Geneva 1.6

„Grid-enabled evolutionary algorithms“

Simplifying the parallelization of highly concurrent workloads

Dr. Sven Gabriel / Dr. Rüdiger Berlich
Gemfony scientific UG (haftungsbeschränkt)

19.03.2015

Contact:
contact@gemfony.eu
Who we are

• Gemfony scientific
  – A spin-off from Steinbuch Centre for Computing at Karlsruhe Institute of Technology

• With particular experience in the fields of
  – Optimization of complex systems
  – Technical- and Science-Consulting
  – Implementation of IT-Solutions
  – Technical Marketing, PR and Training

• Long-standing background in parametric optimization
  – Gemfony maintains the Geneva library collection of distributed optimization algorithms
Parametric Optimization: Finding maxima or minima of \( \bar{Q} = \bar{f}(x_1, x_2, \ldots, x_n) \)

- Any mapping “\( f \)” from input parameters to one or more numeric evaluations can be optimized
- For one evaluation criterion: optimization == finding suitable minima or maxima of solver “\( f \)”
  - Very similar to the search for extreme values of mathematical functions
  - “Solution space” for multiple criteria
- However in the general case, “\( f \)” will be a computer program
  - Solvers may be computationally expensive
- Hence standard mathematical procedures cannot be applied easily
The Geneva Library Collection

- Generic solution for the search for **optimized solutions** of technical and scientific problems
- "Metaheuristic" Optimization
  - Covering *Evolutionary Algorithms*, Swarm Algorithms, Simulated Annealing, Parameter Scans and Gradient Descents
- Data structures allow direct interaction between different optimization algorithms with **just one problem description**
- Written in portable C++
  - Uses the Boost library collection
  - Runs on different Unix variants (Linux, MacOS, ...) and Windows (experimental!)
- > 130,000 LOC (.hpp, .cpp, scripts, ...)
- Available under an Open Source License (see [http://www.launchpad.net/geneva](http://www.launchpad.net/geneva))

Sources:
- Car: Image courtesy of Simon Howden at FreeDigitalPhotos.net
- Particle decay: [https://en.wikipedia.org/wiki/File:CMS_Higgs-event.jpg](https://en.wikipedia.org/wiki/File:CMS_Higgs-event.jpg) Creative Commons Attribution Share-Alike 3.0; By CERN
• Geneva targets particularly complex problems
  – Current use-cases: Optimization of the acoustics of exhaust systems, materials sciences, particle physics
  – May only rely on „local“ information of the quality surface (e.g. no exact gradient available)
• May use long-running evaluation function
  – Typical usage scenario: Simulation running outside of the optimizer
• May need to deal with a particularly large number of evaluations (often thousands of calls)
  – Multi-criterion / Pareto-optimization may significantly increase the required number of evaluations!
• Parallelization needs to happen on the level of concurrent execution of the evaluation function
Parallelization (2)

• Geneva needs to accommodate distributed (cluster / Grid / Cloud) and multi-threaded execution as well as GPGPU

• The problem is simplified by the fact that optimization algorithms are generic in nature and do not need to know anything about the underlying optimization problem (other than that they get back an evaluation for a given parameter set)

• Hence: Need to de-couple optimization algorithms from mode of execution of the evaluation function!
Library Components

Strong modularization allows for an efficient decoupling of evaluation and optimization!
The „Courtier“ Library: Problem-Independent Parallelization

From the server-perspective, in networked mode the Problem may be quite similar to a high-load web-server.
Tuning Geneva Performance

• The term „Performance“ is very problem-dependent
• In a nutshell, on the same hardware, performance improvements may be achieved in numerous ways:
  – Making the code more efficient
    • BUT: Focus on long-running evaluation-functions mandates focus on core-library stability rather than performance
  – Parallelization of parallelizable parts
    • Reducing parallelization-overhead: „Amdahl“ may have a major impact on performance
    • Asynchronous transfer of candidate solutions, current focus
  – Making optimization algorithms converge faster
    • Current focus of work
  – Reducing run-time of the solver(s)
Measuring Network-Overhead, Amdahl

Largest bottle-neck is (de-)serialization. Asynchronous transfers may help here

Source: Gemfony
Thanks!

If you want to try Geneva:
http://launchpad.net/geneva

You may reach us at
contact@gemfony.eu

The project-team wants to thank Karlsruhe Institute of Technology, Steinbuch Centre for Computing as well as the Helmholtz-Association for supporting our work!
<table>
<thead>
<tr>
<th>Address</th>
<th>Gemfony scientific UG (haftungsbeschränkt) Leopoldstr. 122 76344 Eggenstein-Leopoldshafen Germany</th>
</tr>
</thead>
<tbody>
<tr>
<td>Téléphone</td>
<td>+49(0)7247/934278-0</td>
</tr>
<tr>
<td>Fax</td>
<td>+49 (0)7247 934 2781</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:contact@gemfony.eu">contact@gemfony.eu</a></td>
</tr>
<tr>
<td>Registered at</td>
<td>Amtsgericht Mannheim (Germany)</td>
</tr>
<tr>
<td>Registration-Id</td>
<td>HRB 710566</td>
</tr>
<tr>
<td>Ust.-Id</td>
<td>DE274421406</td>
</tr>
<tr>
<td>Managing Director</td>
<td>Dr. Rüdiger Berlich</td>
</tr>
</tbody>
</table>