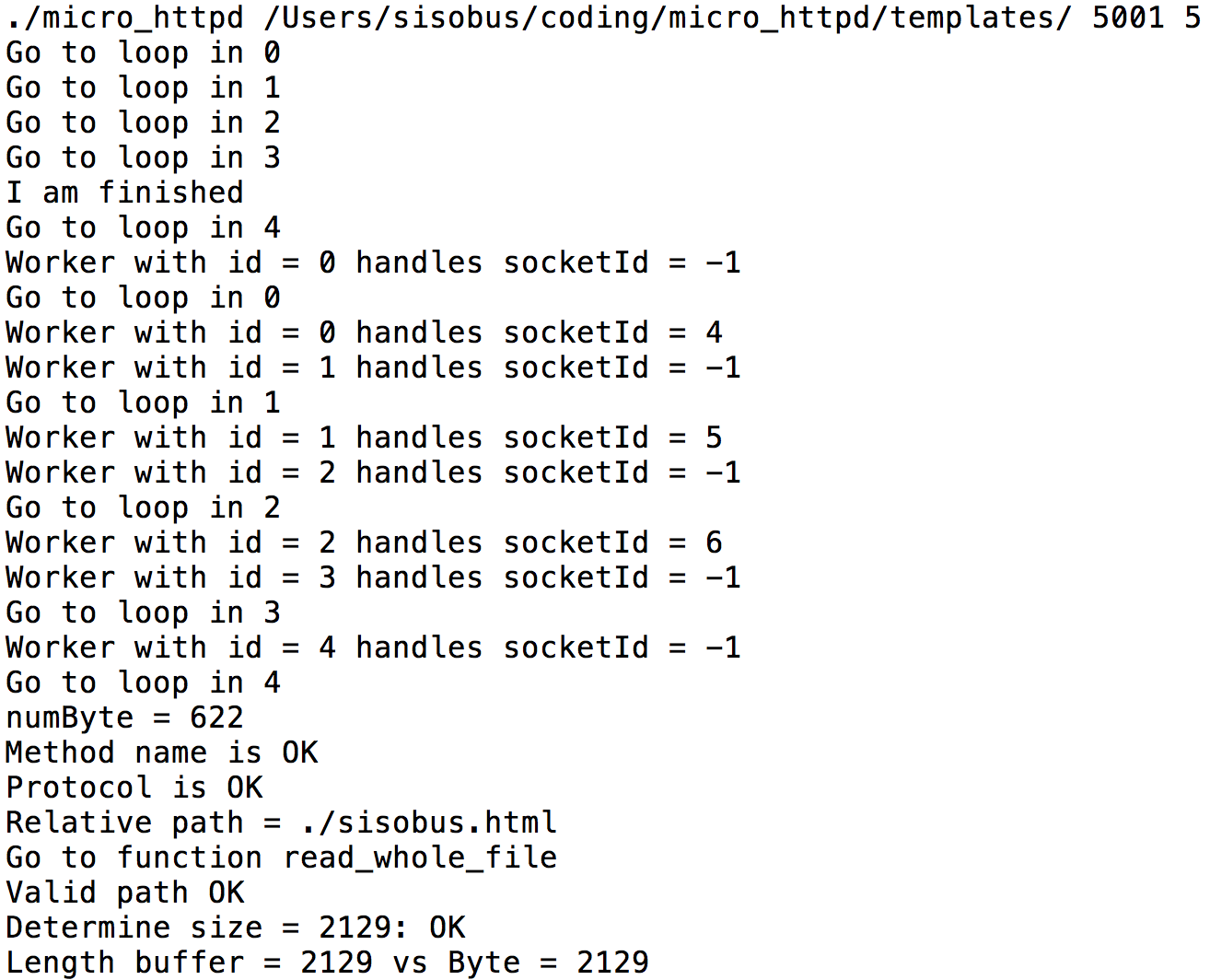
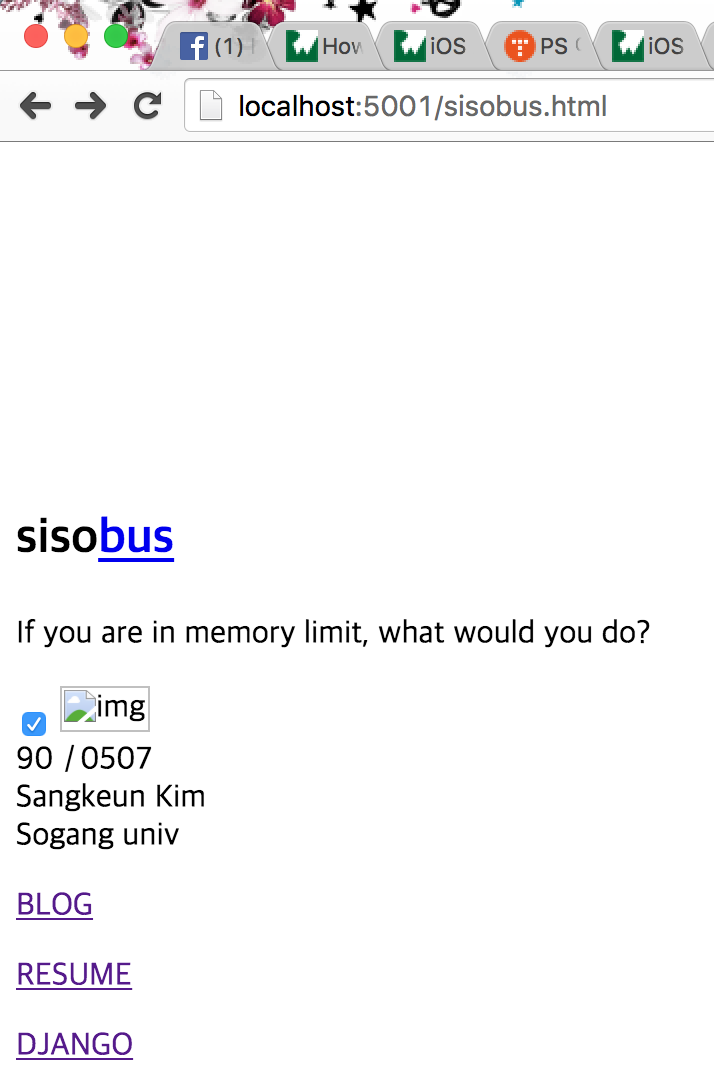
**Distributed System Assignment#2**

