Distributed parametric optimization using the Geneva library

*Dr. Rüdiger Berlich*
*Karlsruhe Institute of Technology*
*Steinbuch Centre for Computing*

*ISGC 2010, Taipei*
Karlsruhe Institute of Technology - North Campus -

• Part of the Helmholtz alliance

• Merger of Forschungszentrum Karlsruhe and University of Karlsruhe

• Many, very diverse research topics, ranging from nano technology to Grid- and Cloud-Computing
Geneva

= **Grid-enabled evolutionary algorithms**

- Parallel optimization of problems from scientific and industrial domains.
- Covering multi-core machines, clusters, Grids and Clouds
- Available as Open Source (Affero GPL v3)
- Implemented in portable C++ (usage of external libraries limited to Boost)
- See [http://www.launchpad.net/geneva](http://www.launchpad.net/geneva) and [http://www.gemfony.com](http://www.gemfony.com)
- Protein folding
- Modelling the Mona Lisa
Defining „Optimization“

- Naive approach:
  - Lat. „Optimum“ == „The best“
  - Define an evaluation criterion
    - Implies agreement on single criterion
      - Amongst contributors and stake-holders
      - Between possibly competing criteria
  - Find all possible solutions /parameter sets to a problem
  - Compare with each other to find best solution
    - Impossible for real-life problems:
      - Imagine 3000 parameters, test 2 values each
      - Means computation of $2^{3000}$ parameter sets
      - Impossible to compute in remaining lifetime of the universe
  - It is usually not possible to reach „the best“ solution
• More realistic approach:
  – Optimization refers to the search for the *best achievable result* under a set of constraints
  – In comparison: „The ideal“ solution is the *best possible result*

• **Strategy:**
  – Identify all relevant parameters, including constraints
  – Assign a (computable) evaluation criterion to the parameters
    ▪ Encapsulates experts knowledge
  – Search for maxima and minima of the criterion using one of many different optimization algorithms
    ▪ Generic approach, applicable to many different problem domains
• **Algorithm:**
  - Step into direction of steepest descent

• **Advantages:**
  - Converges quickly and safely into next available optimum
  - Easy to understand and implement

• **Disadvantages:**
  - Easily gets stuck in local optima
  - Compute time scales with number of parameters
• **Algorithm:**
  – Population of parents (best known solutions) and children
  – Cycle of duplication, mutation, selection
  – Mutation usually through addition of gaussian-distributed random numbers

• **Advantages:**
  – Tolerant wrt. local optima
  – Compute time scales with size of the population
  – Easy to parallelise

• **Disadvantages**
  – Can be slower than gradient descent for smaller problems
  – Many configuration options (e.g. width of gaussian)
Other Optimization Algorithms

- Further interesting algorithms:
  - Genetic algorithms
  - Swarm algorithms
  - Deluge algorithms
  - Line search, Simplex, …

- Swarm algorithms and Gradient Descents will soon be implemented in Geneva

- Both can be parallelized seamlessly with the means provided by Geneva

(Source: Wikipedia; Author Mila Zinkova; published under the Creative Commons license „Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Unported“)
Design Criteria

- **Focus on long-lasting, computationally expensive evaluation functions**
  - Stability of *core library* rated higher than efficiency
- **Serial, multi-threaded and networked execution, transparent to users**
  - Implications of networked and multi-threaded execution:
    - No global variables
    - User-defined data structures must be serializable
- **Familiar interface**
  - STL wrapper for data, individuals, populations, ...
- **Fault tolerance of networked execution:**
  - Algorithm must be able to repair itself in case of missing or late replies from clients
- **Execution of clients in Grid and Cloud:**
  - No push mode means: Server needs public IP, clients don't
- **Easy, portable build environment:**
  - CMake
- **Quality assurance:**
  - Unit-tests, based on Boost.Test library
  - Can be integrated into user code
• **C++**
  - Efficient (cmp. Java)
  - Heavily uses Boost
• **So far largely Linux-based**
  - But: should be portable
  - Tested with Intel C++, var. G++
• **Major components**
  - Repres. of parameter sets
  - Optimization framework
  - Parallelization and communication
  - Random number factory
Implementation

Enabling Grids for E-sciencE

Distributed Parametric Optimization with the Geneva Library

EGEE-III INFSO-RI-222667
Using the Geneva Library

- **Code example**
  - http://www.launchpad.net/geneva
  - Try: Server on Laptop in Taiwan, clients on GridKa centre at Karlsruhe Institute of Technology in Germany, with private IP space

- **Running example**
  - See example „GStartProject“, part of the Geneva distribution
Nehalem system with 2 processors / 8 cores / hyperthreading
• **Roughly:**
  - Speedup scales with the percentage of parallel execution time of the overall application runtime

• **Strong scalability constraints**
  - Need very high percentage of parallel execution time to achieve significant speedup (as function of the number of parallel processing units)

\[ S = \frac{1}{(1 - P) + o(N) + \frac{P}{N}} \leq \frac{1}{1 - P} \]


Author of picture: Bob Schwammerl
Performance / Scalability

Multithreaded execution, 16 evaluation threads. Speedup at theoretical maximum ...

Networked mode (local networking + "ideal" cluster with 16 clients each, GridKa production-mode cluster with 20 clients)

Networked mode (GridKa production-mode cluster, 16 clients)

Speedup when using 16-20 individuals (cluster, multi-core) with population of 16 children
Scalability / Pareto

Fitness

Initial fitness 5.47e+06

Fitness after 20% of the time: 8.35e+05. This is equivalent to 96.5% of the overall improvement
• Many low-hanging fruits for distributed optimization both in industry and science
• Find further information about the Geneva library on http://www.gemfony.com
• Get the software from http://www.launchpad.net/geneva
• We want to build a community. Please do contact us with your optimization problems, we are happy to help getting you started with Geneva
Thanks!

- I want to thank the audience and the organizers.
- Steinbuch Centre for Computing as well as the department IMA of Karlsruhe Institute of Technology have supported my work – thanks a lot!
- Similarly, I want to thank the Helmholtz Society of German research centres for their kind help.
- The Enabling Grids for E-SciencE project has given this work a scientific home – thanks!!
Question ? Questions!

ruediger.berlich@kit.edu
http://ruediger.berlich.com