

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет Запорізька політехніка

МЕТОДИЧНІ ВКАЗІВКИ

**до лабораторних робіт з дисципліни
"Кросплатформне програмування"**

для студентів спеціальностей 121 та 122
всіх форм навчання

2023

Методичні вказівки до лабораторних робіт з дисципліни "Кросплатформне програмування" для студентів спеціальностей 121 та 122 всіх форм навчання / Укл. С.Ю. Скрупський – Запоріжжя: НУ "Запорізька політехніка", 2023. – 34 с.

Укладачі: С.Ю. Скрупський, к.т.н., доцент

Рецензент:

Відповідальний
за випуск: С.Ю. Скрупський, к.т.н., доцент

Затверджено:
на засіданні кафедри
Протокол № 12 від 09.06.2023

Рекомендовано до видання
НМК ФКНТ

ЗМІСТ

ЛАБОРАТОРНА РОБОТА – Створення розподіленої системи на базі Condor	4
ЛАБОРАТОРНА РОБОТА – Розгортання платформи <i>ownCloud</i>	24
ЛІТЕРАТУРА.....	34

ЛАБОРАТОРНА РОБОТА

Створення розподіленої системи на базі HTCondor

Мета роботи – навчитися розгортати та конфігурувати обчислювальний кластер на базі системи *HTCondor*.

4.1 Теоретичні відомості

HTCondor - спеціалізована система управління навантаженням для обчислювальних робіт. Як і інші повнофункціональні пакетні системи, HTCondor забезпечує механізм черги завдань, політику планування, схему пріоритетів, моніторинг ресурсів та управління ресурсами. Користувачі подають свої послідовні або паралельні завдання HTCondor, HTCondor розміщує їх у черзі, вибирає, коли і де виконувати завдання на основі політики, ретельно стежить за їх ходом і в кінцевому підсумку інформує користувача про його завершення.

HTCondor є програмним забезпеченням з відкритим вихідним кодом для крупнозернистого розподіленого розпаралелювання обчислень, ресурс'ємних завдань. HTCondor працює на Linux, Unix, Mac OS X, FreeBSD та Microsoft Windows. HTCondor можуть інтегрувати обидва виділені ресурсів (стійку кластера) і не призначені для настільних машин (цикл очищення) в одну обчислювальну середу.

Апаратна архітектура системи включає три типи комп'ютерів (рис. 4.1), об'єднаних в єдиний пул Condor: центральний сервер, виконуюча (вузлова) та клієнтська машини.



Рисунок 4.1 – Архітектура Condor

Набір компіляторів GNU (GNU Compiler Collection, GCC) — набір компіляторів для різних мов програмування. GCC — вільне програмне забезпечення, розроблене Фондом Вільних Програм під ліцензією GNU GPL та GNU LGPL, і є ключовою складовою набору знарядь розробки GNU (GNU development toolchain). Це стандартний компілятор для вільних Юнікс-подібних операційних систем.

Виконання роботи полягає в інсталяції та налаштування платформи *HTCondor* на прикладі збірки *HTCondor-8.8.9* або новішої.

Набір програмних платформ і засобів, для яких було перевірено працездатність *HTCondor* та на основі яких рекомендується виконувати дану роботу, наступний:

- операційна система *Microsoft Windows 10*;
- система *HTCondor*;
- набір компіляторів *gcc-9.2.0-64*.

4.2 Порядок виконання роботи

Примітка: виконання цієї лабораторної роботи потребує три ПК, на яких будуть знаходитись відповідно серверна, виконуюча (вузлова) та клієнтська частини.

4.2.1 Спочатку потрібно завантажити *HTCondor* на всіх ПК. Переходимо на сторінку HTCondor <https://research.cs.wisc.edu/htcondor/> та натискаємо Download.

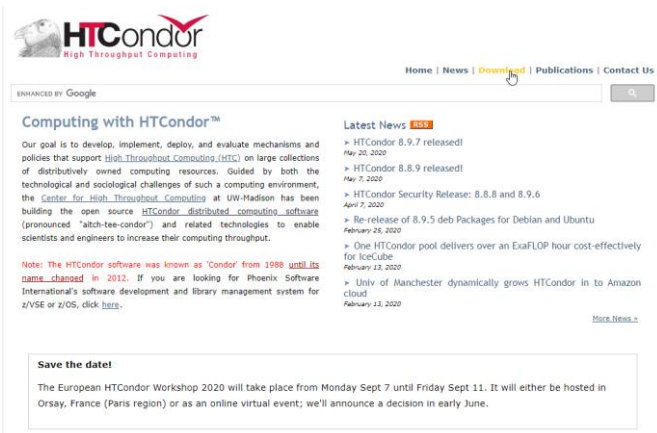


Рисунок 4.2 – Головна сторінка *Condor*

4.2.1.1 Далі обираємо версію 8.8.9 (або актуальнішу), та завантажуємо.

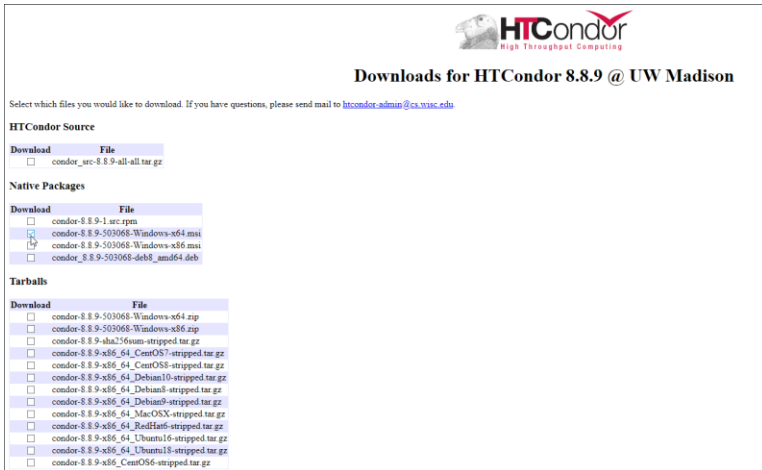


Рисунок 4.2 – Сторінка Download.

4.2.2 Налаштування серверного комп'ютера

4.2.2.1 Для початку встановимо *HTCondor*, натискаємо Next.

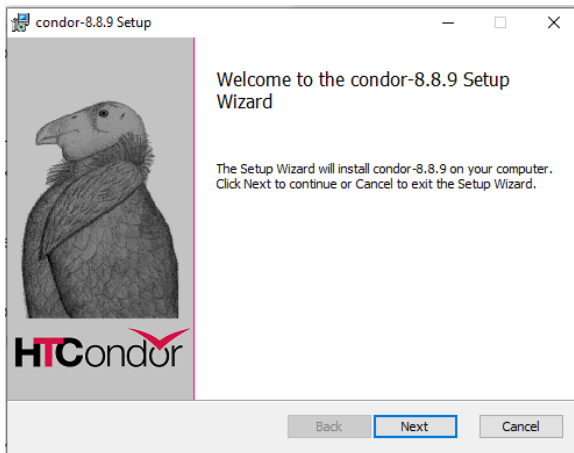


Рисунок 4.3 – Процес встановлення Condor.

4.2.2.2 Перший крок встановлення HTCondor - це вітальний екран та ліцензійна угода.

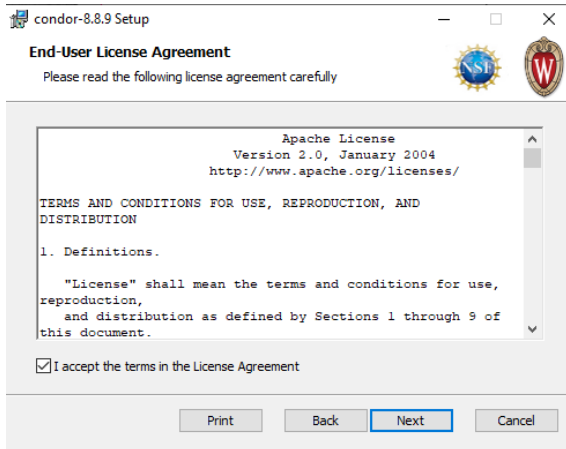


Рисунок 4.4 – Ліцензійна угода

4.2.2.3 Конфігурацію HTCondor потрібно встановити *Create a new HTCondor Pool*. Та вказати ім'я для пулу.