120150241 김상근

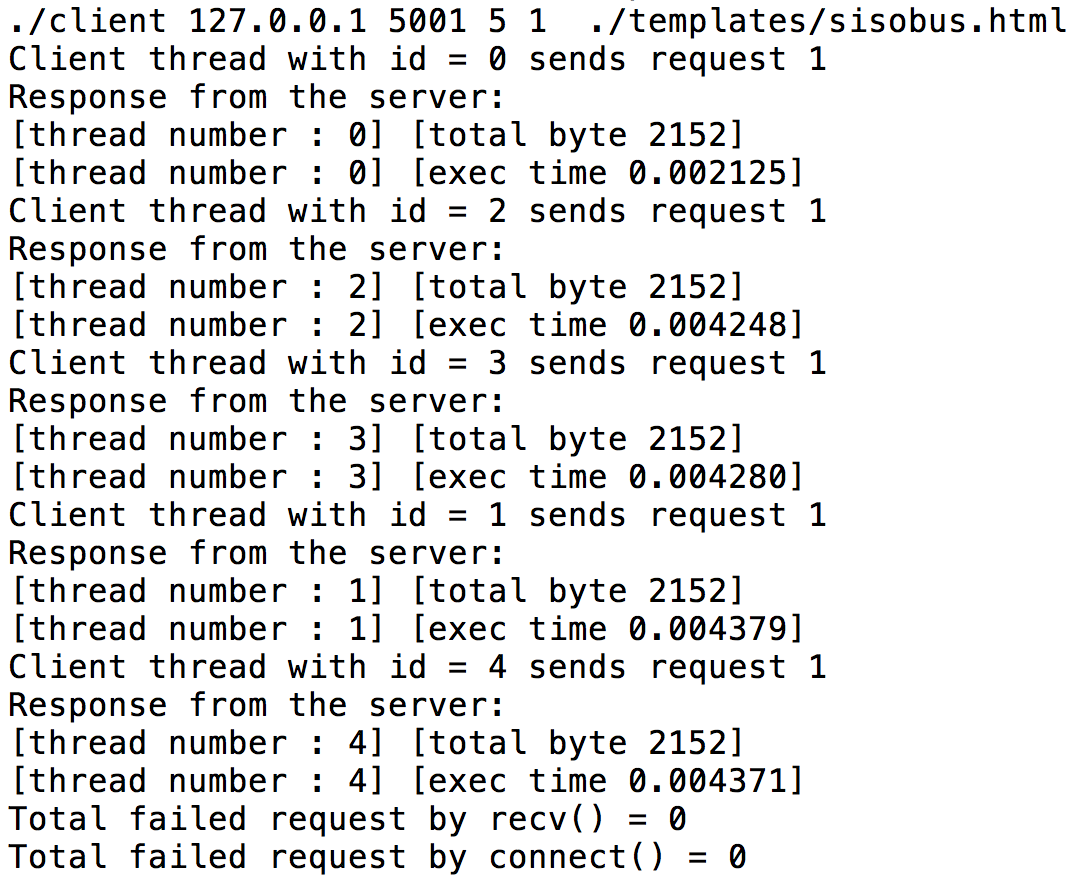
1. Test your web server using your client program



