컴퓨터통신 및 실습
CPU Performance 시뮬레이션

2011270314

컴퓨터정보학과

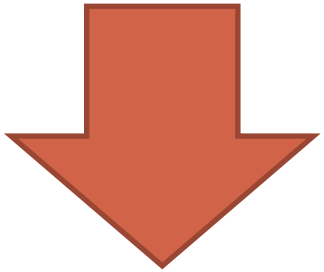
서인석

목차

개요

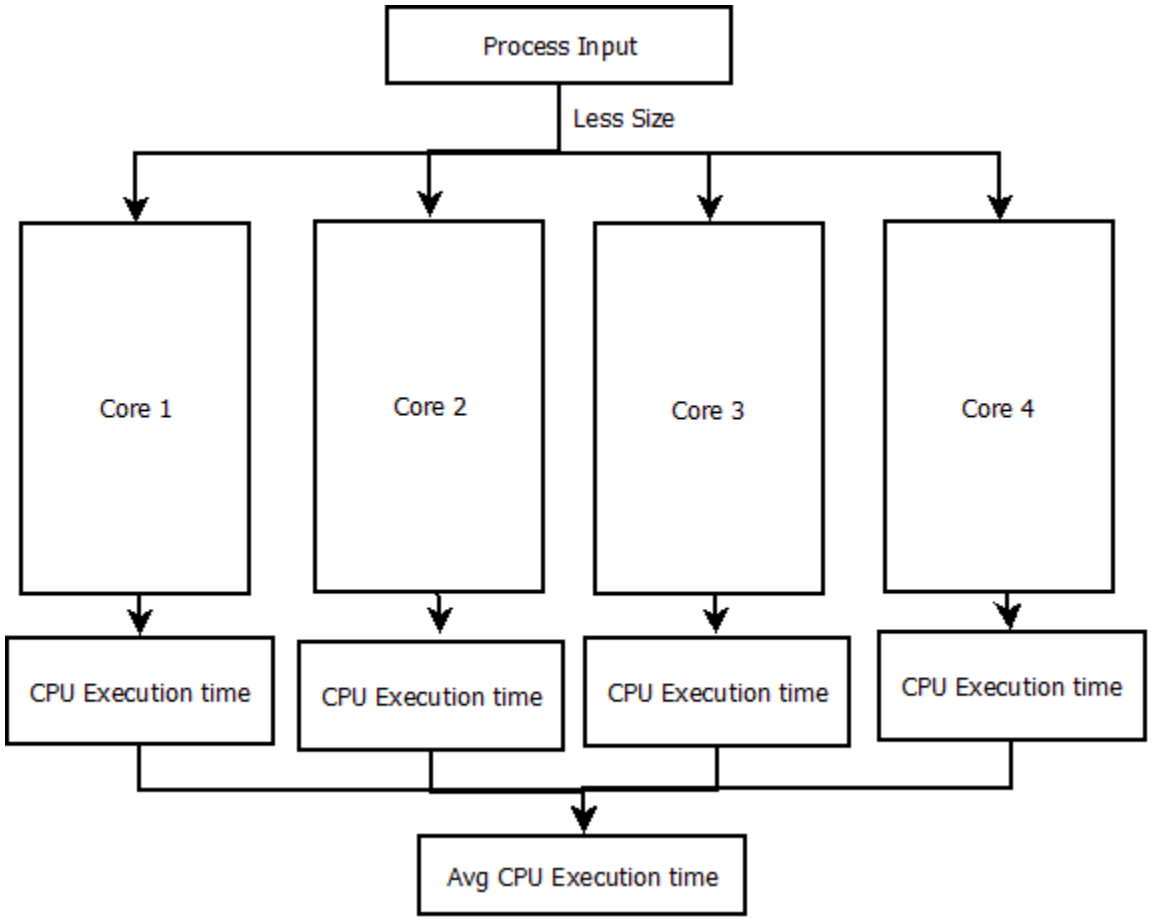
- 목적

CPU의 성능은 곧 컴퓨터의 성능 이라고 해도 과언이 아닐 만큼 컴퓨터 자체의 성능을 판단하는데 큰 몫을 차지하고 있다.

