

**Міністерство освіти і науки України**  
**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №1

з дисципліни «Кросплатформне програмування» на тему:

«Створення розподіленої системи на базі HTCondor»

**Виконав:**

Студент групи КНТ-122

О. А. Онищенко

**Прийняли:**

Викладач:

А. В. Тіменко

2024

# СТВОРЕННЯ РОЗПОДІЛЕНОЇ СИСТЕМИ НА БАЗІ HTCONDOR

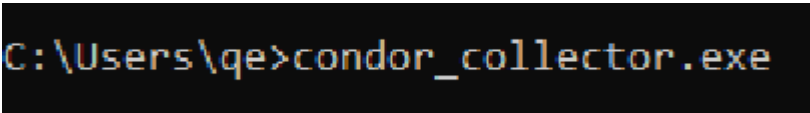
## Мета роботи

Навчитися розгортати та конфігурувати обчислювальний кластер на базі системи HTCondor.

## Результати виконання

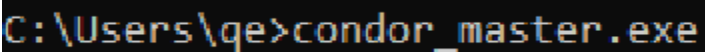
### 1 Налаштування сервера

Запустимо необхідні фонові процеси після встановлення HTCondor:



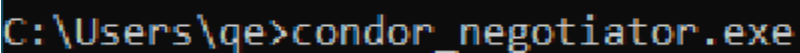
```
C:\Users\qe>condor_collector.exe
```

Рисунок 1.1 – Запуск процесу condor\_collector



```
C:\Users\qe>condor_master.exe
```

Рисунок 1.2 – Запуск процесу condor\_master

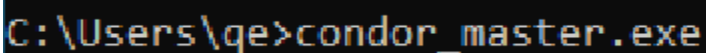


```
C:\Users\qe>condor_negotiator.exe
```

Рисунок 1.2 – Запуск процесу condor\_negotiator

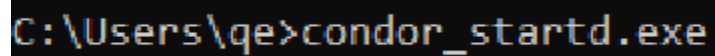
### 2 Налаштування обчислювального комп'ютера

Запустимо необхідні фонові процеси після встановлення HTCondor:



```
C:\Users\qe>condor_master.exe
```

Рисунок 2.1 – Запуск процесу condor\_master

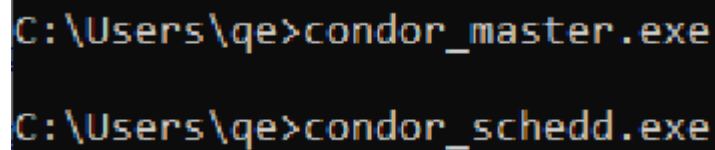


```
C:\Users\qe>condor_startd.exe
```

Рисунок 2.2 – Запуск процесу condor\_startd

### 3 Налаштування користувача

Запустимо необхідні фонові процеси після встановлення HTCondor:



```
C:\Users\qe>condor_master.exe  
C:\Users\qe>condor_schedd.exe
```

Рисунок 3.1 – Запуск процесів condor\_master та condor\_schedd

Встановимо gcc:

GCC-13.2.0-64 [64-bit]