**[그림 1-1] server program**



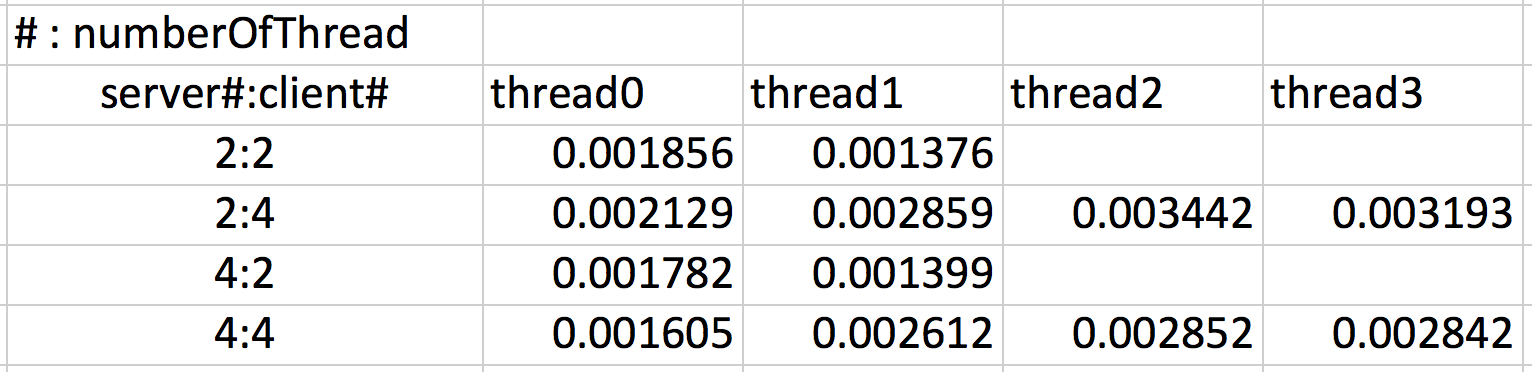
**[그림1-2] web browser**



**[그림1-3] client program**

[그림 1-1] 은 Server program을 base\_path로 /Users/sisobus/coding/-micro\_httpd/templates로 주고, port번호로 5001, thread의 개수로 5개를 주어 실행한 server이고, “Go to loop in 4” 아래쪽의 결과가 [그림1-2]의 web browser로 접속한 결과이다. [그림1-3]은 client program으로 localhost의 5001번 port 5개의 thread, 1번의 request로 ./template/sisobus.html에 접속한 결과이다. 해당 결과는 server program과 client program이 성공적으로 작동함을 보여준다.

