

2017

DESIGN DOCUMENT-PA1

DESIGN

VIKRAM G AND MOUNA GIRI

DESIGN

This programming assignment consists of four components:

1. CPU Benchmark
2. Memory Benchmark
3. Disk Benchmark
4. Network Benchmark

CPU Benchmark:

The code is written in C language.

For multi-threading, we have used the pthread library.

Each of the threads would calculate the FLOPS and IOPS one after the other.

We have automated the running of the program which will run 1,2,4,8 threads automatically.

In the program, we have an integer and floating operation with 14 operations each, we are calculating the Latency and throughput with different threads running the program for few 1000's of loops.

Memory Benchmark:

This benchmark is divided into three different programs to perform the below operations:

1. Read and Write Sequential
2. Sequential Write
3. Random Write

The code is written to check for different level of concurrency by testing for 1,2,4 and 8 Threads. The code is written in C language.

When the code is executed it asks for different sizes of the blocks [8b, 8kb, 8M and 80MB] on which the program is run. Based on the user's input, the code is executed.

We have used memcpy() and memset() function. In the program, the time taken to the tasks is noted. And from the time, we have calculated latency and throughput.

Disk Benchmark:

This benchmark is divided into three different programs to perform the below operations:

4. Read and Write Sequential
5. Sequential Read
6. Random Read

The code is written to check for different level of concurrency by testing for 1,2,4 and 8 Threads. The code is written in Java.

When the code is executed it asks for different sizes of the blocks [8b, 8kb, 8M and 80MB] on which the program is run. Based on the user's input, the code is executed.

In DiskTasks_Sequential – The program does read the given file sequentially based on the entered block size for the threads 1, 2, 4 and 8.

In DiskTasks_Random – The program does read the given file randomly based on the entered block size for the threads 1, 2, 4 and 8.

In DiskTasks_ReadnWrite– The program does read and writes the given file sequentially based on the entered block size for the threads 1, 2, 4 and 8.

In all the above three programs, the time taken to the tasks is noted. And from the time, we have calculated latency and throughput.

Network Benchmark:

The benchmark can be done on both TCP and UDP.

There are 2 files for each of TCP and UDP. One is client and the other is server.

Varying blocks of packets/ buffer size is transferred from client to server. While at server's end, numbers of bytes received are transferred back to the client.

Throughput and Latency are calculated based on the time taken.

Improvements and Extensions to the program:

CPU Benchmark:

- Complicated operations can be used instead of just the current simple operations.
- Once the program is executed, the files are written to text file.
- More than 4 threads can be used.
- GUI can be implemented.
- The functions can be performed for more time.

Memory Benchmark:

- GUI can be Implemented.
- More threads can be used.
- Could work on the cache effects.
- We have used memcpy() and memset() functions to test this benchmark. We can also use alternative methods to test such as swap pointers only, not the data itself etc.

Disk Benchmark:

- More threads can be used.
- The JVM takes time to warm up before we can get appropriate results. The results can be improved by using other languages like C and C++.
- Could Use Memory mapped Byte Buffers for better results.

Network Benchmark:

- This program to wireless networks.
- We could implement to the other node.
- More threads can be used.
- To avoid data loss while transferring it from client to server we can use more strong architecture.
- The Latency and Throughput could be improved with better connections.