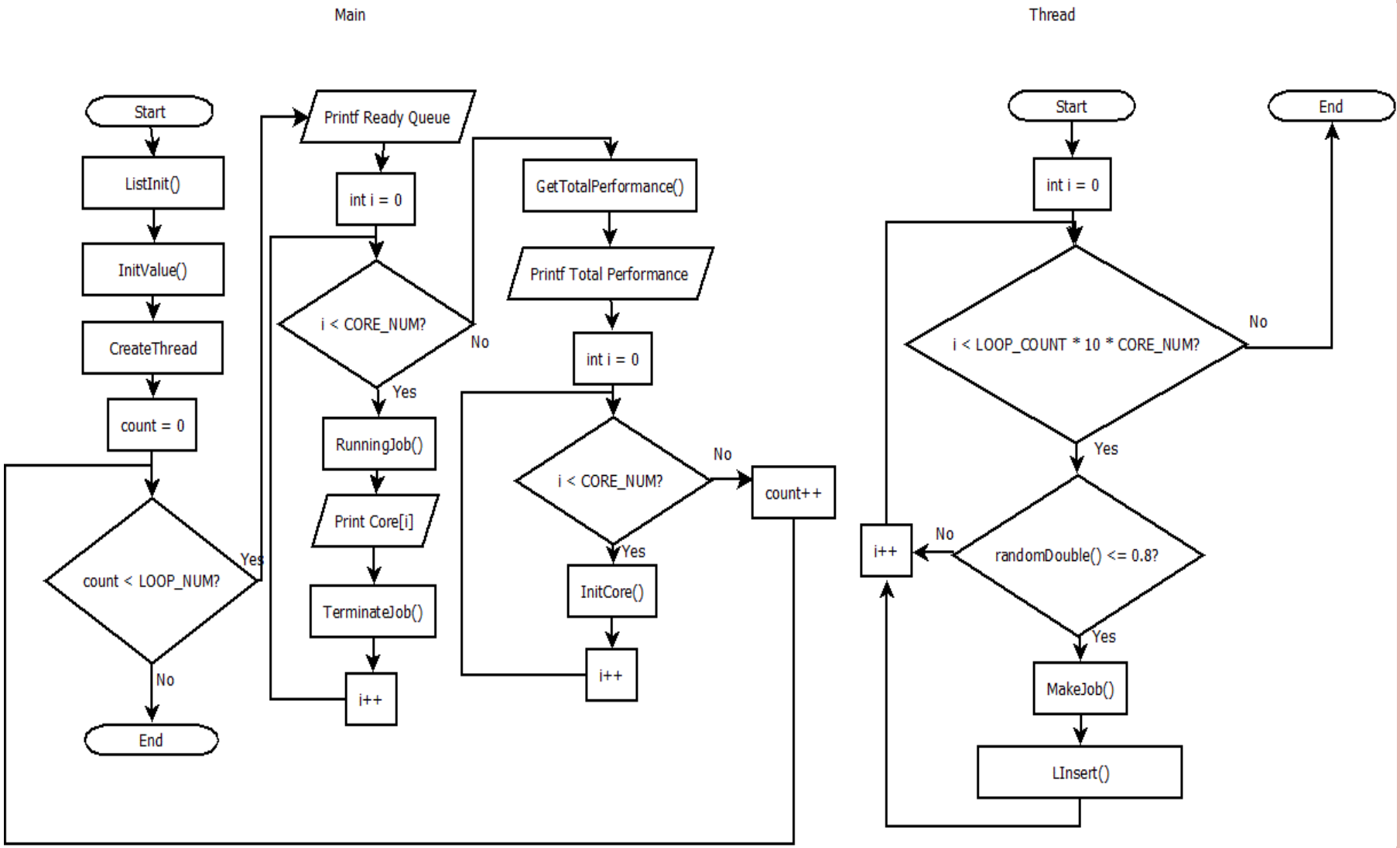


CPU의 성능을 시뮬레이션 하여, 성능을 높이기 위한 방법을 모색 및 기존의 CPU 성능 향상 방법들을 수치적으로 접근하여 확인한다.

시스템 소개


Process Input
Less Size
Core 1
Core 2
Core 3
Core 4
CPU Execution time
CPU Execution time
CPU Execution time
CPU Execution time
Avg CPU Execution time

시스템 소개


Main
Start
ListInit()
InitValue()
CreateThread
count = 0
count < LOOP_NUM?
Yes
No
End
Printf Ready Queue
int i = 0
i < CORE_NUM?
No
Yes
RunningJob()
Print Core[i]
TerminateJob()
i++
GetTotalPerformance()
Printf Total Performance
int i = 0
i < CORE_NUM?
No
Yes
InitCore()
i++
count++
Thread
Start
int i = 0
i < LOOP_COUNT * 10 * CORE_NUM?
No
Yes
End
randomDouble() <= 0.8?
No
Yes
i++
MakeJob()
LInsert()

시스템 소개

- 실험 과정

1) Ready Queue에 job이 도착 순서대로 쌓임
2) Ready Queue내의 job을 조회하여 가장 먼저 도착한 job을 선택
3) Job을 선택하여 코어에 할당 할 때, 코어가 비어있는 순서로 할당
4) 해당 Job의 정보를 확인하여 Execution time을 계산하고, CPU performance를 계산

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

5) Amdahl's law사용하여 코어 전체의 CPU performance를 계산

$$\frac{1}{(1-P)+\frac{P}{S}}$$

P = 성능 향상이 가능한 부분
S = 코어의 수

시스템 소개

- 실험 가정

1) CPU Execution time을 구하기 위한 값인 Instructions, Shared Instructions, 평균 CPI은 uniform Random variable을 사용하고 Cycle은 2GHz와 3Ghz로 각 실험마다 다르게 하여 실험한다.