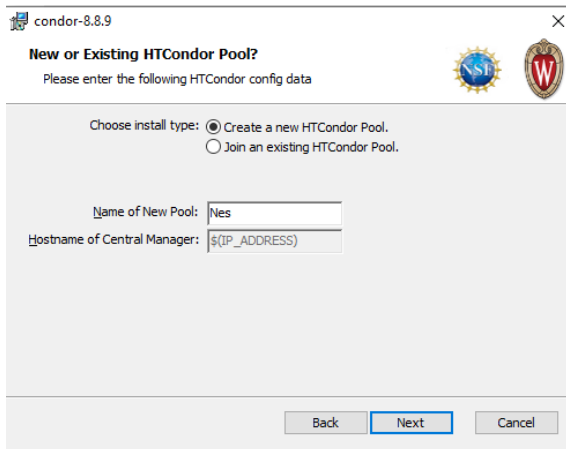


Рисунок 4.5 – Ім'я пулу

4.2.2.4 Натискаємо Do not run jobs on this machine, далі Next.

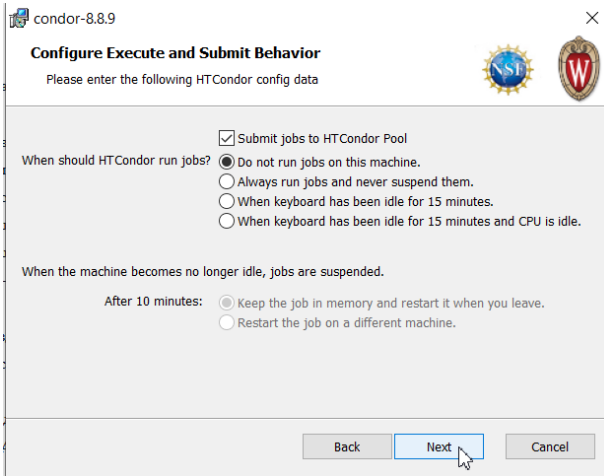


Рисунок 4.6 – Роль машини

4.2.2.5 Домен облікового запису залишаємо порожнім.

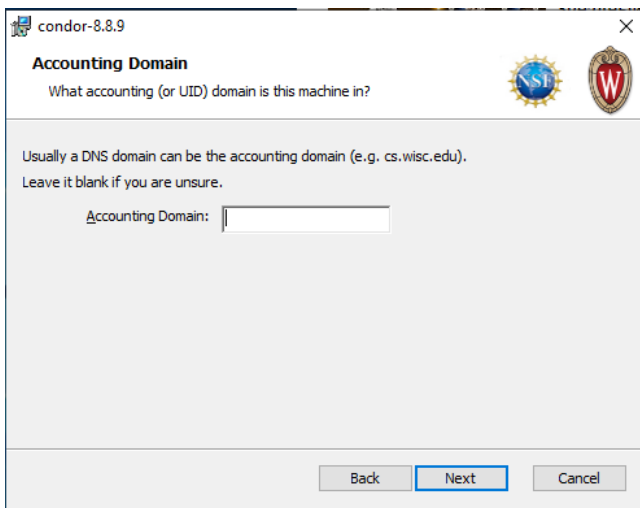


Рисунок 4.7 – Домен облікового запису

4.2.2.6 Натискаємо Next.

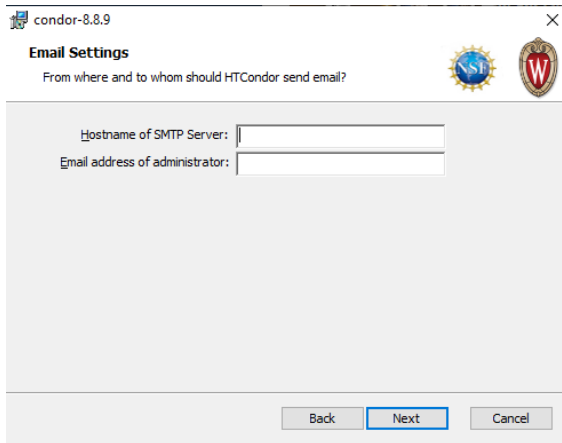


Рисунок 4.8 – Налаштування електронної пошти

4.2.2.7 Вказуємо директорію до jdk.

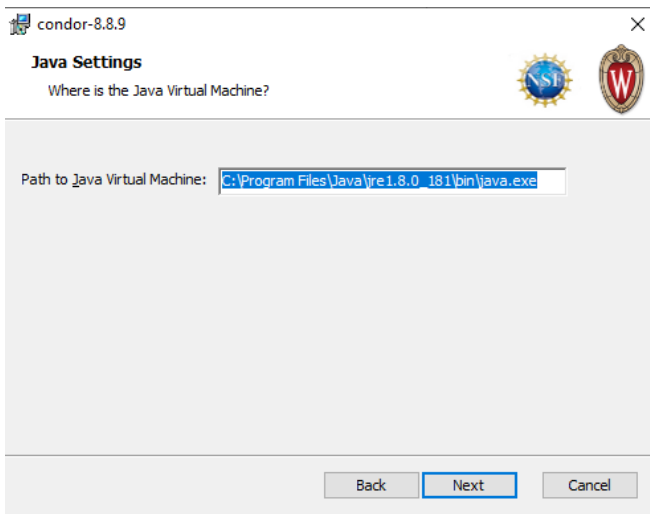


Рисунок 4.9 – Директорія до jdk

4.2.2.8 В полі Host with Write access: встановлюємо * для того щоб всі машини в пулі мали доступ для запису.

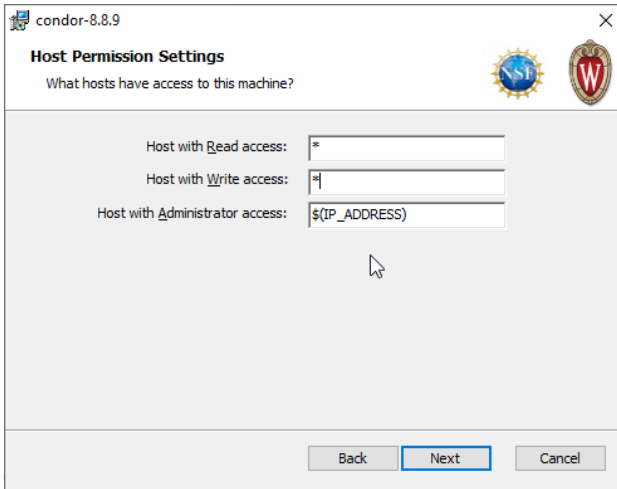


Рисунок 4.10 – Налаштування доступу хоста

4.2.2.9 Натискаємо Next.

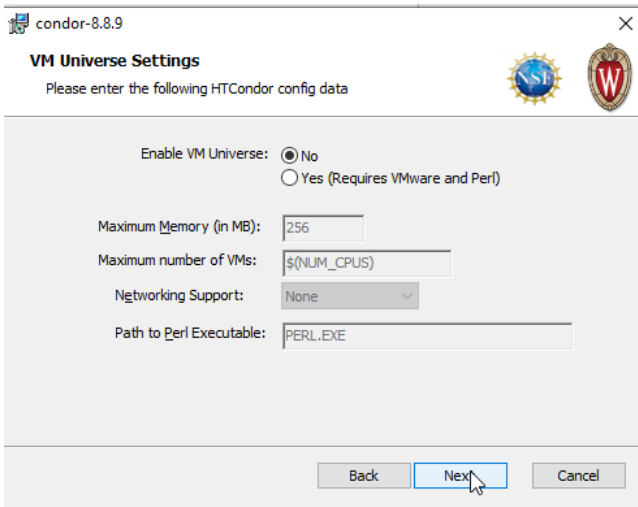


Рисунок 4.11 – Налаштування VM

4.2.2.10 Вибираємо директорію для встановлення.

