Integration of Everest Platform with BOINC-based Desktop Grids

Oleg Sukhoroslov

Institute for Information Transmission Problems of the Russian Academy of Sciences (Kharkevich Institute)
Agenda

- Overview of Everest platform
- Integration of Everest with BOINC
Motivation
Everest

• Web-based platform supporting
  – Publication of computational applications as services
  – Execution of applications on external computing resources
  – Sharing applications and resources with other users
  – Composition of applications (workflows)

• Platform as a Service
  – Remote access via web browser and REST API
  – Single platform instance can be accessed by many users
  – No installation is required

• Public instance with open registration
  – http://everest.distcomp.org/
Everest Architecture
# AutoDock Vina

## Inputs

<table>
<thead>
<tr>
<th>Title</th>
<th>Name</th>
<th>Type</th>
<th>Values</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receptor</td>
<td>receptor</td>
<td>URI</td>
<td></td>
<td></td>
<td>rigid part of the receptor (PDBQT)</td>
</tr>
<tr>
<td>Ligand</td>
<td>ligand</td>
<td>URI</td>
<td></td>
<td></td>
<td>ligand (PDBQT)</td>
</tr>
<tr>
<td>Center X</td>
<td>center_x</td>
<td>number</td>
<td></td>
<td></td>
<td>X coordinate of the center</td>
</tr>
<tr>
<td>Center Y</td>
<td>center_y</td>
<td>number</td>
<td></td>
<td></td>
<td>Y coordinate of the center</td>
</tr>
<tr>
<td>Center Z</td>
<td>center_z</td>
<td>number</td>
<td></td>
<td></td>
<td>Z coordinate of the center</td>
</tr>
<tr>
<td>Size X</td>
<td>size_x</td>
<td>number</td>
<td>[0, MAX]</td>
<td></td>
<td>size in the X dimension (Angstroms)</td>
</tr>
<tr>
<td>Size Y</td>
<td>size_y</td>
<td>number</td>
<td>[0, MAX]</td>
<td></td>
<td>size in the Y dimension (Angstroms)</td>
</tr>
<tr>
<td>Size Z</td>
<td>size_z</td>
<td>number</td>
<td>[0, MAX]</td>
<td></td>
<td>size in the Z dimension (Angstroms)</td>
</tr>
<tr>
<td>Exhaustiveness</td>
<td>exhaustiveness</td>
<td>integer</td>
<td>[1, MAX]</td>
<td>8</td>
<td>exhaustiveness of the global search (roughly proportional to time)</td>
</tr>
</tbody>
</table>

## Outputs

<table>
<thead>
<tr>
<th>Title</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>output</td>
<td>URI</td>
<td>output models (PDBQT)</td>
</tr>
<tr>
<td>Log</td>
<td>log</td>
<td>URI</td>
<td>Vina log file</td>
</tr>
</tbody>
</table>
Parameter Sweep Service

Parameter Sweep Application

Integration with Computing Resources

- Agents
- Everest
- EMI UI
- Dedicated Resources
  - Server (local)
  - Cluster (torque)
  - Cluster (docker)
  - Cluster (slurm)
- Grid Infrastructures
- Desktop Grids
- Clouds
Combined Use of Multiple Resources
Current Status

• Everest website and public platform instance
  – http://everest.distcomp.org/
  – 507 users, 93 applications, 50 resources

• Used in several research projects and educational activities
  – Distributed optimization, computer simulation, parameter study, plasma physics, geophysics, bioinformatics
  – Teaching parallel and distributed computing
Agenda

- Overview of Everest platform
- Integration of Everest with BOINC
Motivation

- Desktop grids
  - An important class of distributed computing infrastructures that aggregate computing power of idle personal computers
  - Provide performance comparable to supercomputers but with much smaller costs, affordable for small research projects

- Goals
  - Enable Everest users to seamlessly run applications on desktop grids and combine them with other types of resources
  - Make it possible to use Everest for building web services for submission and automation of computations in desktop grids
BOINC

- An open source middleware platform for building desktop grids and running volunteer computing projects

- Client-server architecture

- Application development
  - Native application binary for each supported platform
  - Virtual machine (VirtualBox)
  - Docker image (boinc2docker)

- Job submission
  - `create_work` command-line tool on BOINC server
  - Remote job submission mechanism based on WebRPCs
Possible Approaches for Integration

- **Agent-based approach**
  - Reuse the same mechanism that is used for integration with clusters
  - Run Everest agent on BOINC server and use it to translate Everest tasks to BOINC jobs, query their state and download results

- **Remote job submission**
  - Use the new the remote job submission mechanism to directly submit jobs to BOINC server
  - Requires passing authentication token of submission account to Everest
  - More development effort, the mechanism is relatively new and not stable

- **Dynamic agent deployment**
  - Dynamically deploy agents to BOINC hosts by submitting workunits with the agent
  - Use deployed agents to execute Everest tasks, bypassing BOINC server for task scheduling and data transfer (e.g., using more flexible scheduling strategies)
  - Increased load on Everest, not clear how to validate results and grant credits
Prototype Implementation

https://gitlab.com/everest/agent/tree/boinc

Application

submit job

job (tasks)

job state/results

BOINC Server Host

create/monitor workunits

store/read workunit files

Agent

boinc

BOINC Database

File Storage

Desktop Grid

BOINC Client

boot2docker

VirtualBox VM

WebSocket

monitor agent state

submit tasks

monitor task state

HTTP

download/upload task files

Everest

job (tasks)

task input/output files

job state

File Storage

Task Manager

task

task state

Application

Agent Client

https://gitlab.com/everest/agent/tree/boinc
Current Limitations

• Supports only `boinc2docker` application

• Docker image used for running Everest tasks is fixed in the agent configuration

• Requires custom assimilator that copies output files to a directory checked by the agent
Testing

- Virtual screening of 100 ligand molecules using Autodock Vina
  - Typical parameter sweep experiment
  - Uses Vina Linux binary, auxiliary Python script and input files (protein, ligand)

- Test BOINC project with boinc2docker application
  - The agent was deployed on BOINC server and configured to submit workunits using the frolvlad/alpine-python2 Docker image
  - Four desktop hosts were attached and used during the tests

- The experiment was submitted to Everest via the Parameter Sweep service
  - The platform successfully executed the experiment by dispatching all tasks to BOINC and collecting the task results

- Additional tests
  - Mixing multiple resources for parameter sweep run
  - Running other existing Everest applications
Conclusion

- The possible approaches for integration of Everest platform and BOINC-based desktop grids are discussed.
- The prototype integration based on Everest agent and boinc2docker application is presented.
- The proposed integration enables Everest users to seamlessly access computing resources of desktop grids.
- Future work will focus on improving the described implementation and conducting a large-scale experimental evaluation.