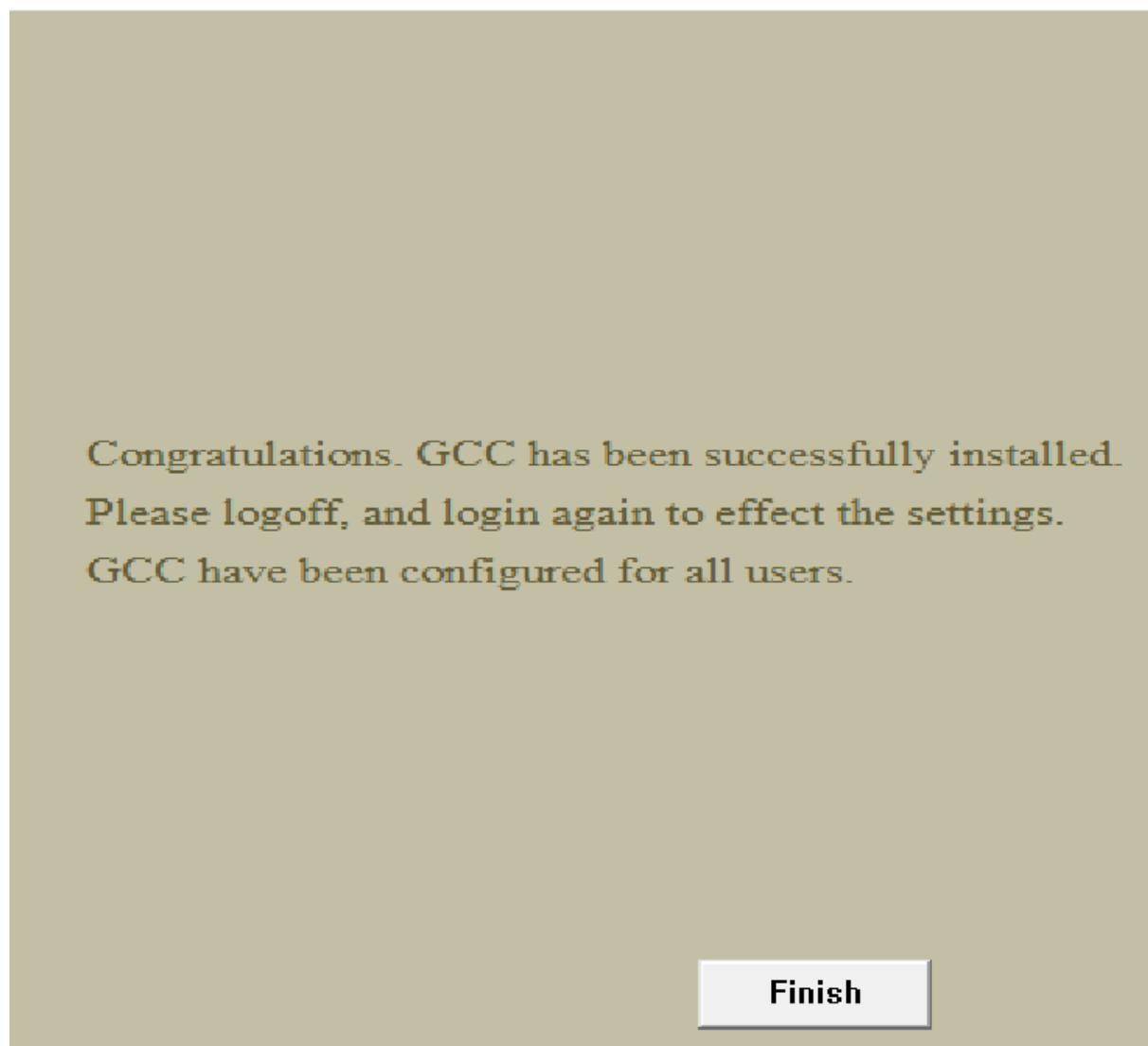



Рисунок 3.2 – Результат встановлення gcc

Створимо необхідні файли:

 simple.c - Notepad  
File Edit Format View Help

---

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <synchapi.h>
int main(int argc, char **argv)
{
    int sleep_time;
    int input;
    int failure;
    if (argc != 3)
    {
        printf("Usage: simple <sleep-time> <integer>\n");
        failure = 1;
    }
    else
    {
        sleep_time = atoi(argv[1]);
        input = atoi(argv[2]);
        printf("Thinking really hard for %d seconds...\n", sleep_time);
        Sleep(sleep_time);
        printf("We calculated: %d\n", input * 2);
        failure = 0;
    }
    return failure;
}
```

Рисунок 3.3 – Створення файлу C


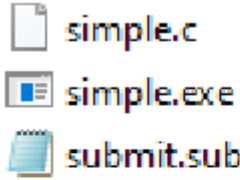

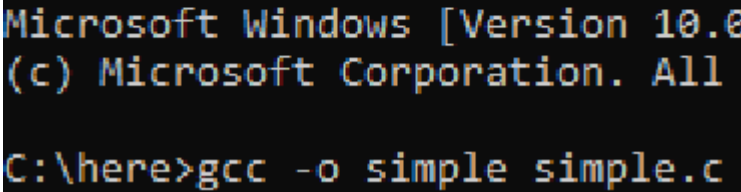
 submit.sub - Notepad  
File Edit Format View Help  
Universe = vanilla  
Executable = simple.exe  
Arguments = 4 10  
Log = simple.log  
Output = simple.\$(Process).out  
Error = simple.\$(Process).error  
Queue  
Arguments = 4 11  
Queue  
Arguments = 4 12  
Queue|

Рисунок 3.4 – Створення фалу sub

Скомпілюємо файл C:

  
 C:\Windows\System32\cmd.exe  