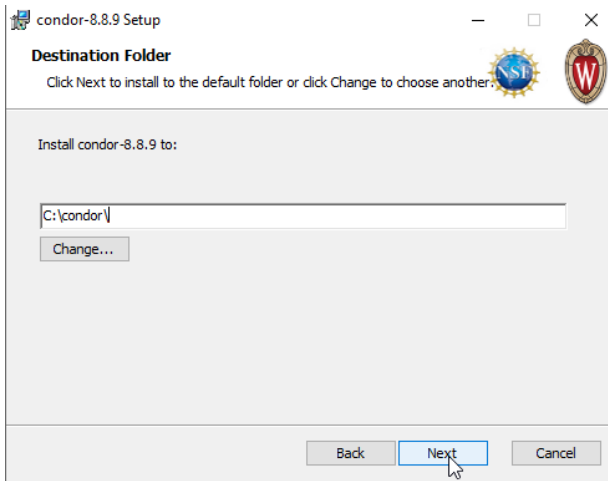


Рисунок 4.12 – Директорія для встановлення

4.2.2.11 Натискаємо install та чекаємо встановлення HTCCondor.

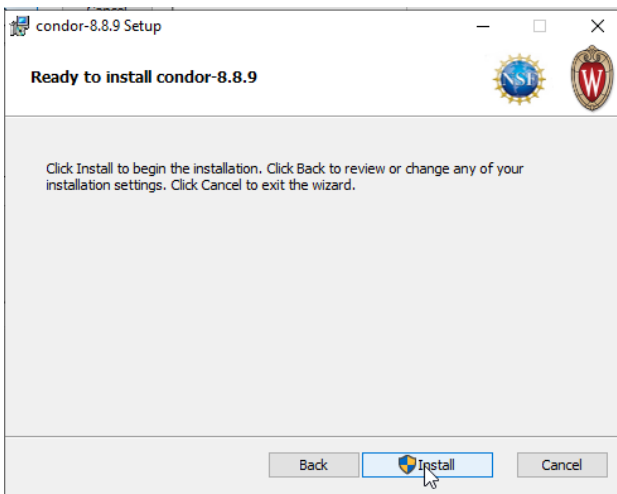


Рисунок 4.13 – Встановлення

4.2.2.12 Перезавантажуємо систему.

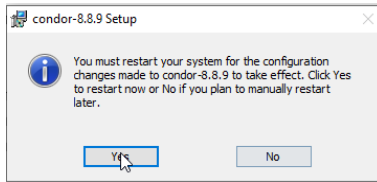


Рисунок 4.14 – Перезавантаження системи

4.2.2.13 Після завершення установки HTCondor служба HTCondor вже запущена. Далі потрібно запуснути наступні задачі:

– `condor_master`. Демон, який контролює роботу всіх демонів Condor, запущених на кожній машині пулу і виконує адміністративні команди системи.

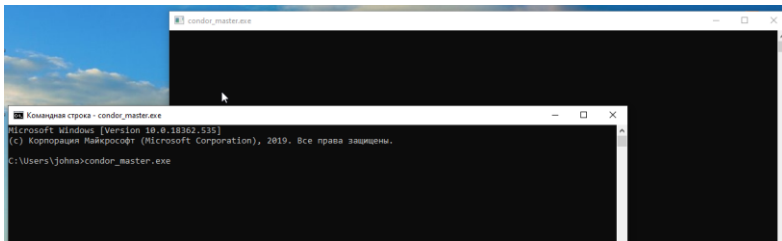


Рисунок 4.15 – Запуск задачі condor_master

– `condor_negotiator`. Демон управління станом Condor. Періодично демон запускає цикл узгодження, протягом якого збираються дані про стан ресурсів.

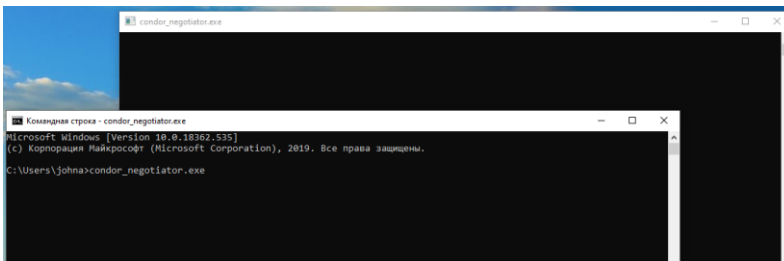


Рисунок 4.16 – Запуск задачі condor_negotiator

– condor_collector. Збір інформації про стан пулу Condor. Всі інші демони періодично посилають дані про свій стан.

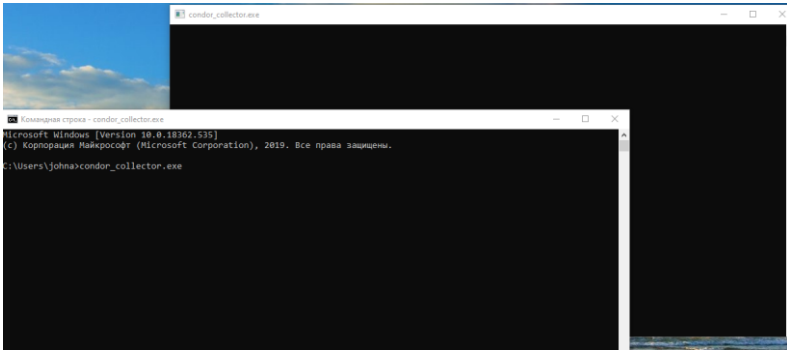


Рисунок 4.17 – Запуск задачі condor_collector

4.2.3 Налаштування обчислювального комп'ютера.

4.2.3.1 Для встановлення HTCondor на обчислювальному комп'ютері необхідно повторити дії, як у пункті 4.2.2.1 – 4.2.2.2. Наступним кроком необхідно встановити значення Join an existing HTCondor Pool. Та в полі Hostname of Central Manager вести ім'я пулу що водили при встановленні на серверному комп'ютері, та натискаємо Next.

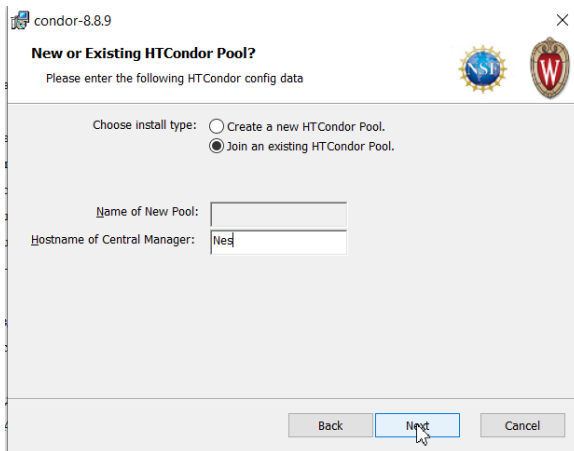


Рисунок 4.18 – Додання в існуючий пул

4.2.3.2 Натискаємо Always run jobs and never suspend them далі Next, далі встановлюємо згідно пунктів 4.2.2.5 – 4.2.2.12.

