

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

ФАКУЛЬТЕТ ИНФОРМАТИКИ  
КАФЕДРА ПРОГРАММНЫХ СИСТЕМ

**Отчет по дополнительному заданию**

Установка Python версии 3.0 и выше для агент системы Everest

Выполнили

Баньков А. Д.

Вальков Ю. Д.

Группа 6301 – 090301

Проверил

Востокин С. В.

Самара 2023

Введение:

Everest – платформа для управления выполнением параллельных задач на распределенных вычислительных ресурсах через Интернет. Она также может моделировать логику и измерять производительность сложного многозадачного рабочего процесса на одном компьютере.

За основу был взят репозиторий: <https://github.com/the-templet-project/templet>, Который предназначен для работы с Python версией 2.7

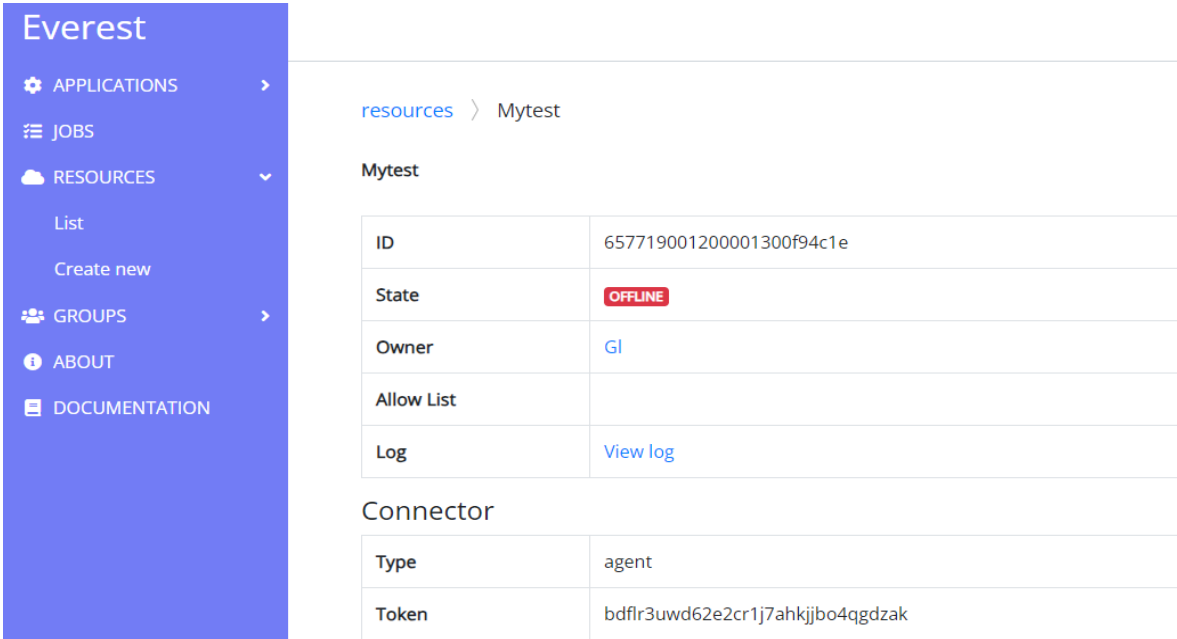
Доступ к платформе Everest осуществляется через сервис MyBinder.

Сервис **mybinder** предоставляют доступ к Jupiter Notebook и JupyterLab со всеми установленными библиотеками, так что пользователь может изучать и запускать созданный вами проект, имея под рукой только браузер. Вы также можете создать свою собственную систему с помощью JupyterHub, который, в частности, позволяет совместно работать нескольким пользователям с одним и тем же блокнотом, аналогично Google-документам, то есть один пользователь видит в процессе, какие изменения вносятся остальными.

Для того, чтобы проверить как выполняется та или иная задача в Everest, необходимо, запустить репозиторий через mybinder. После чего откроется окно jupyterlab, где необходимо провести подключение к агенту.