2) Job 은 uniform Random variable을 사용 하여 80%의 확률로 생성되어, Ready queue에 저장된다.

3) 프로세스의 스케줄링은 FCFS(First come First Served)를 사용한다.

4) CPU는 멀티코어임을 가정한다.

시뮬레이션(코어가 2개인 CPU)

```
##################################################################################################

READY QUEUE
--------------------------------------------------------------------------------------------------
NUM                 |                 IN                 |                 CPU_ExecutionTime
--------------------------------------------------------------------------------------------------
65                                    55.221183                           1.957320
66                                    56.024119                           1.006684
67                                    58.581377                           0.159311
68                                    59.318349                           0.151305
69                                    60.842058                           0.171928
--------------------------------------------------------------------------------------------------
Ready Queue Size : 5


CORE[0]
--------------------------------------------------------------------------------------------------
NUM                 |                 IN                 |                 CPU_ExecutionTime
--------------------------------------------------------------------------------------------------
65                                    55.221183                           1.957320
--------------------------------------------------------------------------------------------------
Core[0] Performance : 0.510903


CORE[1]
--------------------------------------------------------------------------------------------------
NUM                 |                 IN                 |                 CPU_ExecutionTime
--------------------------------------------------------------------------------------------------
66                                    56.024119                           1.006684
--------------------------------------------------------------------------------------------------
Core[1] Performance : 0.993361

TotalPerformance : 0.415254
```

시뮬레이션(코어가 4개인 CPU)

```
READY QUEUE
------------------------------------------------------------------------------------------
NUM                |              IN                |               CPU_ExecutionTime
------------------------------------------------------------------------------------------
7                                 7.332685                          0.418445
8                                 9.188367                          1.098070
9                                 9.541165                          0.388749
10                                9.599022                          0.881201
------------------------------------------------------------------------------------------
Ready Queue Size : 4


CORE[0]
------------------------------------------------------------------------------------------
NUM                |              IN                |               CPU_ExecutionTime
------------------------------------------------------------------------------------------
7                                 7.332685                          0.418445
------------------------------------------------------------------------------------------
Core[0] Performance : 2.389798

CORE[1]
------------------------------------------------------------------------------------------
NUM                |              IN                |               CPU_ExecutionTime
------------------------------------------------------------------------------------------
8                                 9.188367                          1.098070
------------------------------------------------------------------------------------------
Core[1] Performance : 0.910689

CORE[2]
------------------------------------------------------------------------------------------
NUM                |              IN                |               CPU_ExecutionTime
------------------------------------------------------------------------------------------
9                                 9.541165                          0.388749
------------------------------------------------------------------------------------------
Core[2] Performance : 2.572352

CORE[3]
------------------------------------------------------------------------------------------
```

시뮬레이션 결과

- 5000개의 데이터를 추출하여 텍스트로 저장

Core[1]_data - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

0.829162	1.814472	1.814472
3.092165	12.743359	12.743359
3.812756	3.299998	3.299998
3.994156	0.516477	0.516477
4.468778	9.067438	9.067438
5.556585	2.767705	2.767705

Core[2]_data - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

33.584106	1.130150	34.184880	0.671207	0.534711
35.461094	0.885690	37.815893	7.719509	1.222507
38.248676	1.159730	39.423486	0.776749	0.496382
40.724275	4.688300	41.089308	3.933425	2.753437
41.257586	1.640976	41.852257	9.419865	1.310487
42.242398	0.760542	42.599156	1.244675	0.525609
43.539836	1.144001	43.650545	1.050662	0.577482
44.737627	1.432387	45.449472	1.245491	0.968157
45.455380	1.374293	46.831079	5.701187	1.258133

Core[4]_data - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

230.730805	13.410686	231.661760	1.762856	232.025431
237.200576	4.353406	237.860733	0.756078	239.969879
240.196344	1.324615	240.307565	1.145908	240.527798
244.087337	0.718598	244.251255	6.157764	244.655881
250.702185	1.626929	250.954066	2.560522	252.796755
252.981168	1.577688	253.352684	1.920883	253.541982
258.743677	3.466307	260.328755	74.580630	261.558082
265.578121	8.641656	265.785955	2.116977	266.478096
267.889610	0.707911	268.001035	0.862962	271.140504
273.080100	0.543405	273.135761	0.796422	273.528158
276.177751	1.156586	276.397832	1.220754	276.483793
277.160021	0.500075	277.796658	31.279653	277.866355
278.639375	1.402907	279.100114	1.386752	279.981822