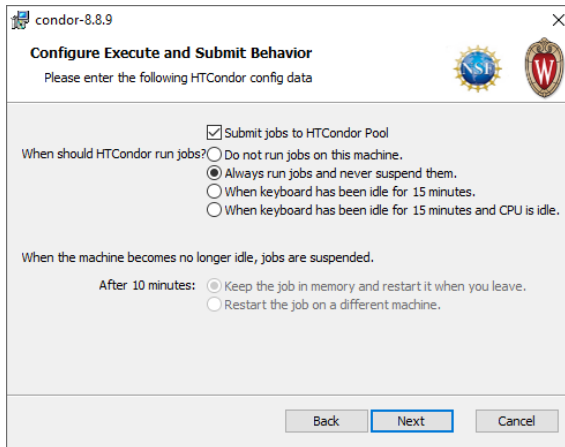


Рисунок 4.19 – Роль машини

4.2.3.3 Після завершення установки HTCondor служба HTCondor вже запущена. Та потрібно запуснути наступні задачі:

– condor_master. Демон, який контролює роботу всіх демонів Condor, запущених на кожній машині пулу і виконує адміністративні команди системи.

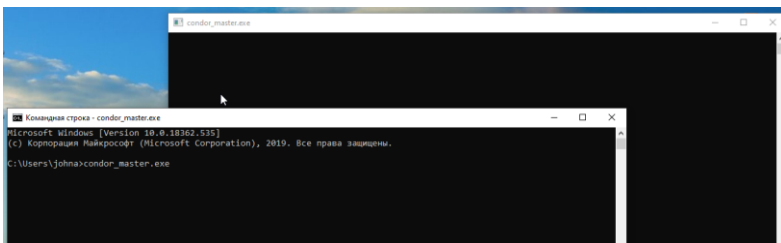


Рисунок 4.20 – Запуск задачі condor_master

– condor_startd. Демон для подання ресурсів в пулі. В разі її готовності до виконання завдання запускає демон condor_starter.

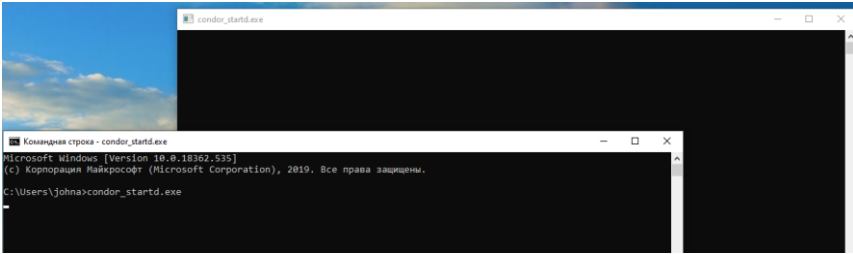


Рисунок 4.21 – Запуск задачі condor_startd

4.2.4 Створення та налаштування клієнту.

4.2.4.1 Для встановлення HTCondor на клієнті необхідно повторити дії, як у пункті 4.2.2.1 – 4.2.2.2. Наступним кроком необхідно встановити значення Join an existing HTCondor Pool. Та в полі Hostname of Central Manager вести ім'я пулу що водили при встановленні на серверному комп'ютері, та далі встановлюємо згідно пунктів 4.2.2.4 – 4.2.2.12.

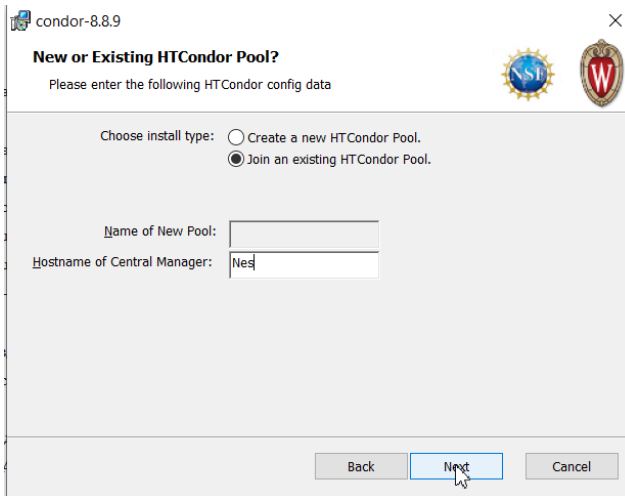


Рисунок 4.22 – Додання в існуючий пул

4.2.4.2 Після завершення установки HTCondor служба HTCondor вже запущена. Далі необхідно запустити наступні задачі:

– `condor_master`. Демон, який контролює роботу всіх демонів Condor, запущених на кожній машині пулу і виконує адміністративні команди системи.

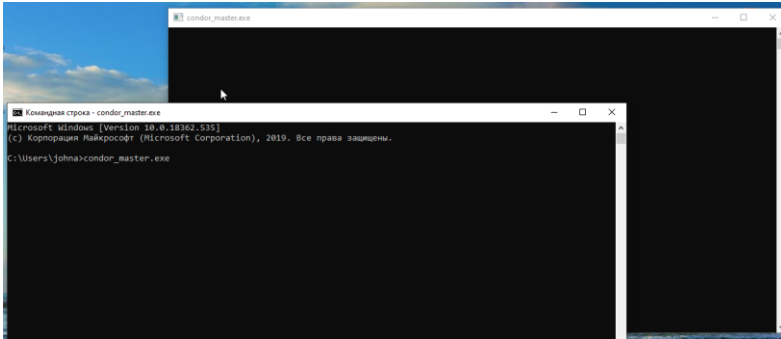


Рисунок 4.23 – Запуск задачі `condor_master`

– `condor_schedd`. Демон управління ресурсами, необхідними для пулу Condor. При запуску завдання відбувається звернення до `schedd`, який розміщує завдання в черзі. При збої демона `schedd` ніяка подальша робота неможлива.

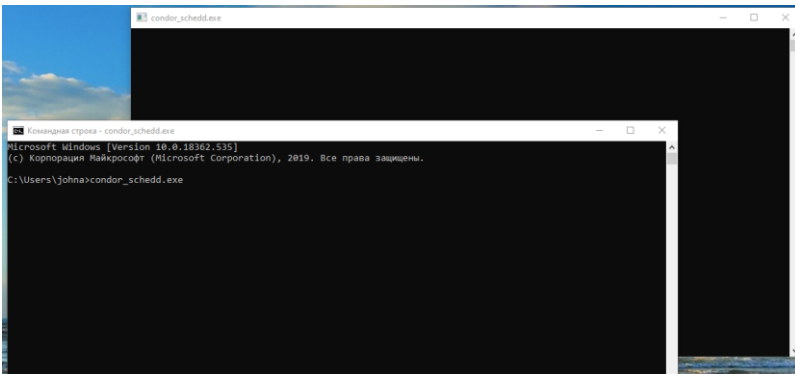


Рисунок 4.24 – Запуск задачі `condor_schedd`

4.2.5 Наступний крок встановлення *gcc-9.2.0-64*.

4.2.5.1 Погоджуємося з ліцензійною угодою.

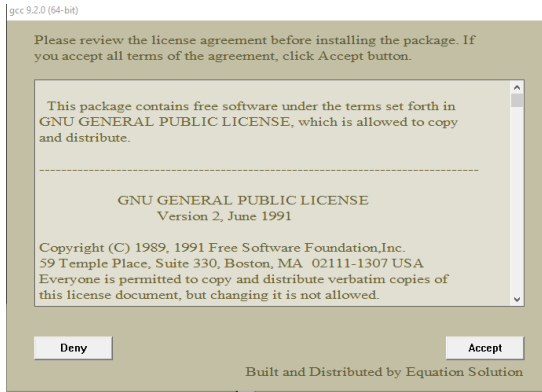


Рисунок 4.25 – Ліцензійна угода

4.2.5.2 Вибираємо директорію для встановлення.