Это происходит так:

1) Необходимо перейти на Everest. Открыть вкладку Resources и создать новый агент ресурс. Получить токен этого ресурса и ID



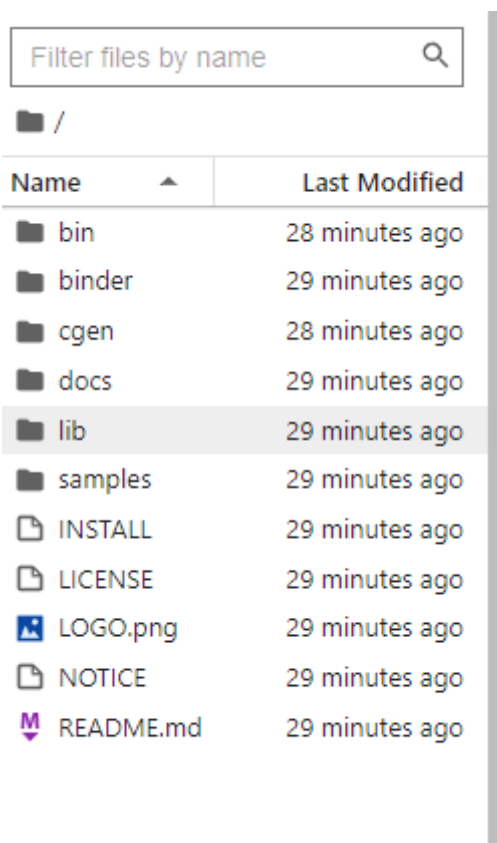
The screenshot shows the Everest web interface. On the left is a blue sidebar with navigation options: APPLICATIONS, JOBS, RESOURCES (expanded), GROUPS, ABOUT, and DOCUMENTATION. The main content area shows the 'resources' page for 'Mytest'. It contains a table with the following data:

Mytest	
ID	657719001200001300f94c1e
State	OFFLINE
Owner	GI
Allow List	
Log	<a href="#">View log</a>

Below the table is a section for the 'Connector' with the following data:

Type	agent
Token	bdf1r3uwd62e2cr1j7ahkjjbo4qgdzak

2) Когда откроется jupyterlab, необходимо перейти во вкладку samples/evrstagnt и откыть everest.ipynb.



The image shows a JupyterLab notebook with the following content:

```
Installing the Everest agent:

[ ]: !~/bin/everest.sh

Overwriting the configuration file from backup:

[ ]: !cp ~/everest_agent/everest_agent/agent.conf.default ~/everest_agent/conf/agent.conf; echo -n AGENTTOKEN > agent-token.txt

Writing a new Everest agent token to the configuration file, saving resource ID:

[ ]: #include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
{
    string old_token = "AGENTTOKEN";
    (ifstream file("agent-token.txt"); if(file) file >> old_token);
    cout << "Please enter the Everest agent Token:" << endl;
    string new_token; cin >> new_token;
    ofstream file("agent-token.txt"); file << new_token;
    string sed_command = "sed -i -e 's'"old_token"+"new_token"' ~/everest_agent/conf/agent.conf";
    system(sed_command.c_str());
    cout << "Please enter the Everest agent ID:" << endl;
    string agent_ID; cin >> agent_ID;
    ofstream idfile("agent-ID.txt"); idfile << agent_ID;
}

Starting the Everest agent:

[ ]: #include <cstdlib>
#include <fstream>
#include <iostream>
using namespace std;
{
    ifstream file("../everest_agent/agent.pid"); if(file){cout << "Everest agent is already running, please stop it first.";return;}
    system("~/everest_agent/bin/start.sh -c ~/everest_agent/conf/agent.conf");
    ifstream file("../everest_agent/agent.pid"); if(file){cout << "Everest agent is running.";}
}

Opening a session, getting a session token:
```

3) После этого необходимо запустить подключение к агенту. При поочередном запуске вас попросит ввести токен и ID, полученные ранее. Необходимо их ввести и подключение произойдет.

```
[*]: #include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
{
    string old_token = "AGENTTOKEN";
    {ifstream file("agent-token.txt"); if(file) file >> old_token;}
    cout << "Please enter the Everest agent Token:" << endl;
    string new_token; cin >> new_token;
    ofstream file("agent-token.txt"); file << new_token;
    string sed_command = "sed -i -e 's="+old_token+"="+new_token+"' ~/everest_agent/conf/agent.conf";
    system(sed_command.c_str());
    cout << "Please enter the Everest agent ID:" << endl;
    string agent_ID; cin >> agent_ID;
    ofstream idfile("agent-ID.txt"); idfile << agent_ID;
}
}
```