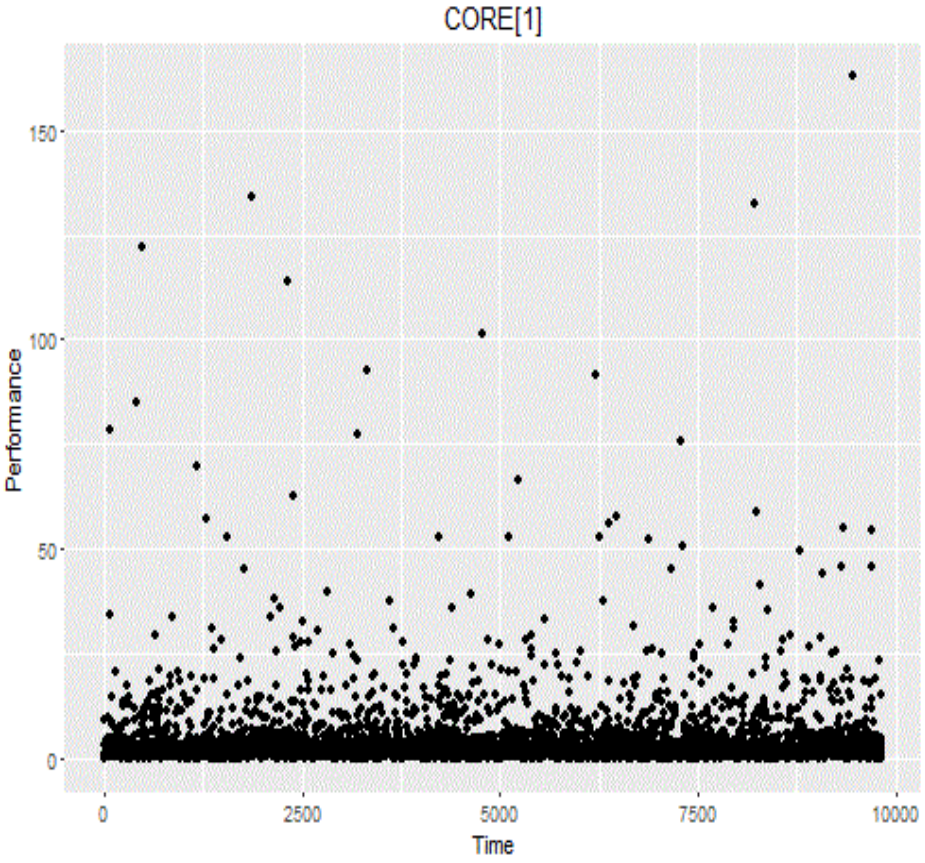
Core[8]_data - 메모장

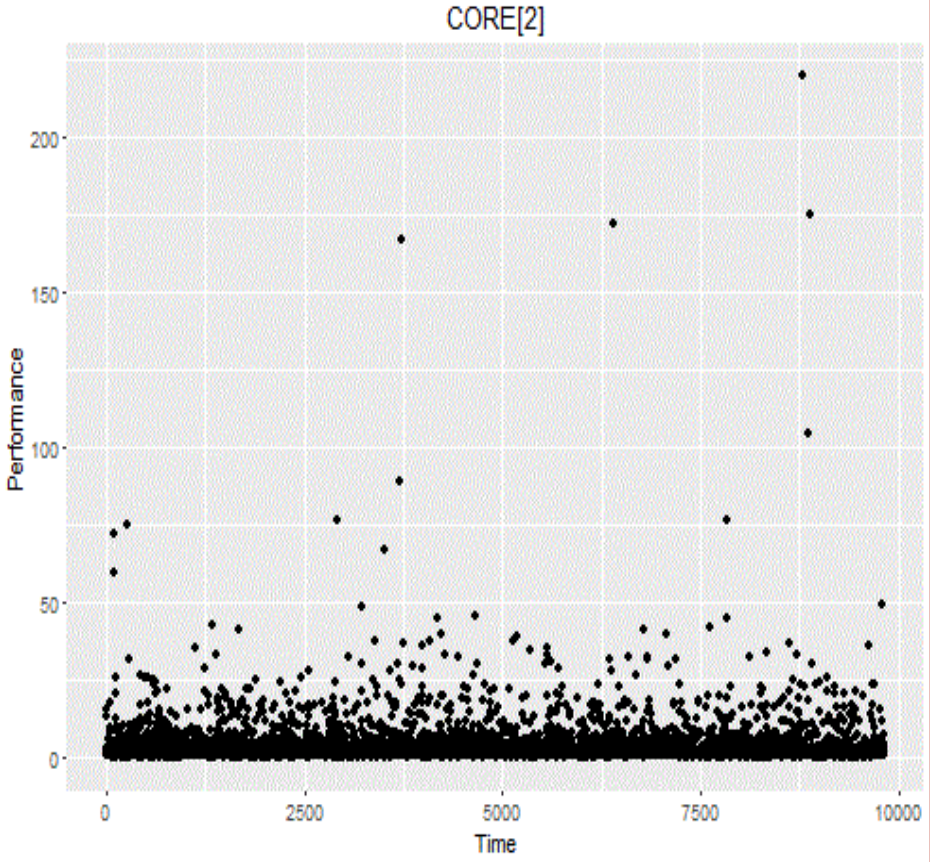
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