```
Microsoft Windows [Version 10.0  
(c) Microsoft Corporation. All  
C:\here>gcc -o simple simple.c
```

Рисунок 3.5 – Результат компілювання файлу C

Запускаємо роботу:

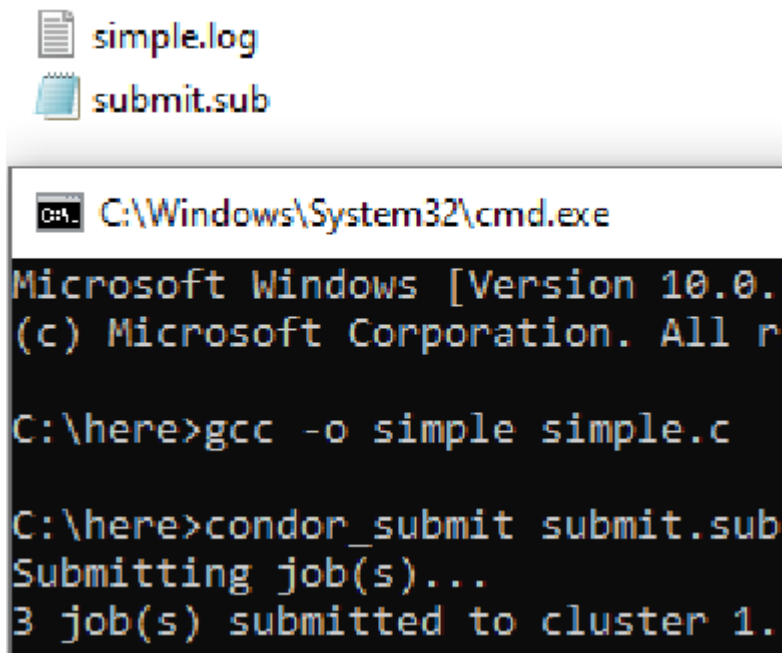


Рисунок 3.6 – Результат запуску роботи

Починаємо моніторинг:

```
C:\here>condor_q -sub qe
Error: Couldn't contact the condor_collector on Nes.

Extra Info: the condor_collector is a process that runs on the central
manager of your Condor pool and collects the status of all the machines and
jobs in the Condor pool. The condor_collector might not be running, it might
be refusing to communicate with you, there might be a network problem, or
there may be some other problem. Check with your system administrator to fix
this problem.

If you are the system administrator, check that the condor_collector is
running on Nes, check the ALLOW/DENY configuration in your condor_config, and
check the MasterLog and CollectorLog files in your log directory for possible
clues as to why the condor_collector is not responding. Also see the
Troubleshooting section of the manual.
```

Рисунок 3.7 – Результат початку моніторингу

Відображаємо результуючі файли:

```
C:\here>dir simple*.out
Volume in drive C has no label.
Volume Serial Number is FCEF-D628

Directory of C:\here

File Not Found
```

Рисунок 3.8 – Результат відображення результуючих файлів

Відкриваємо файл журналу:

```
C:\here>type simple.log
000 (001.000.000) 2024-03-01 19:57:23 Job submitted from host: <10.0.2.15:9618?addrs=10.0.2.15-9618&alias=user&noUDP&sock=schedd_6472_0ede>
...
000 (001.001.000) 2024-03-01 19:57:23 Job submitted from host: <10.0.2.15:9618?addrs=10.0.2.15-9618&alias=user&noUDP&sock=schedd_6472_0ede>
...
000 (001.002.000) 2024-03-01 19:57:23 Job submitted from host: <10.0.2.15:9618?addrs=10.0.2.15-9618&alias=user&noUDP&sock=schedd_6472_0ede>
...
```

Рисунок 3.9 – Результат відкриття файлу журналу

Переглядаємо результуючі файли:

```
C:\here>type simple.0.out
The system cannot find the file specified.

C:\here>type simple.1.out
The system cannot find the file specified.

C:\here>type simple.2.out
The system cannot find the file specified.
```

Рисунок 3.10 – Результат перегляду результуючих файлів

## Висновки



Таким чином, ми навчилися розгортати та конфігурувати обчислювальний кластер на базі системи HTCondor.

## **Контрольні питання**

### **Що таке HTCondor?**

HTCondor - це система високопропускних обчислень (HTC), яка дозволяє розподілено обробляти завдання через мережу комп'ютерів. Вона призначена для обробки великомасштабних, паралельних і розподілених обчислювальних завдань, що робить її корисною для наукових досліджень та інших застосувань, які вимагають значних обчислювальних ресурсів.

### **Мета використання проміжного програмного забезпечення**

Метою використання проміжного програмного забезпечення в HTCondor є полегшення комунікації між різними компонентами системи. Проміжне програмне забезпечення виконує роль сполучної ланки, дозволяючи виконувати завдання, планувати їх виконання та розподіляти роботу по мережі. Воно керує ресурсами, визначає пріоритетність завдань і гарантує, що завдання виконуються ефективно.

### **Які необхідно запустити команди для роботи HTCondor?**

Щоб запустити HTCondor, зазвичай потрібно встановити програмне забезпечення HTCondor на локальній машині і на всіх віддалених машинах, які будуть використовуватися для обчислень. Після цього слід запустити служби HTCondor, які є фоновими процесами, що керують системою. Конкретні команди для запуску HTCondor можуть відрізнятися

залежно від операційної системи і конкретних налаштувань, але, як правило, слід використовувати такі команди, як `condor_startup` для запуску сервісів і `condor_submit` для створення завдань.

### **Які існують аналоги HTCondor?**

Аналогами HTCondor є інші обчислювальні системи з високою пропускнуою здатністю, такі як Sun Grid Engine (SGE), LSF та PBS Pro. Ці системи також надають можливості розподілених обчислень і використовуються для подібних цілей, таких як запуск великомасштабних симуляцій або обробка великих наборів даних.