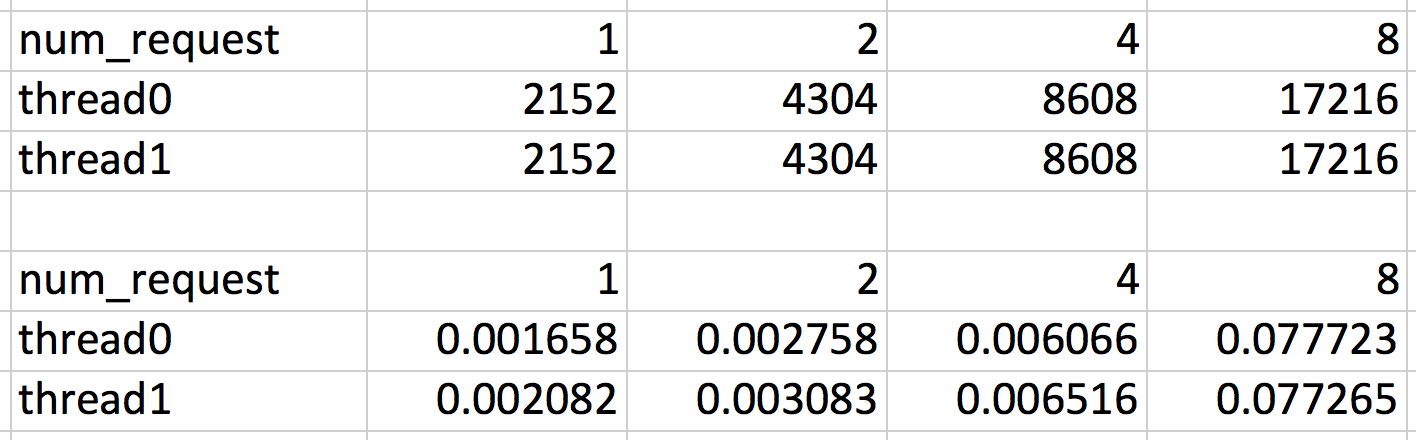
1. Vary the number of threads in both client and server and report your observations as well as the performance of the server