1423.839222	1.538212	1425.711833	1.593803	1426.707141
1433.649953	0.000000	1435.374644	0.596588	1435.533505
1442.237974	2.273907	1442.815371	0.536639	1442.840934
1448.187930	9.332137	1448.339804	1.422559	1449.865757
1455.711196	2.056782	1458.935949	0.935213	1459.284932
1465.371744	3.175835	1465.663963	2.487824	1466.965314
1470.817736	6.509138	1471.052954	1.787743	1471.648953
1477.804921	3.780401	1477.848746	1.522984	1479.126711
1485.019639	1.055143	1485.329509	0.958066	1489.593288
1494.562878	2.147515	1495.030656	0.668301	1496.579611
1505.830627	1.345347	1505.897025	1.066315	1507.558355

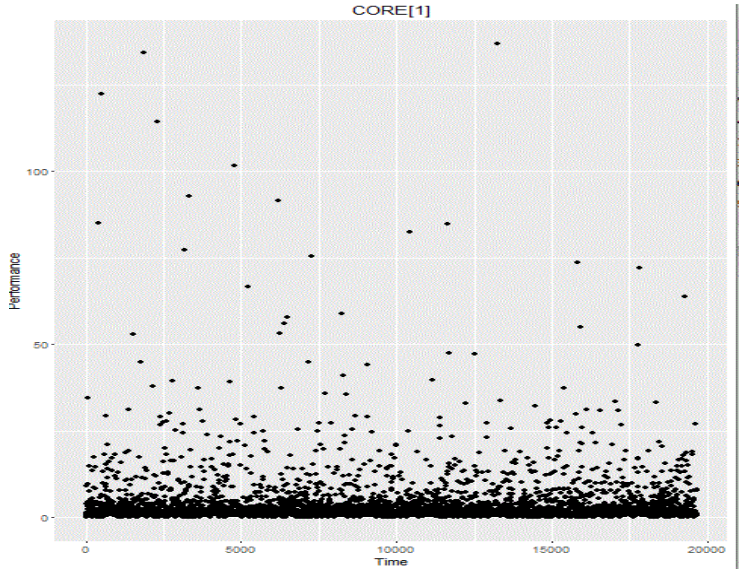
시뮬레이션 결과
(코어가 2개인 CPU)