Рисунок 4.26 – Директорія для встановлення

4.2.6 Далі необхідно створити директорію для виконання, та створити файл `C` з назвою `simple.c` вміст файлу знаходиться в лістингу 4.1

Лістинг 4.1 – Вміст файлу *simple.c*

```
#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;
}

```

4.2.6.1 Далі створюємо файл опису подання `submit.sub`. Вміст файлу знаходиться в лістингу 4.2

Лістинг 4.2 – Вміст файлу *submit.sub*

```

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

```

4.2.7 Далі необхідно відкрити командний рядок та скомпілювати файл `simple.c` наступною командою:

```
gcc -o simple simple.c
```

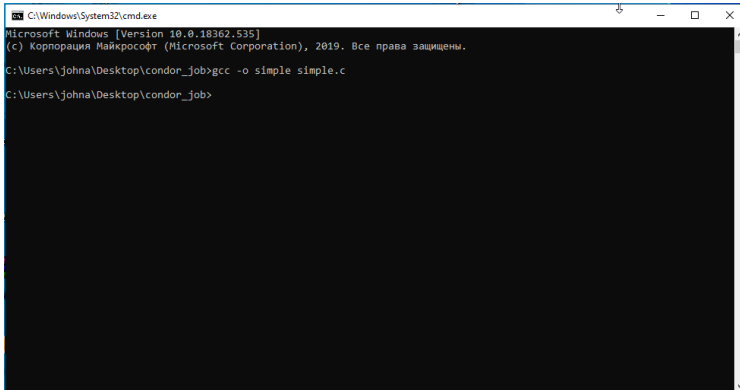


Рисунок 4.27 – Компіляція файлу `simple.c`

4.2.7.1 За допомогою наступної команди передаємо роботу HTCondor.

```
condor_submit submit.sub
```

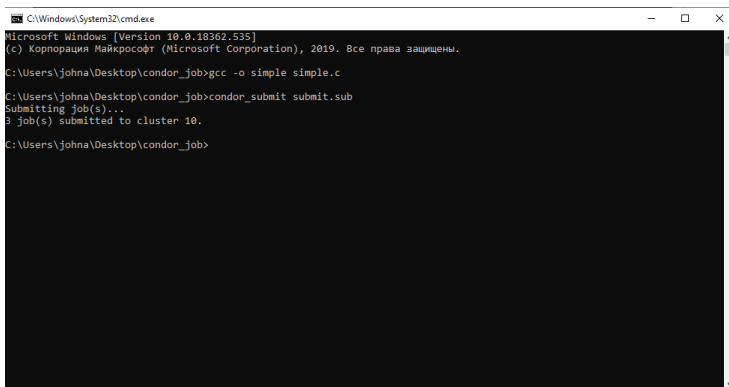


Рисунок 4.28 – Передача роботи HTCondor

4.2.7.2 Далі щоб побачити моніторинг роботи необхідно вести наступну команду

```
condor_q -sub <назва ПК>
```

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.535]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\johna\Desktop\condor_job>gcc -o simple simple.c

C:\Users\johna\Desktop\condor_job>condor_submit submit.sub
Submitting job(s)...
3 job(s) submitted to cluster 10.

C:\Users\johna\Desktop\condor_job>condor_q -sub johna

-- Submitter: johna@DESKTOP-5JUM87F : <192.168.0.105:9618?.. : DESKTOP-5JUM87F @ 05/28/20 18:25:56
OWNER BATCH_NAME SUBMITTED DONE RUN IDLE TOTAL JOB_IDS
johna ID: 10 5/28 18:25 - - 3 3 10.0-2

Total for query: 3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended
Total for all users: 3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended

C:\Users\johna\Desktop\condor_job>condor_q -sub johna

-- Submitter: johna@DESKTOP-5JUM87F : <192.168.0.105:9618?.. : DESKTOP-5JUM87F @ 05/28/20 18:26:08
OWNER BATCH_NAME SUBMITTED DONE RUN IDLE HOLD TOTAL JOB_IDS

Total for query: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
Total for all users: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended

```

Рисунок 4.29 – Моніторинг роботи

4.2.7.3 В результаті в директорії яку ми створювали в пункті 4.2.6 з'являться файли з розширенням .out, .log, .error в яких знаходяться:

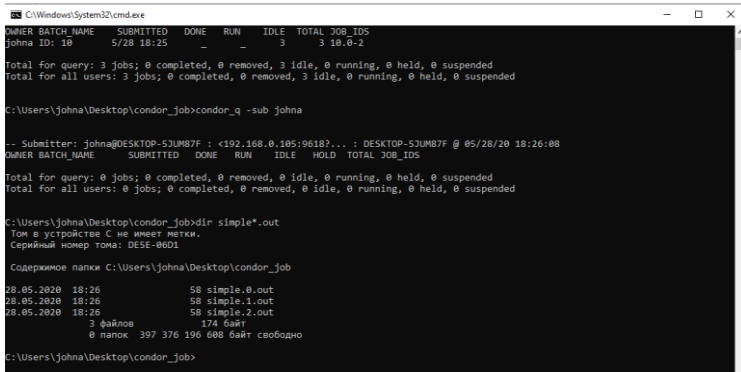
- результат роботи програми та час затримки в файлі simple.out;
- журнал подій в файлі simple.log;
- помилки при компіляції та роботі програми в файлі simple.error.

Имя	Дата изменения	Тип	Размер
simple.0.error	28.05.2020 18:26	Файл "ERROR"	0 КБ
simple.0.out	28.05.2020 18:47	Файл "OUT"	1 КБ
simple.1.error	28.05.2020 18:26	Файл "ERROR"	0 КБ
simple.1.out	28.05.2020 18:26	Файл "OUT"	1 КБ
simple.2.error	28.05.2020 18:26	Файл "ERROR"	0 КБ
simple.2.out	28.05.2020 18:26	Файл "OUT"	1 КБ
simple.c	28.05.2020 16:57	JetBrains CLion	1 КБ
simple.exe	28.05.2020 18:25	Приложение	56 КБ
simple.log	28.05.2020 18:47	Текстовый докум...	6 КБ
submit.sub	28.05.2020 17:35	Файл "SUB"	1 КБ

Рисунок 4.30 – Вміст директорії

4.2.7.4 Відображення результуючих файлів

```
dir simple*.out
```



```

C:\Windows\System32\cmd.exe
OWNER BATCH_NAME      SUBMITTED   DONE    RUN    IDLE  TOTAL JOB_IDS
johna ID: 10           5/28 18:25 -      -      3      3 10-0-2

Total for query: 3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended
Total for all users: 3 jobs; 0 completed, 0 removed, 3 idle, 0 running, 0 held, 0 suspended

C:\Users\johna\Desktop\condor_job>condor_job -sub johna

-- Submitter: johna@DESKTOP-SJUM87F : <192.168.0.105:9618>... : DESKTOP-SJUM87F @ 05/28/20 18:26:08
OWNER BATCH_NAME      SUBMITTED   DONE    RUN    IDLE  HOLD  TOTAL JOB_IDS

Total for query: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
Total for all users: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended

C:\Users\johna\Desktop\condor_job>dir simple*.out
Том в устро́йстве C не имеет метки.
Серийный номер тома: DE5E-B6D1

Содержимое папки C:\Users\johna\Desktop\condor_job

28.05.2020 18:26                58 simple.0.out
28.05.2020 18:26                58 simple.1.out
28.05.2020 18:26                58 simple.2.out
                3 файла              174 байт
0 папок 397 376 196 680 байт свободно

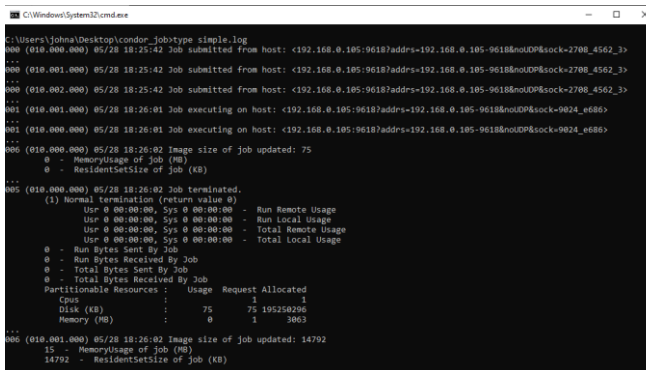
C:\Users\johna\Desktop\condor_job>

```

Рисунок 4.31 – Відображення результуючих файлів

4.2.7.5 Наступною командою відкриємо файл журналу події. Кожна подія у файлі відокремлена рядком. Для кожної події перше 3-розрядне значення - це номер події.