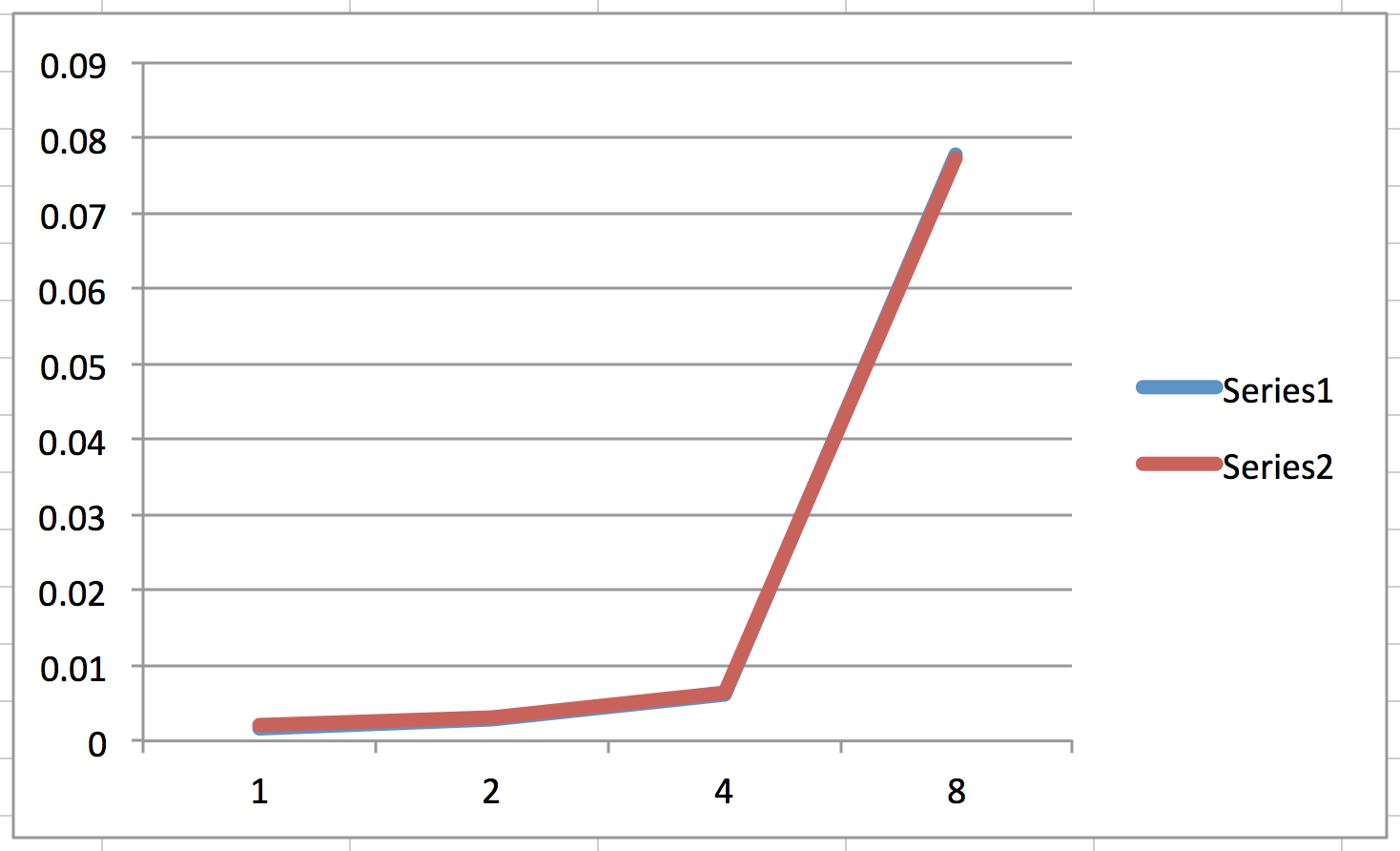


server와 client의 thread수를 달리하여 test해보았다. 여기서 소수값은 수행 시간을 측정한 결과이다. 이 결과로 Server thread가 client thread보다 클 때 비교적 더 빠름을 볼 수 있었다. 그다지 큰 용량이 아니어서 너무 빠른 수행 시간때문에 큰 차이를 나타내지는 않지만, server thread가 client thread보다 적을 때에는 어느정도의 지연시간 때문에 이러한 결과를 보이는 것 같다.