Please enter the Everest agent Token:

bdflr3uwd62e2cr1j7ahkjjbo4qgdzak

```
[*]: #include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
{
    string old_token = "AGENTTOKEN";
    {ifstream file("agent-token.txt"); if(file) file >> old_token;}
    cout << "Please enter the Everest agent Token:" << endl;
    string new_token; cin >> new_token;
    ofstream file("agent-token.txt"); file << new_token;
    string sed_command = "sed -i -e 's="+old_token+"="+new_token+"' ~/everest_agent/conf/agent.conf";
    system(sed_command.c_str());
    cout << "Please enter the Everest agent ID:" << endl;
    string agent_ID; cin >> agent_ID;
    ofstream idfile("agent-ID.txt"); idfile << agent_ID;
}
}
```

Please enter the Everest agent Token:

bdflr3uwd62e2cr1j7ahkjjbo4qgdzak

Please enter the Everest agent ID:

657719001200001300f94c1e

```
[4]: #include <cstdlib>
#include <fstream>
#include <iostream>
using namespace std;
{ifstream file("../everest_agent/agent.pid"); if(file){cout << "Everest agent is already running, please stop it first.;"return;}}
system("~/everest_agent/bin/start.sh -c ~/everest_agent/conf/agent.conf");
{ifstream file("../everest_agent/agent.pid"); if(file){cout << "Everest agent is running.;"}}
```

Everest agent is running.

4) После подключения, State должен измениться на online и версия Python должна будет быть 3.9

ID	657719001200001300f94c1e
State	<b>ONLINE</b>
Owner	<a href="#">GI</a>
Allow List	
Log	<a href="#">View log</a>

## Connector

Type	agent
Token	bdf1r3uwd62e2cr1j7ahkjbo4qgdzak

## System Info

System	Linux
Agent Version	2.2
Python Version	3.9.18
Node	jupyter-glor27-2dtemplet-2d5111fp6y
Release	5.15.0-72-generic
Version	#79-Ubuntu SMP Wed Apr 19 08:22:18 UTC 2023
Machine	x86_64
Processor	x86_64

5) После этого можно выполнять задачи через платформу Everest. Для этого перейдите во вкладку Application/List.

Для примера запустим задачу Sleep.

The screenshot shows the Everest application interface. On the left is a blue sidebar with navigation options: APPLICATIONS, JOBS, RESOURCES, GROUPS, ABOUT, and DOCUMENTATION. The main content area displays a list of applications. The 'Sleep' application is selected, showing its details: 'Sleeps like there's no tomorrow.', 3 stars, 845 jobs, version 0.1.0, updated on 02 Feb 2020 01:40:45, and owned by 'sol'. Other applications visible include 'Coarse-grained B&B' and 'Generic Run'.

б) Для того, чтобы выполнить задачу необходимо нажать на нее и выбрать Submit Job. Там необходимо ввести задержку и выбрать ресурс, который мы создавали заранее. После этого нажать submit

The screenshot shows the 'Submit Job' page for the 'Sleep' application. The 'Job Name' is 'Sleep' and 'Duration (seconds)' is '7'. The 'Resources' section shows 'Mytest' selected. The 'Job description' section has a text area with 'Write something...' and a 'Submit' button. The page also includes tabs for 'About', 'Parameters', 'Submit Job', and 'Discussion', and buttons for 'Preview request' and 'Submit'.

7) Спустя 7 секунд нам выведет результат Done, что наша программа отработала.

The screenshot shows the job completion page for 'Sleep'. It displays 'Submitted by GI Application Sleep (version 0.1.0)', 'Runtime 11 Dec 2023 18:26:27 - 11 Dec 2023 18:26:37', and 'Data Size 0.00 KB'. The 'Outputs' section shows 'stderr'. The page also includes buttons for 'Delete' and 'Resubmit', and a 'DONE' status indicator.

