CODE PARALLELIZATION AND APPLICATION OPTIMIZATION FOR A HETEROGENEOUS COMPUTING
Agenda

- About heterogeneous computing
- Heterogeneous computing advantage
- Current services
- Engagement process
- Evaluation of expected optimization and code porting success
- Application fields
- Expected performance gains
- Examples of performance gains
- ttgLib utility
About heterogeneous computing

Today there are many high performance computing systems, employing NVIDIA Tesla accelerators based on GF1x0 GPUs. These accelerators can run dual precision computations much more efficiently than standard, general purpose x86 microprocessors.

A recent advent of computational clusters, based on hybrid architecture, helps significantly reduce runtime for the most resource-demanding computational jobs and offer certain advantages. CUDA or OpenCL environment allow to parallelize computing, to reach significant application performance gains by utilizing 512 or 1024 computational cores of NVIDIA GF1x0 on each node.

However, rather often applications that exist today, do not use the entire potential of computational system in general and accelerators in particular.
Heterogeneous computing advantage

GPU-based computing helps to:

-Accelerate applications by dozens of time, reducing overall runtime
-Decrease the number of required compute nodes
-Reduce computational cluster footprint
-Reduce electricity bills to cover cluster’s power and cooling
-Significantly improve Gflops/Watt and Gflops/$ ratio
-Reduce TCO
Current services

- Adaptation of an application for GPU environment
- CUDA/OpenCL-based program optimization
- Program porting to NVIDIA CUDA technology
- Consulting
- Solving technical issues, related to NVIDIA CUDA and OpenCL environments
- Solving issues, associated with integration of existing applications with the new technologies
- System utilities to simplify programming for CUDA and OpenCL
Engagement process

- If you are not satisfied with the current application code
- Contact T-Platforms
- Specialists would assess FOR FREE the potential of computational code optimization/porting to GPU
- If required at consulting stage, customer can pay for the exact estimation of required resources and perspective performance gains, to receive detailed recommendations report as well as RFP documentation
- In case project is realized by T-Platforms the customer would get:
  - modified code and instructions on how it was compiled
  - instructions on code assembly and launch routine
  - preassembled code versions
  - optimal application with high performance
The next scheme is used for the evaluation:

- The customer launches a utility, to gather information on existing code and on computational system (if such system exists)
- Based on received data, specialists do an approximation on amount of work required and send expected pricing range variation
- After NDA is ratified and the source code is received, both resources and costs are finalized
Application fields

- **Science**
  - air, hydro and gas dynamics
  - fluid and gas mechanics
  - rigid body mechanics
  - thermal physics and heat mass transfers
  - acoustics
  - structural analysis
  - computational biology
  - geo and seismic research
  - ecology and natural cataclysm forecasts

- **Industry**
  - aerospace
  - automotive
  - engineering
  - shipbuilding

- **Business**
  - trading floor and financial analysis
  - business analytics
Expected GPU-based performance gains for the most common applications

<table>
<thead>
<tr>
<th>Field</th>
<th>An expected acceleration, times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aero and hydro dynamics, thermal dynamics, structural analysis</td>
<td>7-30</td>
</tr>
<tr>
<td>Image and video editing, machine vision</td>
<td>10-40</td>
</tr>
<tr>
<td>Rendering and 3D effects</td>
<td>20-70</td>
</tr>
<tr>
<td>Monte-Carlo methods</td>
<td>50-300</td>
</tr>
<tr>
<td>Molecular dynamics, protein analysis, drug research</td>
<td>20-100</td>
</tr>
<tr>
<td>Oil and gas</td>
<td>15-40</td>
</tr>
<tr>
<td>Computational chemistry</td>
<td>15-80</td>
</tr>
<tr>
<td>Crypto analysis</td>
<td>7-50</td>
</tr>
</tbody>
</table>
An example of performance increase for GPU-based image editing (Laplace’s equation-based)
An example of GPU-based performance increase for heat and mass transfer

![Bar chart comparing CPU and GPU performance for different matrix sizes in thousands of nodes per second. The chart shows a significant increase in performance with GPU compared to CPU, especially for larger matrix sizes.](chart.png)
ttgLib utility

**ttgLib** – is a developer instrument to automate the process of program’ adaption for the combination of ‘entire system+ input data’ to lead to a 50%+ performance gains

**Features:**

- Dynamic learning ability. The program processes each new data set faster and faster
- Dynamic adaptation for the combination of HW and input data. Dynamic adaptation helps to reach the maximal performance, for different systems and application tasks
- A collection of hybrid programming primitives helps to easily change the parameters of existing software, which in most cases leads to a minimum of 50% performance gains even without significant changes in code
Performance gains with dynamic learning-based ttgLib utility
Thank you!