```
type simple.log
```



```

C:\Users\johna\Desktop\condor_job>type simple.log
(010.000.000) 05/28 18:25:42 Job submitted from host: <192.168.0.105:9618>addr=192.168.0.105-9618noudpsock=2708_4562_3>
...
(010.001.000) 05/28 18:25:42 Job submitted from host: <192.168.0.105:9618>addr=192.168.0.105-9618noudpsock=2708_4562_3>
...
(010.002.000) 05/28 18:25:42 Job submitted from host: <192.168.0.105:9618>addr=192.168.0.105-9618noudpsock=2708_4562_3>
...
(010.001.000) 05/28 18:26:01 Job executing on host: <192.168.0.105:9618>addr=192.168.0.105-9618noudpsock=9024_e086>
...
(010.000.000) 05/28 18:26:02 Job updated: 75
0 - MemoryUsage of Job (MB)
0 - ResidentSetSize of Job (KB)
(010.000.000) 05/28 18:26:02 Job terminated.
(1) Normal termination (return value 0)
  User 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
  User 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
  User 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
  User 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
0 - Run Bytes Sent By Job
0 - Run Bytes Received By Job
0 - Total Bytes Sent By Job
0 - Total Bytes Received By Job
Partitionable Resources :  Usage Request Allocated
  Cpu      :      1      1
  Disk (KB):      75  75 195250296
  Memory (MB):      0      1    3063
(010.000.000) 05/28 18:26:02 Image size of job updated: 14792
15 - MemoryUsage of Job (MB)
14792 - ResidentSetSize of Job (KB)

```

Рисунок 4.32 – Файл журналу подій

4.2.7.6 Тепер переглянемо усі результуючі файли.

```
type simple.0.out
type simple.1.out
type simple.2.out
```

```
C:\Windows\System32\cmd.exe
...
000 (010.002.000) 05/28 18:16:04 Image size of job updated: 14804
15 - MemoryUsage of Job (MB)
14804 - ResidentSetSize of Job (KB)
...
005 (010.002.000) 05/28 18:20:04 Job terminated.
(1) Normal termination (return value 0)
  User 0 00:00:00, Sys 0 00:00:00 - Run Remote Usage
  User 0 00:00:00, Sys 0 00:00:00 - Run Local Usage
  User 0 00:00:00, Sys 0 00:00:00 - Total Remote Usage
  User 0 00:00:00, Sys 0 00:00:00 - Total Local Usage
  0 - Run Bytes Sent By Job
  0 - Run Bytes Received By Job
  0 - Total Bytes Sent By Job
  0 - Total Bytes Received By Job
Partitionable Resources :   Usage   Request Allocated
  Cpus                   :         1           1
  Disk (KB)              :        75       75 195250296
  Memory (MB)            :         15          1    3063
...
C:\Users\Johna\Desktop\condor_jobtype simple.0.out
Thinking really hard for 4 seconds...
we calculated: 20
C:\Users\Johna\Desktop\condor_jobtype simple.1.out
Thinking really hard for 4 seconds...
we calculated: 22
C:\Users\Johna\Desktop\condor_jobtype simple.2.out
Thinking really hard for 4 seconds...
we calculated: 24
C:\Users\Johna\Desktop\condor_job>
```

Рисунок 4.33 – Результат

4.2.7.7 Після перегляду результуючих файлів можна зробити висновок, що програма працює коректно значення аргументів помножених на 2 підраховано вірно, та затримка перед виконанням наступного аргументу була 4 секунди.

4.2.7.8 В таблиці 4.1 наведено команди які потрібно водити на відповідних вузлах з пулу.

Таблиця 4.1 Команди для кожного з вузлів

Сервер	Виконуюча	Клієнт
<ul style="list-style-type: none"> condor_master.exe condor_collector.exe condor_negotiator 	<ul style="list-style-type: none"> condor_master.exe condor_startd.exe 	<ul style="list-style-type: none"> condor_master.exe condor_shedd.exe gcc -o simple simple.c condor_submit submit.sub condor_q -sub johna dir simple*.out type simple.log type simple.0.out type simple.1.out type simple.2.out

4.3 Зміст звіту

- 4.3.1 Титульний лист.
- 4.3.2 Мета роботи.
- 4.3.3 Послідовність дій роботи.
- 4.3.4 Відповіді на контрольні питання.

4.4 Контрольні питання

- 4.4.1 Що таке HTCondor?
- 4.4.2 Мета використання проміжного програмного забезпечення.
- 4.4.3 Які необхідно запустити команди для роботи HTCondor?
- 4.4.4 Які існують аналоги HTCondor?

ЛАБОРАТОРНА РОБОТА

Розгортання платформи *ownCloud*

Мета роботи – навчитися розгортати та конфігурувати платформу хмарних обчислень *ownCloud*.

5.1 Теоретичні відомості

Хмарні обчислення (*Cloud Computing*) – сучасна технологія реалізації розподілених обчислень, яка, на відміну від розглянутої раніше *Grid*-технології, призначена здебільшого для її корпоративного застосування.

Визначення хмарних обчислень дається організацією *National Institute of Standard and Technologies (NIST)*: хмарні обчислення – це модель реалізації зручного суцільного (*ubiquitous*) доступу за вимогою до розподіленого обсягу (*pool*) конфігурованих обчислювальних ресурсів (мережі, сервери, накопичувачі, додатки, сервіси), які можуть бути швидко надані у користування, і так само швидко звільнені, з мінімальними витратами на управління та втручаннями провайдера [11].

Одна з основних переваг від використання технології хмарних обчислень полягає у можливості одержання віддаленого доступу до даних та додатків, розміщених на різноманітних мережних пристроях. Такий доступ можна здійснювати, зокрема, з використанням веб-браузерів.

В роботі виконуватимемо розгортання власної *Cloud*-платформи у *Windows*-середовищі. Для цього скористаємось наступними засобами:

- набір сервісів *Microsoft IIS (Internet Information Services)*, що включає веб-сервер *IIS* з підтримкою протоколу *CGI (Common Gateway Interface)*, призначеного для взаємодії веб-сервера із зовнішніми програмними компонентами;

- гіпертекстовий препроцесор *PHP (Hypertext Preprocessor)*;

- система управління базами даних (СУБД) *MySQL*.

Набір програмних платформ і засобів, для яких було перевірено працездатність *ownCloud* та на основі яких рекомендується виконувати дану роботу, наступний:

- операційна система *Microsoft Windows 7*;

- пакет *ownCloud* – збірка *owncloud-8.0.3.zip* [12];

- препроцесор *PHP* – збірка *php-5.6.9-nts-Win32-VC11-x86.zip*;
- СУБД *MySQL*-серверу – збірка *mysql-5.5.30-win32.msi*.

Варто відзначити, що у випадку використання операційної системи *Microsoft Windows XP*, версії *PHP 5.5* та наступні вже не підтримуються.

Виконання роботи полягає в інсталяції та налаштування платформи *ownCloud* на прикладі збірки *owncloud-8.0.3.zip*.

Здійсненню підлягають наступні кроки:

- активація сервісів *Microsoft IIS*;
- інсталяція гіпертекстового препроцесора *PHP*;
- інсталяція та налаштування системи *MySQL*;
- розгортання платформи *ownCloud*;
- налаштування платформи *ownCloud*.

5.2 Порядок виконання роботи

5.2.1 Активація сервісів *Microsoft IIS*.

5.2.1.1 Виконати наступні дії: "Панель управління" > "Програми і компоненти" > "Увімкнення чи відключення компонентів Windows". Названа опція при цьому може не з'явитися. Шляхи подолання цієї перепони наступні:

– здійснити "Увімкнення чи відключення компонентів Windows" від імені адміністратора: "Пуск" > "Стандартні" > відкрити командний рядок від імені адміністратора > виконати з командного рядка утиліту *OptionalFeatures.exe*;

– (та/або) відредагувати системний реєстр, виконавши від імені адміністратора у командному рядку команду *regedit > HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Control > Windows* > встановити значення змінної *CSDVersion* в 0 та перезавантажити систему.

5.2.1.2 Встановити налаштування сервісів, відзначивши нижченаведені позиції прапорцями (рис. 5.1):

- встановити фільтрацію запитів у якості налаштування безпеки;
- встановити підтримку *CGI* як компоненту розробки додатків;
- задати загальні функції протоколу *HTTP* (*HyperText Transfer Protocol*) – документ за замовчанням, помилки *HTTP*, перегляд каталогу, статичний вміст;
- встановити опції перевірки працездатності та діагностування – ведення журналу *HTTP* та монітор запитів;

– у якості функції підвищення швидкодії встановити "Стиснення статичного вмісту";

– у якості засобу керування веб-сайтом задати "Консоль керування IIS".

Набуття сили внесених змін займе декілька хвилин.

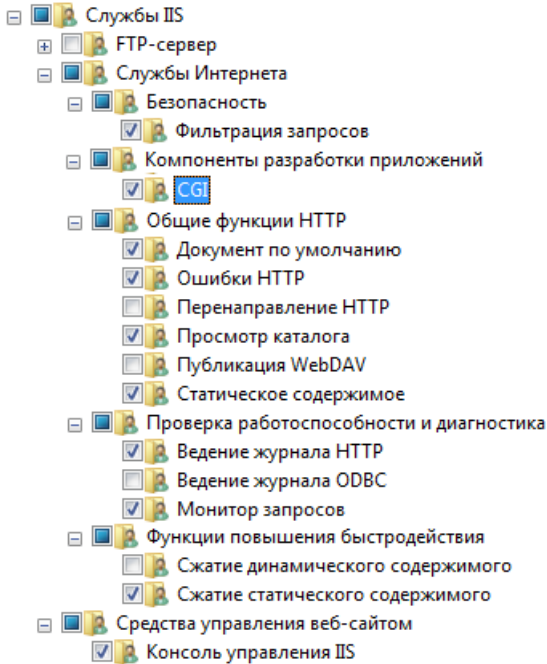


Рисунок 5.1 – Налаштування сервісів *Microsoft IIS*

Підтвердженням успішності активації сервісів *Microsoft IIS* має бути типова відповідна стартова сторінка у вікні веб-браузера, що доступна за адресою *http://localhost* (рис. 5.2).



Рисунок 5.2 – Стартова сторінка веб-сервера *Microsoft IIS*

5.2.2 Інсталяція і налаштування гіпертекстового препроцесора *PHP*.

5.2.2.1 Розпакувати вміст архіву *php-5.6.9-nts-Win32-VC11-x86.zip* у кореневому каталозі логічного диску *C:*; перейменувати одержаний однойменний каталог у *php*.

5.2.2.2 Модифікувати системну змінну оточення *Path*, додавши до неї шлях *C:\php*:

