Templet Web: The Experimental Use of Volunteer Computing Approach in Scientific Platform-as-a-Service Implementation

Sergei Vostokin, Yuriy Artamonov, and Danil Tsarev

BOINC:FAST-2017, Petrozavodsk
28 August - 01 September 2017
Templet Web is an academic cloud service, what belongs to the category of Platform-as-a-Service.

The aim of the service is to provide affordable high performance computing (HPC) tool for researchers and students.

What makes the HPC affordable in the Templet Web service?

• Access procedure for a high-performance system.
• Application deployment automation.
• Parallel programming automation.
**Problem:** providing access to a remote cluster  
**Method:** volunteer computing approach

**Problem:** forecasting computational load of the cluster  
**Method:** adaptive forecast models:  
similar patterns extrapolation +  
artificial neural networks

**Problem:** parallel programming  
**Method:** algorithmic skeletons +  
actor-like computation model
1. Consumer submits the program source code and input data to Volunteer
2. Volunteer runs the program on his/her own behalf on the remote system
3. Volunteer returns result of the run to Consumer
Templet Web system acts as a broker and assumes the following obligations:

- **storing of source code, data and the result** of computations for mutual audit of Consumers and Volunteers actions;

- organizing the access to this information by both Volunteer and Consumer (**auditing facility**);

- **multiplexing access** for many Consumers to the same Volunteer account.
create a project

add a consumer to the project
add an environment to the project

with a link to the remote system
get 12-hour load forecast
The Template contains sample code that users can adapt to their algorithm; the script that controls the assembly of the Task on the Volunteer system; there are also a start script and the script for downloading the results.

The attributes of a task are the code of the Customer program, the input data, the output data and the execution status. The Task is generated from the Template.

Each Environment contains information for connecting to the Volunteer system. Environment implements the life cycle of Tasks in Volunteer system.
**APPLICATION DEPLOYMENT AUTOMATION (2/5)**

List of templates:

<table>
<thead>
<tr>
<th>№</th>
<th>Name</th>
<th>Creation Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>C++ App</td>
<td>07.08.2016 13:52</td>
<td>Шаблон программы на C++ для выполнения в пакетном режиме на суперкомпьютере</td>
</tr>
<tr>
<td>3</td>
<td>Tempel</td>
<td>07.08.2016 13:52</td>
<td>Проверка тождества (\sin 2x + \cos 2x = 1). Пример синтаксиса языка разметки</td>
</tr>
<tr>
<td>2</td>
<td>Taskbag</td>
<td>07.08.2016 13:52</td>
<td>Скелет &quot;Портфель задач&quot;. Пример построчного параллельного умножения</td>
</tr>
<tr>
<td>10</td>
<td>RingMult</td>
<td>07.08.2016 13:52</td>
<td>Параллельное умножение матрицы на кольце процессов с использованием POSIX</td>
</tr>
<tr>
<td>1</td>
<td>POSIX App</td>
<td>07.08.2016 13:52</td>
<td>Шаблон многопоточной программы, использующей библиотеку POSIX</td>
</tr>
<tr>
<td>7</td>
<td>Pipeline C++11</td>
<td>07.08.2016 13:52</td>
<td>Реализация скелета &quot;Конвейер&quot; на языке разметки Tempel с применением POSIX</td>
</tr>
<tr>
<td>4</td>
<td>Pipeline</td>
<td>07.08.2016 13:52</td>
<td>Скелет &quot;Конвейер&quot; Пример алгоритма метода Зейделя решения задачи на основе</td>
</tr>
</tbody>
</table>

Самарский национальный исследовательский университет имени академика С.П. Королёва

34, Moskovskoye shosse, Samara, 443086, Russia, tel.: +7 (846) 335-18-26, fax: +7 (846) 335-18-36 www.ssau.ru, e-mail: ssau@ssau.ru
Press run button to deploy the program.

Tasks appear in the list.
you can observe the status of deployed task
you can browse results when the task is completed
Bag-of-tasks skeleton

```c
struct task{
    /* to be filled by the user */
};
struct result{
    /* to be filled by the user */
};
struct bag{
    bool get(task*t){ /* to be filled by the user */
    void put(result*r){ /* to be filled by the user */
    /* to be filled by the user */
};
void proc(task*t,result*r){
    /* to be filled by the user */
    int main(int argc, char* argv[])
{
    bag b;  /* to be filled by the user */
    b.run();  /* to be filled by the user */
    return EXIT_SUCCESS;
}
```

Extension points

- abstract task
- abstract result of task calculation
- bag of tasks
- when and how to get a new task
- how to put and accumulate results
- how to process a task
- how to input data to the program
- how to output the program results

All implementation details are hidden from the programmer
Table 1. *The dynamics of Templet Web user growth*  

<table>
<thead>
<tr>
<th>Year</th>
<th>Total in the period</th>
<th>Accumulated total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-2015</td>
<td>212</td>
<td>212</td>
</tr>
<tr>
<td>2016</td>
<td>63</td>
<td>278</td>
</tr>
<tr>
<td>2017 (first six months)</td>
<td>88</td>
<td>366</td>
</tr>
</tbody>
</table>

Table 2. *The number of task runs on Templet Web*  

<table>
<thead>
<tr>
<th>Year</th>
<th>Total in the period</th>
<th>Accumulated total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-2015</td>
<td>141</td>
<td>141</td>
</tr>
<tr>
<td>2016</td>
<td>2597</td>
<td>2738</td>
</tr>
<tr>
<td>2017 (first six months)</td>
<td>1308</td>
<td>4046</td>
</tr>
</tbody>
</table>
### Table 3. The dynamics of creating projects in Templet Web

<table>
<thead>
<tr>
<th>Year</th>
<th>Total in the period</th>
<th>Accumulated total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-2015</td>
<td>153</td>
<td>153</td>
</tr>
<tr>
<td>2016</td>
<td>126</td>
<td>279</td>
</tr>
<tr>
<td>2017 (first six months)</td>
<td>237</td>
<td>516</td>
</tr>
</tbody>
</table>

### Table 4. The number of projects managed in browser

<table>
<thead>
<tr>
<th>Year</th>
<th>Total in the period</th>
<th>Accumulated total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013-2015</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2016</td>
<td>97</td>
<td>97</td>
</tr>
<tr>
<td>2017 (first six months)</td>
<td>226</td>
<td>323</td>
</tr>
</tbody>
</table>
Sergei Vostokin
Professor, Information Systems and Technologies Department, Samara National Research University

e-mail: vostokin_sv@ssau.ru,
Tel.: +7(846)267-45-17