processes.

* Reference values are useful computation, IPC and total instructions based on 32 ranks.
### Efficiency

<table>
<thead>
<tr>
<th></th>
<th>32</th>
<th>48</th>
<th>64</th>
<th>96</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel Efficiency</td>
<td>91.74%</td>
<td>90.56%</td>
<td>88.74%</td>
<td>84.66%</td>
<td>86.41%</td>
<td>78.95%</td>
</tr>
<tr>
<td>Load Balance</td>
<td>94.60%</td>
<td>92.49%</td>
<td>93.40%</td>
<td>85.84%</td>
<td>87.05%</td>
<td>81.32%</td>
</tr>
<tr>
<td>Comm. Efficiency</td>
<td>96.97%</td>
<td>97.92%</td>
<td>95.01%</td>
<td>98.63%</td>
<td>99.26%</td>
<td>97.08%</td>
</tr>
<tr>
<td>Serialization</td>
<td>96.99%</td>
<td>97.95%</td>
<td>95.05%</td>
<td>98.70%</td>
<td>99.37%</td>
<td>97.54%</td>
</tr>
<tr>
<td>Transfer</td>
<td>99.98%</td>
<td>99.97%</td>
<td>99.96%</td>
<td>99.93%</td>
<td>99.89%</td>
<td>99.52%</td>
</tr>
<tr>
<td>Computation Scalability*</td>
<td>100.00%</td>
<td>102.51%</td>
<td>102.60%</td>
<td>103.55%</td>
<td>101.17%</td>
<td>95.64%</td>
</tr>
<tr>
<td>Global Efficiency</td>
<td>91.74%</td>
<td>92.84%</td>
<td>91.05%</td>
<td>87.67%</td>
<td>87.42%</td>
<td>75.50%</td>
</tr>
</tbody>
</table>

**Table 1. Time efficiencies for the FOA from executions using 16 to 256 processes.**

<table>
<thead>
<tr>
<th></th>
<th>32</th>
<th>48</th>
<th>64</th>
<th>96</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPC Scalability*</td>
<td>100.00%</td>
<td>101.33%</td>
<td>101.33%</td>
<td>101.33%</td>
<td>100.44%</td>
<td>98.22%</td>
</tr>
<tr>
<td>Instructions Scalability*</td>
<td>100.00%</td>
<td>101.34%</td>
<td>102.02%</td>
<td>101.90%</td>
<td>100.85%</td>
<td>97.71%</td>
</tr>
</tbody>
</table>

**Table 2. Other efficiencies for the FOA from executions using 16 to 256 processes.**

* Reference values are useful computation, IPC and total instructions based on 32 ranks.
More on structure ➔ clustering

• Structure
  • Different behaviour every fourth iteration
  • Different behaviours at the first and last ranks in some phases

• Sequential performance insight
  • Imbalance in instructions and IPC accumulate
  • Variability in IPC
Load Balance – Main Contributors

- Two loops within `RADIATION_FVM` (radi.f90:611) beginning at line 1113 and 1177
- `DIVERGENCE_PART_1` (divg.f90:14) and its subroutine `SPECIES_ADVECTION` (difg.f90:857).
- `DUMP_BNDF` (dump.f90:7075)
Refactoring?

• Techniques
  • Taskify + DLB?
  • Balance IPC?
  • Domain decomposition?
  • ...

• Within reach, interest, ... of customer?
Refactoring?

• Techniques
  • Taskify + DLB?
  • Balance IPC?
  • Domain decomposition?
  • ...

• Within reach, interest, ... of customer?

Decomposition: X
  Load balance: 80%

Decomposition: XY
  Load balance: 81%
  Rel. runtime: 95%

Decomposition: XYZ
  Load balance: 91%
  Rel. runtime: 80%
Case study: Kratos

- **Customer:**
  - Research center
  - Developer of the code

- **Code:** Kratos
  - Multi-physics FE

- **Code Area:** engineering

- **Performance Audit:**
  - Happy with MPI scaling
  - Concerns on OpenMP scaling
• In reality two different codes, similar structure
  • Multigrid non linear solver
OpenMP runs Scaling
OpenMP runs efficiencies

- Serial performance (4 longest regions)

- Instruction efficiency: slight increase in total instruction count
  - Atomics ??
OpenMP Serial performance

- Longer 4 regions

- Reason?
  - Computational?
  - NUMAness?
  - Numbering?
  - Combined?
  - None?