```
setx path "%path%;c:\php"
```

5.2.2.3 Здійснити перехід до "Пуск" > "Панель керування" > "Адміністрування" > "Диспетчер служб IIS".

5.2.2.4 Обрати елемент "Співставлення обробників".

5.2.2.5 Заповнити поля форми згідно з рис. 5.3. Одержуваний результат має бути таким, який подано на рис. 5.4.

 The image is a screenshot of the 'Handler Mappings' configuration window in the Microsoft Management Console (MMC) for Internet Information Services (IIS). The window has a light grey background and contains several labeled fields:

- Путь запроса:** A text box containing the value `*.php`. Below it, an example is shown: `Пример: *.bas, wsvc.axd`.
- Модуль:** A dropdown menu currently showing `FastCgiModule`.
- Исполняемый файл (необязательно):** A text box containing the path `C:\php\php-cgi.exe`. To the right of this box is a small button with three dots (ellipsis).

Рисунок 5.3 – Налаштування *PHP* під *CGI*

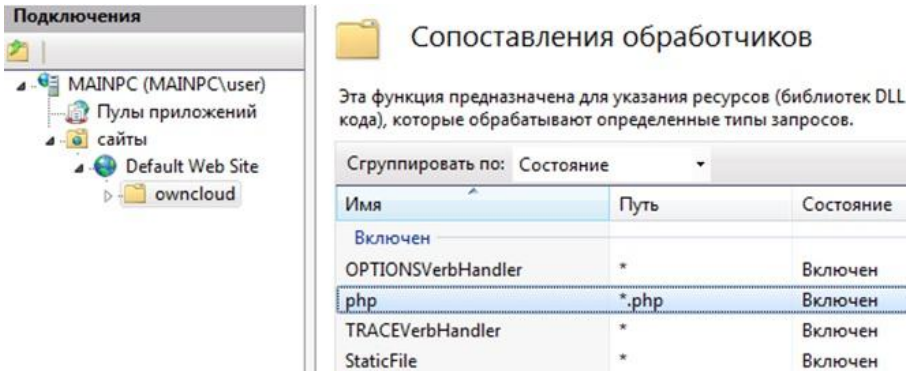


Рисунок 5.4 – Співставлення обробників

5.2.2.6 Встановити сторінку *index.php* у якості стартової (за замовчанням): на панелі інструментів *Диспетчера служб IIS* обрати "Документ за замовчанням", для якого задати назву *index.php*.

5.2.2.7 Перевірити зроблені налаштування гіпертекстового препроцесора. Для цього створити файл *test.php* у каталозі *C:\inetpub\wwwroot\owncloud*. Файл повинен мати наступний вміст:

```
<?php phpinfo(); ?>
```

5.2.2.8 У веб-браузері виконати наступний запит:

```
http://localhost/owncloud/test.php
```

Має з'явитися сторінка, наведена на рис. 5.5.

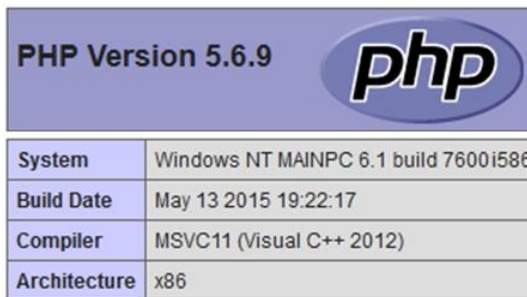


Рисунок 5.5 – Перевірка налаштувань PHP

Зауваження: при налаштуванні *PHP* можуть виникнути помилки. Шляхи їх вирішення можуть бути наступними:

- перевірити вміст конфігураційного файлу *web.config*, розміщеного у каталозі *C:\inetpub\wwwroot*;

- внести зміни до конфігураційного файлу *applicationHost.config*, розміщеного у каталозі *C:\Windows\System32\inetsrv\config*:

а) у випадку виникнення помилки HTTP Error 500.19 – Internal Server Error, необхідно замінити тег

```
<section name="handlers" overrideModeDefault="Deny" />
```

тегом

```
<section name="handlers" overrideModeDefault="Allow" />
```

б) у випадку виникнення помилки 0x800700b7, позбутися дублювання тегу `<add accessType="Allow" users="*" />`, та/або закоментувати тег