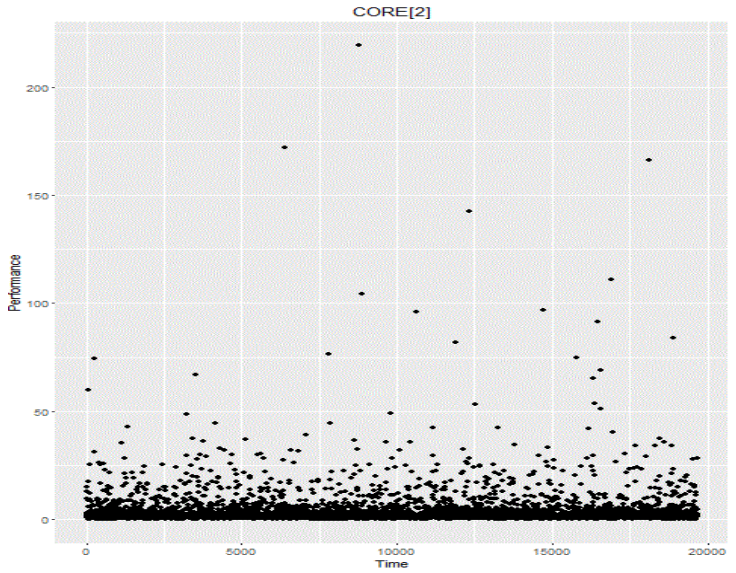
- 뽑은 데이터를 전처리 후 프로그램 “R”로 시각화

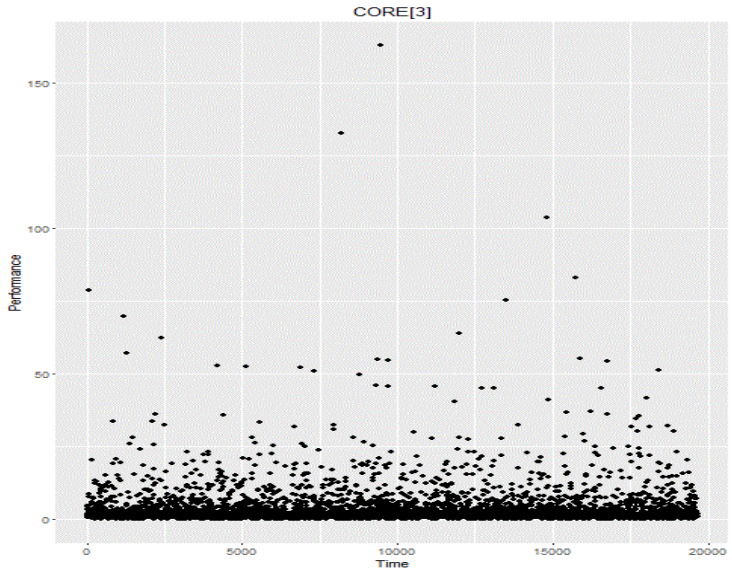


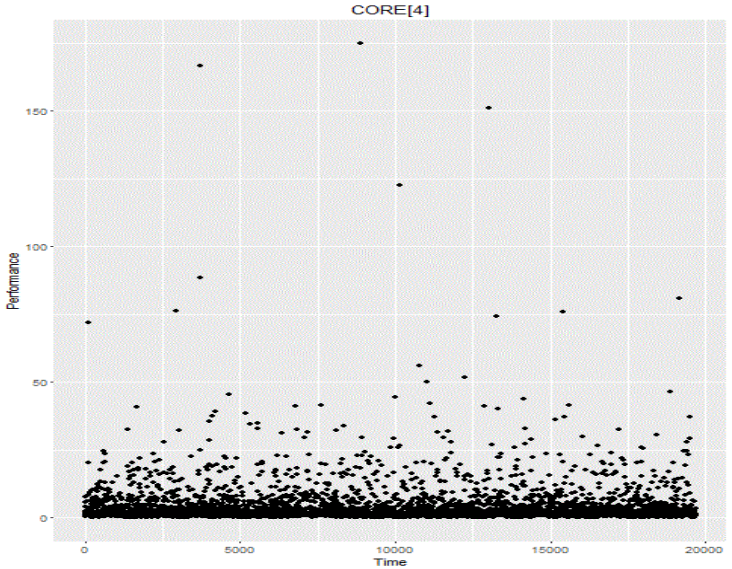


시뮬레이션 결과
(코어가 4개인 CPU)