Для того чтобы установить Python 3.9 для работы Everest, мы изменили репозиторий github.

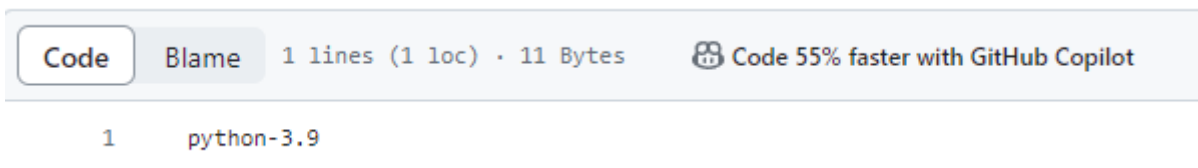
```
#!/bin/bash

mamba install -y -c conda-forge xeus-cling
mamba create -y --name everest python=3.9 tornado=4.5.3 six=1.16

cd ~/cgen
g++ skel.cpp -o skel
g++ cgen.cpp lexer.cpp parse.cpp -o cgen
g++ acta.cpp -o acta

mv skel ~/bin/skel
mv cgen ~/bin/cgen
mv acta ~/bin/acta

chmod 755 ~/bin/skel
chmod 755 ~/bin/cgen
chmod 755 ~/bin/acta
chmod 755 ~/bin/everest.sh
```



```
1 python-3.9
```

Это изменения версии, который мы провели в папке binder с файлами – postBuild. Основным изменением стало установкой библиотеки six.

Six — библиотека совместимости Python 2 и 3. Она содержит функции, сглаживающие различия в версиях Python, и позволяет писать код, совместимый с обеими версиями Python.

Была установлена новейшая версия six на данный момент, версия 1.16

Также была установлена новейшая рабочая версия Tornado = 4.5.3

```
Code Blame 12 lines (7 loc) · 228 Bytes Code 55% faster with GitHub Copilot

1  #!/bin/sh
2
3
4
5  git clone https://gitlab.com/everest/agent.git ~/everest_agent
6
7  cd ~/everest_agent
8  mkdir ./conf
9  cp everest_agent/agent.conf.default conf/agent.conf
10
11 cp -f ~/bin/start4python3.sh bin/start.sh
12 chmod 755 bin/start.sh
```

```
Code Blame 15 lines (13 loc) · 437 Bytes Code 55% faster with GitHub Copilot

1  #!/bin/bash
2  BIN_DIR=`dirname "$(cd ${0%/*} && echo $PWD/${0##*/})"`
3  AGENT_HOME=`dirname "$BIN_DIR"`
4  cd $AGENT_HOME
5
6  PID_FILE=agent.pid
7
8  if [ -f $PID_FILE ] && [ -f /proc/$(cat $PID_FILE)/status ]; then
9      echo "Agent is already running"
10 else
11     PYTHON=/srv/conda/envs/everest/bin/python3
12     PYTHONPATH=$(dirname $0)/..:$PYTHONPATH $PYTHON -m everest_agent.start 2>errors.txt $@ &
13     echo $! > $PID_FILE
14     echo "Agent is started"
15 fi
```

Здесь изменения, которые мы произвели с файлами start4python3.sh и everest.sh в папке bin.

