

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”
Кафедра автоматизації проектування енергетичних процесів і систем

Організація оптимального обчислювального процесу в корпоративній мережі

Виконав:
Симоненко Б.О.

Керівник:
Медведева В.М.

Мета дослідження

Дослідити можливість використання комп'ютерів працівників компанії як вузлів кластеру для виконання розподілених unit-тестів

Для реалізації мети сформульовані такі задачі:

- проаналізувати існуючі рішення;
- вдосконалити алгоритм вибору вузлів для мінімізації перетину з роботою працівників;
- здійснити програмну реалізацію програмного комплексу.

Об'єктом дослідження є програмне забезпечення розподілених систем

Предметом дослідження є програмне забезпечення вибору вузлів для виконання розподілених задач

Наукова новизна

- удосконалено алгоритм розподілення обчислювальних задач з метою мінімізації перетину виконання задач на вузлі із звичайною роботою працівника, за рахунок збору і використання даних про активність працівника;
- набуло подальшого розвитку застосування програмного інструментарію для розподіленого виконання unit-тестів

Аналоги

Для проектів на основі .NET є такі існуючі рішення:

- P NUnit – розширення фреймворку для unit-тестування NUnit
- Akka.NET – набір програмного інструментарію для створення розподіленого програмного забезпечення

Недоліки аналогів

- Не враховують зайнятість вузлів
- Не враховують технічних характеристик вузлів
- Не вміють аналізувати програмний код задач для відстеження змін та прогнозування часу їх виконання
- Відсутність підтримки розробниками (PNUit)

Пошук рішення

Для мінімізації перетину виконання задач на вузлах і роботи працівників необхідно:

1. Знати коли вузол звільнюється і на скільки часу, швидкодію вузла
2. Вміти прогнозувати тривалість виконання задачі, щоб її виконання завершилось до того як повернеться співробітник

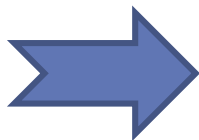
Модуль телеметрії на вузлі

- Повіdomляє коли користувач працює і коли відсутній
- При звільненні комп'ютеру користувач повідомляє тривалість відсутності шляхом запуску однієї із 3 програм на панелі швидкого доступу (30хв, 1год, 2год)
- При звільненні комп'ютеру виконує тест Люка — Лемера для оцінки швидкодії ЦП (CPU)

Прогнозування тривалості виконання

```
[Method]
public string LongRunning4()
{
    double pow = 1;
    for (int i = 0; i < 50; i++)
    {
        pow = Math.Pow(i, 2);
        pow /= 2;
    }

    return pow.ToString();
}
```



```
0000 : nop
0001 : ldc.r81072693248
0010 : stloc.0
0011 : ldc.i4.0
0012 : stloc.1
0013 : br.s 0050
0015 : nop
0016 : ldloc.1
0017 : conv.r8
0018 : ldc.r81073741824
0027 : call System.Double System.Math::F
0032 : stloc.0 0033 : ldloc.0
...
0070 : ret
```



Hash: 567AA56BBD04A313A1C215FD24913848

Прогнозування тривалості виконання

$$t_{\text{оч}} = \frac{\sum_{i=1}^n P_i * t_i}{P_B * n} \quad \text{– прогнозована тривалість виконання unit-тесту}$$