Instructions

IPC

L2 miss ratio

L1 miss ratio
OpenMP Serial performance

• Finer grain regions

• Reason?
  • Computational?
  • NUMAness?
  • Numbering?
  • Combined?
  • None?
• Many coupled effects
  • NUMAness, variability in cache miss ratios, atomics overheads (and contention?)

• Recommendations
  • NUMA initialization
    • Though they were doing it. Inadvertedly happened to be in the wrong control flow branch
    • Really activated → std::vector NUMA unfriendly issues …. Took some more time to fix
  • Explore potential benefits of more dynamic schedules
  • Work on numbering schemes
    • WIP: Not only balances IPC but also improves it
  • Eliminate atomics. Commutative multideps clause (OmpSs)?
    • Verified high atomics overhead (running version with races)
Ongoing progress

• Refactoring being implemented by customer
Case study: GraGLeS2D Audit

- **User:**
  - University
  - Developper

- **Code: GraGLeS2D**
  - Simulates the grain growth in polycrystalline materials

- **Code Area: Material Science**

- **Performance Audit:**
  - Poor scaling on a NUMA machine with 128 cores
GraGLeS2D Audit Analysis

• Analysis of OpenMP with 8 – 128 cores
  • 4 boards x 4 sockets x 8 cores

• Observations from Audit
  • Work balance good except for the first iteration
  • Data sharing causing remote memory access reduces scalability
  • Detected consuming loops that can be vectorised

• PoC proposed and implemented
GraGLeS2D Proof of Concept

• PoC Plan
  • improve data-locality by thread pinning and load-distribution
  • improve vectorisation and serial performance

• Results on test input
  • parallel regions: speedup 6.4
  • overall application: speedup 2.2
## Codes analyzed

<table>
<thead>
<tr>
<th>DPM</th>
<th>Quantum Espresso</th>
<th>DROPS</th>
<th>Ateles</th>
<th>SHP-Fluids</th>
<th>GraGLeS2D</th>
<th>NEMO</th>
<th>VAMPIRE</th>
<th>psOpen</th>
<th>GYSELA</th>
<th>AIMS</th>
<th>OpenNN</th>
<th>FDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baleen</td>
<td>Mdynamix</td>
<td>ParFlow</td>
<td>GITM</td>
<td>BPMF</td>
<td>FIRST</td>
<td>SHEMAT</td>
<td>GS2</td>
<td>ADF</td>
<td>DFTB</td>
<td>ICON</td>
<td>dwarf2-ellipticsolver</td>
<td>EPW</td>
</tr>
<tr>
<td>Code Saturne</td>
<td>ONETEP</td>
<td>Ms2</td>
<td>SIESTA</td>
<td>Oasys GSA</td>
<td>SOWFA</td>
<td>BAND</td>
<td>NGA</td>
<td>Fidimag</td>
<td>LAMMPS</td>
<td>ScalFMM</td>
<td>CHAPSIM K.W.</td>
<td>ArgoDSM</td>
</tr>
<tr>
<td>CIAO</td>
<td>FFEA</td>
<td>k-Wave</td>
<td>DSHplus</td>
<td>RICH</td>
<td>COOLFluiD</td>
<td>Ondes3D</td>
<td>ATK</td>
<td>Molcas</td>
<td>GBMol_DD</td>
<td>Kratos</td>
<td>cf-python</td>
<td></td>
</tr>
</tbody>
</table>

+ few under NDAs
Performance Optimisation and Productivity
A Centre of Excellence in Computing Applications

Contact:
https://www.pop-coe.eu
mailto:pop@bsc.es

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 676553.