```
<add name="php" path="*.php" verb="*" modules="FastCgiModule" scriptProcessor="C:\php\php-cgi.exe" resourceType="Unspecified" />
```

, розміщений між тегами `<handlers accessPolicy="Read, Script">` та `</handlers>`, якщо у файлі *applicationHost.config* вже існує конструкція, наведена в лістингу 5.1.

Лістинг 5.1 – Вміст файлу *applicationHost.config*

```
<location path="Default Web Site/owncloud">
    <system.webServer>
        <handlers>
            <add name="php" path="*.php" verb="*"
modules="FastCgiModule" scriptProcessor="C:\php\php-
cgi.exe" resourceType="Unspecified" />
        </handlers>
    </system.webServer>
</location>
```

– модифікувати файл *php.ini* – скопіювати файл *php.ini-development* з каталогу *C:\php* до каталогу *C:\Windows*, перейменувавши цей файл у *php.ini*; встановити значення змінних перейменованого файлу згідно лістингу 5.2.

Лістинг 5.2 – Вміст файлу *php.ini*

```
log_errors=On
cgi.force_redirect = 0
cgi.fix_pathinfo = 1
fastcgi.impersonate = 1
fastcgi.logging = 0
```

При цьому змінна `open_basedir` має бути закоментована.

5.2.3 Інсталяція і налаштування MySQL.

5.2.3.1 Встановити MySQL-сервер – збірку *mysql-5.5.30-win32.msi*.

5.2.3.2 Модифікувати вміст файлу *php.ini*, розміщеного у каталозі *C:\Windows*:

- задати значення змінній `extension_dir`: `extension_dir = "C:\php\ext"`;
- розкоментувати рядки, наведені в лістингу 5.3.

Лістинг 5.3 – Модифікація вмісту файлу *php.ini*

```
extension=php_curl.dll
extension=php_gd2.dll
extension=php_mysql.dll
extension=php_mysqli.dll
extension=php_pdo_mysql.dll
```

5.2.4 Розгортання платформи *ownCloud*.

5.2.4.1 Розпакувати вміст архіву *owncloud-8.0.3.zip* до однойменного каталогу.

5.2.4.2 Перейменувати одержаний каталог в *owncloud*.

5.2.4.3 Помістити каталог *owncloud* до каталогу *C:\inetpub\wwwroot*.

5.2.4.4 Змінити права доступу до каталогу *owncloud*: "Властивості" > "Безпека" > "Користувачі" > "Змінити" > дозволити запис для групи "Користувачі".

5.2.4.5 Перевірити успішність інсталяції *ownCloud*, виконавши у адресному рядку браузера наступний запит:

`http://localhost/owncloud/`

У випадку успіху має з'явитися сторінка, подібна до такої, наведеної на рис. 5.6.

Рисунок 5.6 – Підтвердження успішності інсталяції *ownCloud*

Зауваження: у випадку виникнення помилок можливі наступні шляхи їх вирішення:

- створити пустий каталог *data* у каталозі *C:\inetpub\wwwroot\owncloud*;
- аби позбутися попереджень (*warning*-повідомлень) про непрацездатність *.htaccess*, розміщеного у каталозі *C:\inetpub\wwwroot\owncloud\config*, необхідно модифікувати вміст файлу *config.php*, розміщеного в цьому ж каталозі, додавши елемент '*datadirectory*', що міститиме шлях до каталогу даних *data*, подібно до того як наведено у лістингу 5.4;

Лістинг 5.4 – Доповнений вміст файлу *config.php*

```
<?php
$CONFIG = array (
    'instanceid' => 'oc8c64jmy7ci',
    'datadirectory' => 'D:/VADIM/emc/data/',
);
```

– створити відповідний каталог *data* за вказаним у файлі шляхом;

– перезапустити веб-сервер *IIS*, виконавши в командному рядку наступну команду:

```
iisreset /restart
```

Знову спробувати виконати завдання 5.2.4.5.

5.2.5 Налаштування платформи *ownCloud*.

5.2.5.1 Заповнити поля форми, наведеної на рис. 5.6.

Зауваження: логін та пароль облікового запису адміністратора *ownCloud* не мають співпадати із логном та паролем СУБД *MySQL*. У першому випадку введемо пару значень (*mag, mag*), у другому – (*root, root*).

5.2.5.2 У поле "*Database name*" (рис. 5.6) введемо значення *owncloud_db*.

5.2.5.3 Зафіксувати введені дані, натиснувши на елемент управління "*Finish setup*", у результаті чого має з'явитися сторінка-привітання, наведена на рис. 5.7.

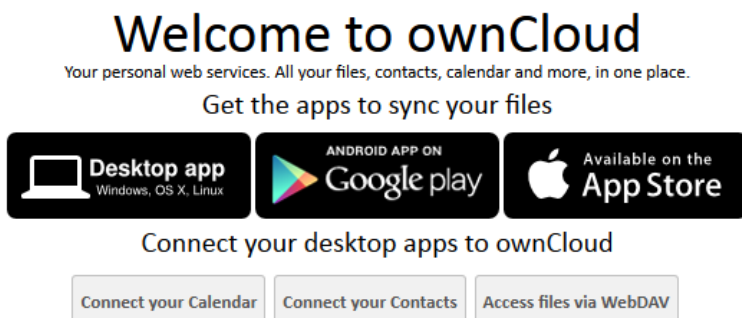


Рисунок 5.7 – Початкова сторінка середовища *ownCloud*

5.2.5.4 Одержати доступ до файлів, розміщених у розгорнутому хмарному середовищі, ввівши в адресному рядку браузера шлях

`http://localhost/owncloud/index.php/apps/files/`

де знаходиться, зокрема, файл з інформацією для користувача – *ownCloudUserManual.pdf*, який переглянути безпосередньо у браузері.

5.3 Зміст звіту

5.3.1 Титульний лист.

5.3.2 Мета роботи.

5.3.3 Знімок екрану, що підтверджує успішність розгортання платформи *ownCloud*.

5.3.4 Відповіді на контрольні питання.

5.4 Контрольні питання

5.4.1 Призначення технології кросплатформних обчислень.

5.4.2 Компоненти (та їх призначення), необхідні для розгортання системи *ownCloud*.

5.4.4 Етапи розгортання та налаштування системи *ownCloud*.

ЛІТЕРАТУРА

1. Кирьянов А. Введение в технологию Грид / А. Кирьянов, Ю. Рябов. – Гатчина, 2006. – 39с.
2. Шпаковский Г. Применение технологии MPI в Грид / Г.И. Шпаковский, В.И. Стецюренко, А.Е. Верхотуров, Н.В. Серикова. – Минск, 2008. – 137с.
3. GridSim: A Grid Simulation Toolkit For Resource Modeling And Application Scheduling For Parallel And Distributed Computing [Електронний ресурс] // University of Melbourne. – Режим доступу: <http://www.cloudbus.org/gridsim/>.
4. Дорошенко А.Ю. Паралельна розподілена реалізація моделювання паралельних обчислень / А.Ю. Дорошенко, М.В. Гнинюк // Проблеми програмування. – 2014. – № 1. – С. 40 – 48.
5. SimJava [Електронний ресурс] // Institute for Computing Systems Architecture (ICSA), School of Informatics, University of Edinburgh. – Режим доступу: <http://www.icsa.inf.ed.ac.uk/research/groups/hase/simjava/>.
6. JPPF documentation [Електронний ресурс] – Режим доступу: <http://jppf.org/doc/v5/>.
7. Botta A. Integration of Cloud computing and Internet of Things: A survey / A. Botta, W. de Donato, V. Persico, A. Pescapé // Future Generation Computer Systems. – 2016. – Vol. 56, Issue C. – P. 684–700. doi: 10.1016/j.future.2015.09.021
8. OwnCloud Documentation Overview [Електронний ресурс] – Режим доступу: <https://doc.owncloud.org>.