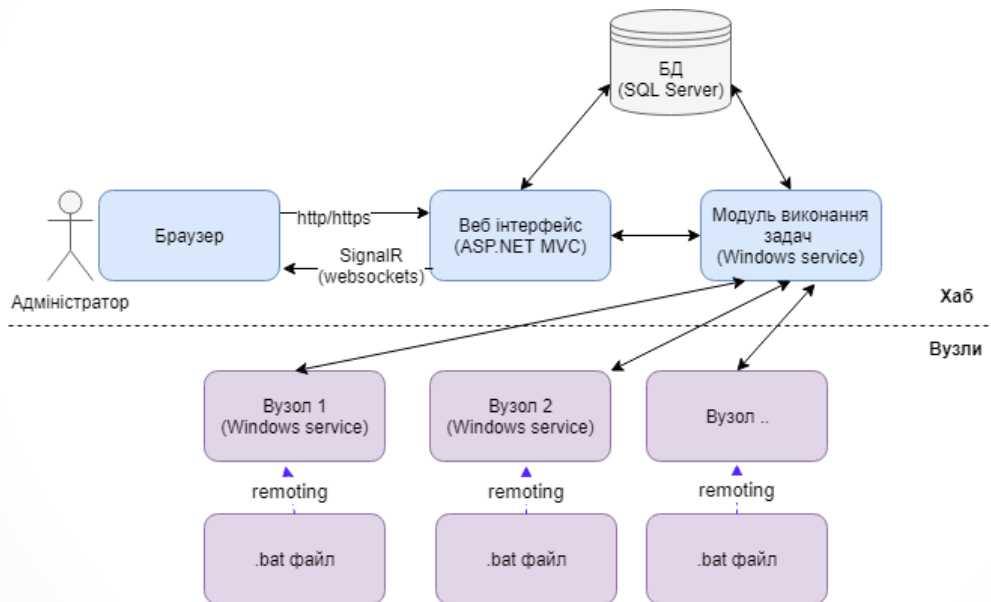
n – кількість виконань unit-тесту

P_i – оцінка швидкодії вузла під час виконання i -ого тесту на ньому

t_i – час виконання i -ого тесту

P_B – оцінка швидкодії вузла на якому передбачається поточне виконання

Архітектура програмного комплексу



Інтерфейс створення задачі

Tasks

Description

Required nodes count

1

.dll file

Choose File

No file chosen

Add

Clear

Description

status

Required Nodes Count

Operations

task with 5

0

1

5



defence

0

1

5



t1

0

1

0



Інтерфейс деталей задачі

 localhost:1966/tasks 

[Distributed](#) [Home](#) [Machines](#) [Tasks](#) [Task Executions](#)

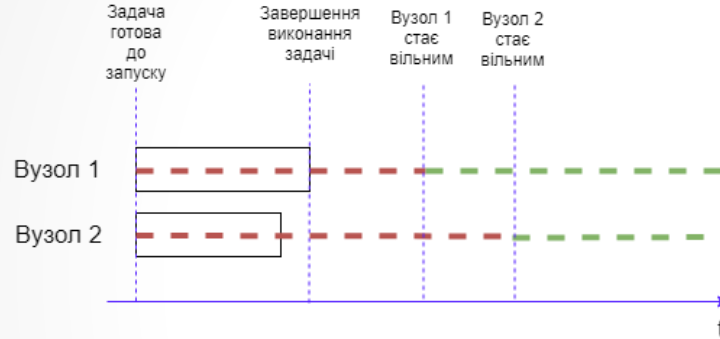
Tasks

Assembly	Type	Method	Method Instructions	Hash
PredefinedTestMethods, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null	PredefinedTestMethods.Class1	Class1Task1	0000 : nop 0001 : ldstr "ResClass1Task1" 0006 : stloc.0 0007 : br.s 0009 0009 : ldloc.0 0010 : ret	809778BF94DE001E0F0463768030DB6A
PredefinedTestMethods, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null	PredefinedTestMethods.Class1	Class1Task2	0000 : nop 0001 : ldc.i4.2 0002 : stloc.0 0003 : call System.Void System.Console::WriteLine() 0008 : nop 0009 : ldstr "ResClass1Task2" 0014 : stloc.1 0015 : br.s 0017 0017 : ldloc.1 0018 : ret	96E25F950AFF559D3E6317A01AB3C7EB
PredefinedTestMethods, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null	PredefinedTestMethods.Class1	Class1BoolTask	0000 : nop 0001 : ldc.i4.0 0002 : stloc.0 0003 : br.s 0005 0005 : ldloc.0 0006 : ret	8AD243325798676514BD49C5F8E787BA

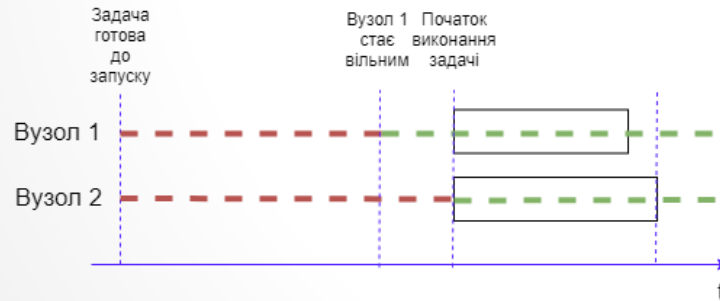
Інтерфейс результату виконання задачі

localhost:1966/taskExecutions				
Distributed Home Machines Tasks Task Executions				
Task executions				
Finished	Status	Result	Result message	Node address
17.04.2018 10:24:00	1	0	ResClass1Task1	akka.tcp://iot-client@localhost:56001/user/iot-client
17.04.2018 10:24:00	1	0	ResClass1Task2	akka.tcp://iot-client@localhost:56001/user/iot-client
17.04.2018 10:24:00	1	0	False	akka.tcp://iot-client@localhost:56001/user/iot-client
17.04.2018 10:24:00	1	0	18678.23	akka.tcp://iot-client@localhost:56001/user/iot-client
17.04.2018 10:24:00	1	0	True	akka.tcp://iot-client@localhost:56001/user/iot-client
17.04.2018 10:24:00	1	0	18678.23	akka.tcp://iot-client@localhost:56001/user/iot-client

Порівняння з аналогами



Запуск задач PUnit



Запуск задач реалізованим рішенням

Висновки

- Проведено аналіз існуючих рішень для запуску розподілених unit-тестів
- Проведено аналіз можливості запуску unit-тестів на комп'ютерах працівників компанії
- Вдосконалено алгоритм вибору вузлів
- Проведено порівняльний експеримент з доведенням ефективності запропонованого рішення

Дякую за увагу