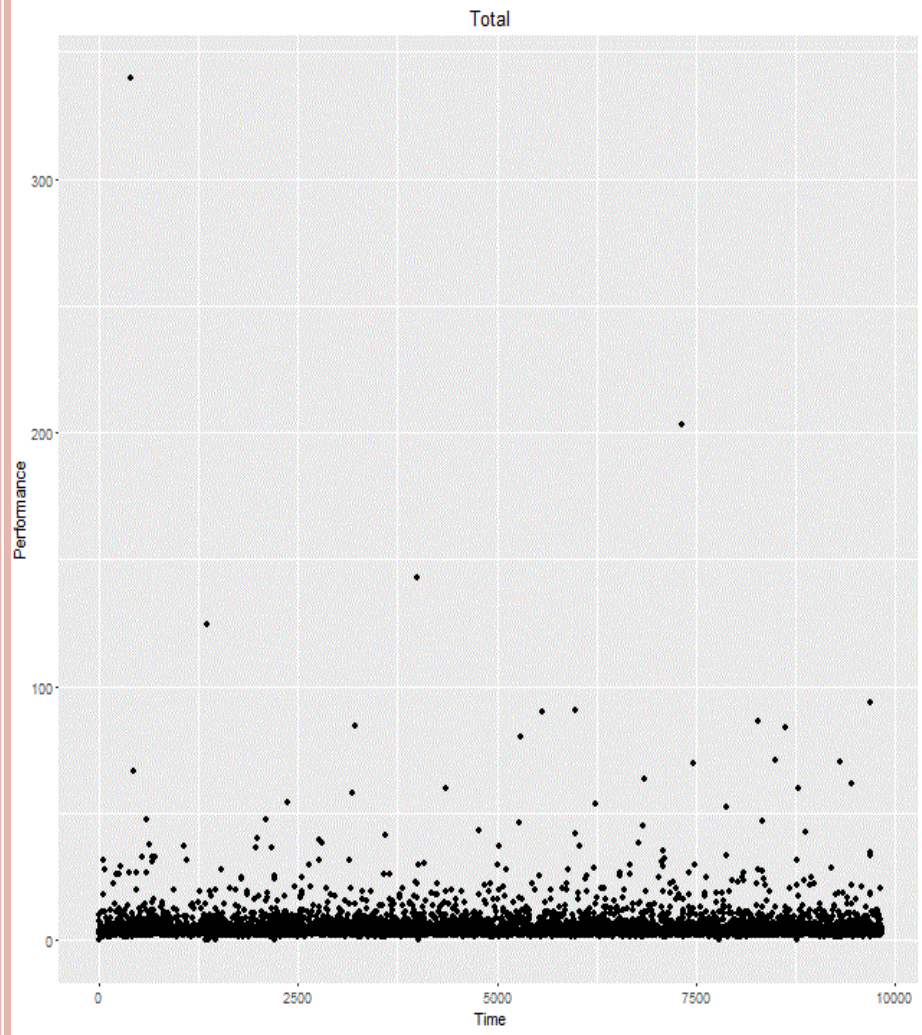




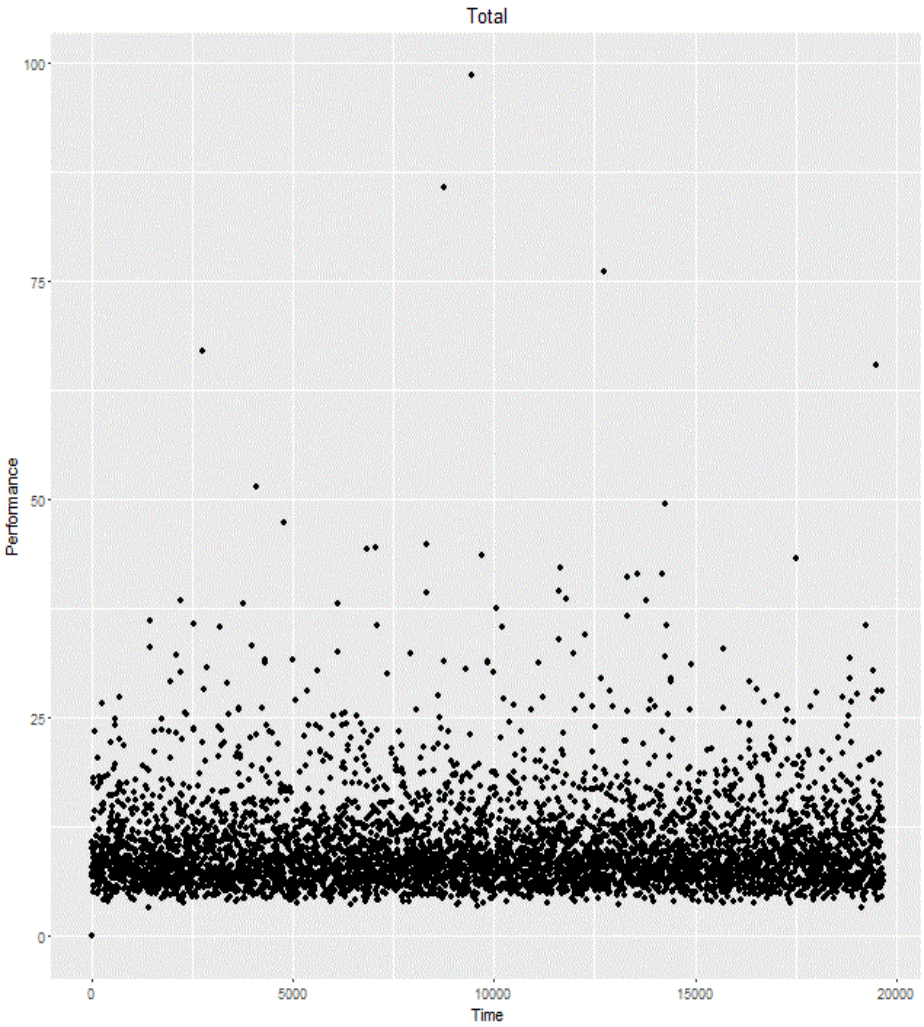


시뮬레이션 결과

(각 CPU의 TOTAL PERFORMANCE)