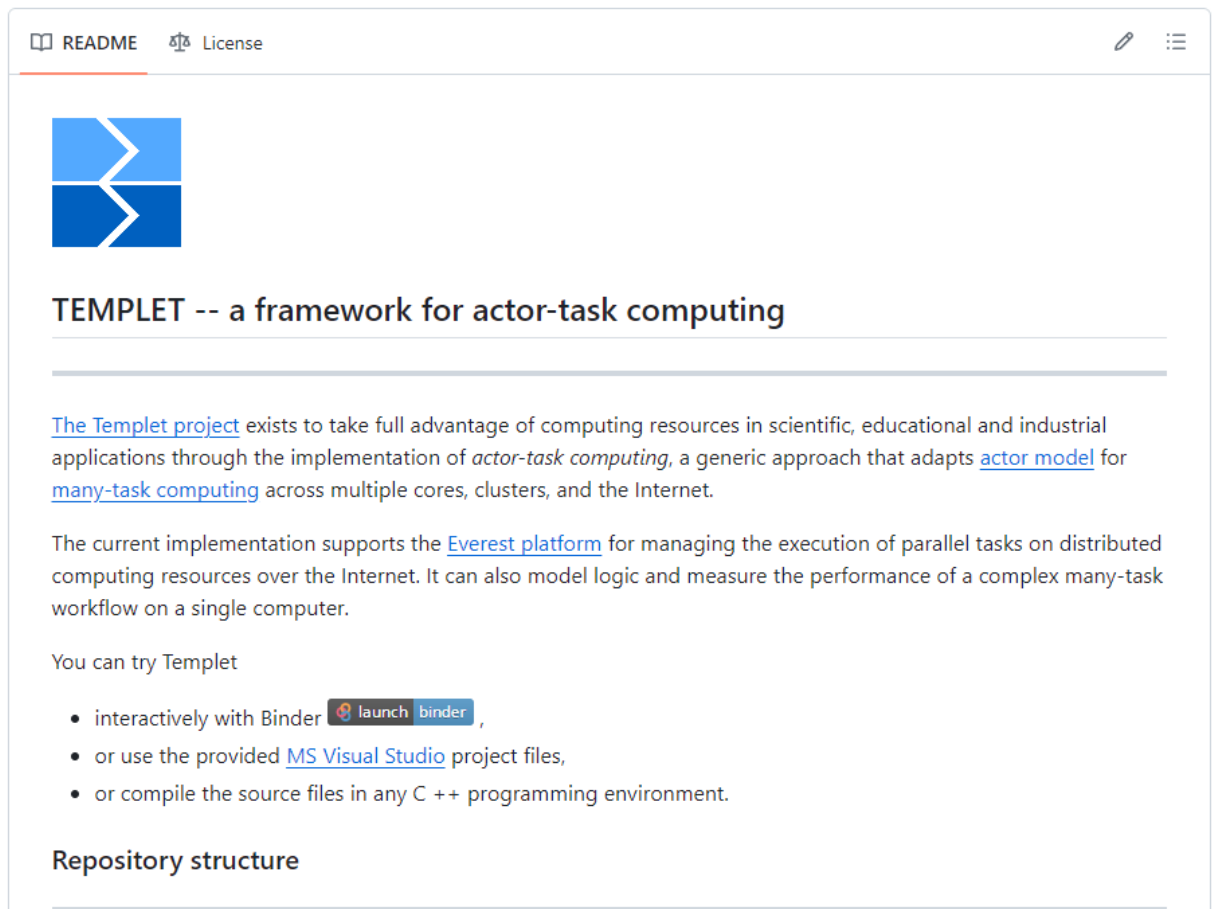
В результате мы получили свой репозиторий, который запускает агент на python 3.9

Ниже ссылка на репозиторий. Произвести запуск вы можете вставив ссылку на официальном сайте Mybinder. Или в репозитории, нажав на значок mybinder, он автоматически перекинет на страничку запуска этого репозитория.



Ссылка - <https://github.com/Glor27/templet.git>

Страничка репозитория.



The screenshot shows the GitHub README for the 'templet' repository. At the top, there are tabs for 'README' and 'License'. Below the tabs is a blue logo consisting of two stylized arrows pointing right. The main heading is 'TEMplet -- a framework for actor-task computing'. The text describes the project's purpose: to take full advantage of computing resources in scientific, educational, and industrial applications through the implementation of *actor-task computing*, which adapts the *actor model* for *many-task computing* across multiple cores, clusters, and the Internet. It also mentions that the current implementation supports the *Everest platform* for managing parallel tasks on distributed computing resources over the Internet. A section titled 'You can try Templet' lists three options: interacting with Binder (with a 'launch binder' button), using provided MS Visual Studio project files, or compiling source files in any C++ programming environment. The section 'Repository structure' is partially visible at the bottom.

Нажать на значок launchbinder.

В первый раз будет грузить долго, но во время этой загрузки используется сохранени образа, который будет использоваться дальнейших запусков.