1. Vary also other parameters and report your observations as well as the performance of the server

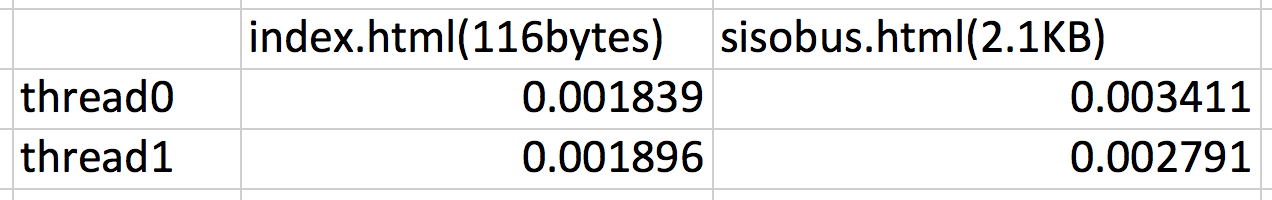
3-1. request를 증가시켜가며 time 비교





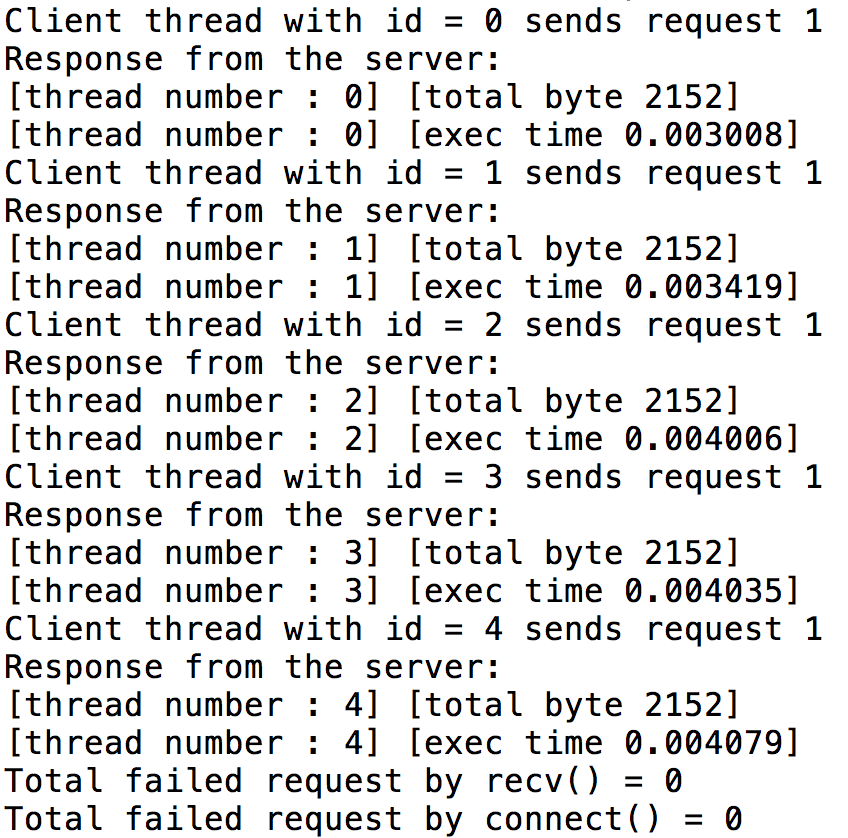
total bytes가 증가함에 따라서 수행 시간도 함께 증가하였다. 이로 보아, 파일 사이즈가 작은 것과 큰 것을 비교하여도 큰 것이 수행시간이 더 늦을 것으로 예상할 수 있다.

3-2. 파일 사이즈가 다른 request의 비교



위에서 예상한대로 파일이 큰 것이 더 오래걸림을 보여준다.

1. Try to modify the server code so that it uses the thread-per-request architecture and compare the performance between the two architectures for various conditions
2. Be careful of the caching effect. Since you are making repeated requests, your files are likely to be cached. Check the performance of your server when the files are cached, thereby a little disk I/O may occur. How much is the performance benefit in this environment ? Try to make your server having substantial I/O and check the performance benefits. Report your observations.



위의 결과를 보면 thread 2,3,4의 수행 시간은 0.004즈음인 것에 비하여 thread 0,1의 수행시간은 0.003대로 빨라진 것을 볼 수 있다. 따라서 thread2,3,4가 먼저 파일을 읽고 cache에 저장한 다음 thread 0,1이 cache를 읽은 것으로 보인다. 만약 파일의 크기가 크다면 빨리 수행한 thread와 늦게 수행한 thread의 수행시간 차이가 더 커질 것이다.

1. When implementing a web server using worker pool architecture, the number of threads in the pool is a parameter we need to consider. If we increase the number of threads, the performance (i.e., latency) may be improved. But the resource may be over-provisioned by creating multiple threads in advance. Suggest and implement your own solution so that we have an (near) optimal number of threads in the pool. Test and report your observations.

위 실험들을 해본 결과 수행시간에 가장 영향을 끼치는 것은 당연히 request의 수였다. Thread는 많으면 많을수록 지연시간을 더디게 하므로 좋겠지만, 무작정 많이 만들면 pool에서 대기하는 thread의 오버헤드가 있으므로 적당히 주어야 한다. 따라서 최적의 thread의 수는 request가 몰릴 때 많이 주고, 적을 때에는 최소한의 thread만 남겨놓는 것이 좋을 것 같다.