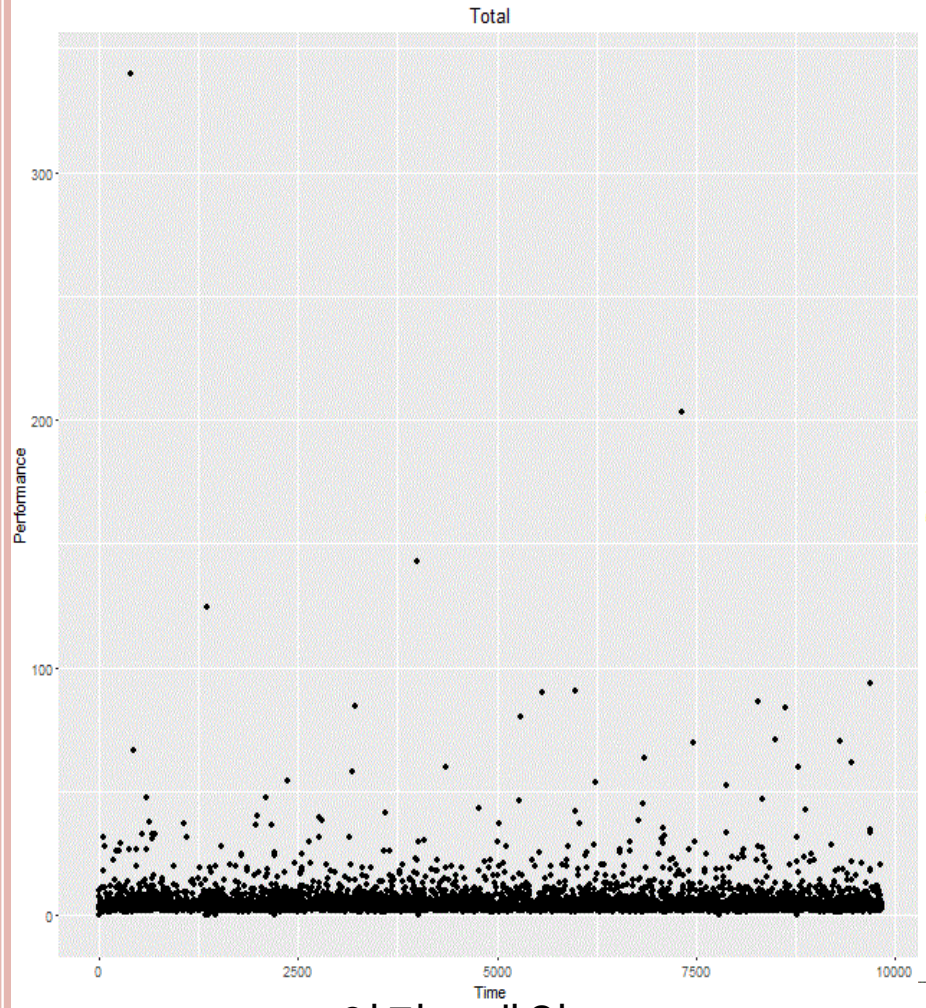
코어가 2개인 CPU



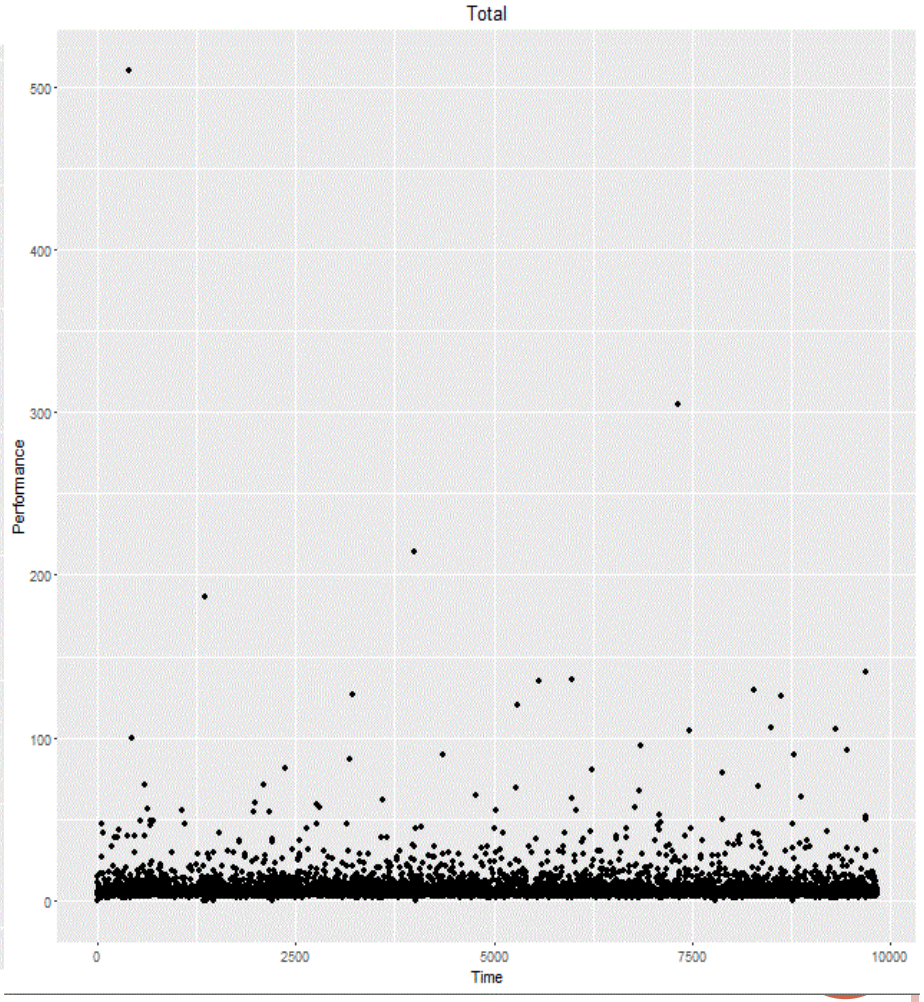
코어가 4개인 CPU

시뮬레이션 결과

(각 CPU의 TOTAL PERFORMANCE)



코어가 2개인 CPU
Clock : 2GHZ



코어가 2개인 CPU
Clock : 3GHZ

결론

- 시뮬레이션 결과 해석
  - Job이 같은 조건일 때, Core가 더 많은 CPU의 Performance가 더 높음을 알 수 있다.
  - 두 번의 실험모두 같은 수의 횟수만큼 루프를 실행했음에도 불구하고, , Core가 더 많은 CPU가 더 많은 시간 동안 작업을 했다. 즉 더 많은 Job을 처리했다.
  - X축인 Total Performance의 범위는 Core가 2개인 CPU가 더 크지만, 두 CPU의 Total Performance의 평균을 비교하면,

    CPU_2 : 5.526344

    CPU_4 : 10.10697 으로 Core가 4개인 CPU가 더 크다.

  - 코어는 같지만 Clock이 다른 경우, 두 CPU의 Total Performance의 평균을 비교하면,

    CPU_2(2GHZ) : 5.526344

    CPU_2(3GHZ) : 8.289516 으로 3GHZ인 CPU가 더 크다.

Q&A