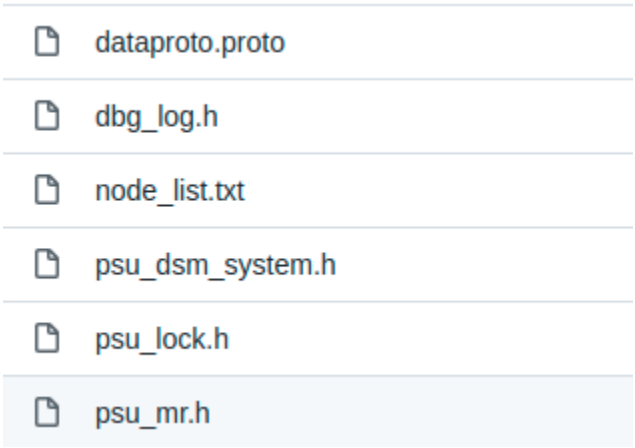
Project 1: Distributed Threads with Distributed Shared Memory System

-Abhijeet Kumar(amk7371), Abhishek Kumar (azk6085)

Directory structure



1. Dataproto.proto => Protocol definition for Distributed mutex lock and distributed shared memory. Protobuf file needed for gRPC calls.
2. dbg_log.h => Redirecting dbg cout to a file (keep printf for important messages in terminal).
3. node_list.txt => list of hostname participating in DisSys.
4. psu_dsm_system.h => Library file with all the definition of invalidation-based DSM system.
5. psu_lock.h => Implementation of Ricarta-Agarwala Algorithm for distributed mutex.
6. psu_mr.h => Map Reduce framework in distributed system.

Application code

Apart from testcode provided we developed 2 more applications to verify functionality of MapReduce framework. Each of the folder in repo has the softlink to node_list.txt, and makefile which includes all the necessary header files and creates an executable.

project-2-dsm-and-map-reduce-framework-abhijeeetkumar/**word_count**/app2.cc

project-2-dsm-and-map-reduce-framework-abhijeeetkumar/**k_means**/app4.cc

To Run any test-

cd <test folder>;

make clean;

make;

./.app4 <input txt file>

dbg_log_<machine_name>.txt is created with higher order of verbosity in log.

Sequential consistency-

In this experiment, we are running 3 different programs (P1, P2 & P3) on three different machine (135-08, 135-13, 135-01) and all of them sharing variable (a and b initialized by 0). Here in this scenario, P1 is setting variable a = 1, where as P2 is setting b = 1 only if(a == 1) , and P3 is printing a & b if both of them are non-zero!

By analyzing the prints of all three machines, we can observe that initially P3 alone had the RW rights to that address space. Then P1 transitioned its state from Invalid to ReadWrite to set a =1; and then later P2 changed its state to Read to exit from busy-waiting loop and set b = 1. Once a & b is set, P3 read both the pages and print the value.

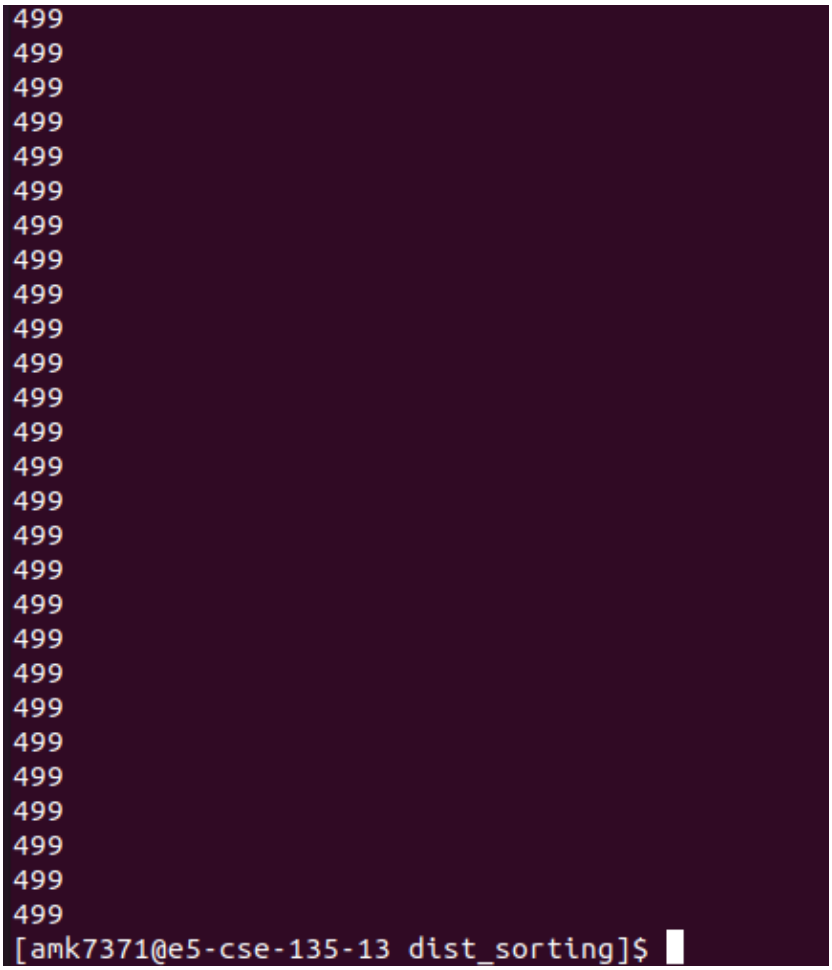
```
[amk7371@e5-cse-135-08 seq-consistency]$ ./p1
setting 0 as protect_flag for 0x7cf000 size 8192
: Invalid write attempt to 0x7cf000
setting 3 as protect_flag for 0x7cf000 size 4096
setting 1 as protect_flag for 0x7cf000 size 4096
setting 0 as protect_flag for 0x7d0000 size 4096
[amk7371@e5-cse-135-08 seq-consistency]$
```

```
[amk7371@e5-cse-135-13 seq-consistency]$ ./p2
setting 0 as protect_flag for 0x7cf000 size 8192
: Invalid read attempt from 0x7cf000
setting 3 as protect_flag for 0x7cf000 size 4096
setting 1 as protect_flag for 0x7cf000 size 4096
setting 0 as protect_flag for 0x7cf000 size 4096
: Invalid read attempt from 0x7cf000
setting 3 as protect_flag for 0x7cf000 size 4096
setting 1 as protect_flag for 0x7cf000 size 4096
: Invalid write attempt to 0x7d0000
setting 3 as protect_flag for 0x7d0000 size 4096
setting 1 as protect_flag for 0x7d0000 size 4096
setting 1 as protect_flag for 0x7cf000 size 4096
```

```
[amk7371@e5-cse-135-01 seq-consistency]$ ./p3
0x7cf000 0x7d0000setting 3 as protect_flag for 0x7cf000 size 8192
setting 1 as protect_flag for 0x7cf000 size 4096
setting 0 as protect_flag for 0x7cf000 size 4096
setting 0 as protect_flag for 0x7cf000 size 4096
setting 0 as protect_flag for 0x7d0000 size 4096
: Invalid read attempt from 0x7d0000
setting 0 as protect_flag for 0x7d0000 size 4096
setting 3 as protect_flag for 0x7d0000 size 4096
setting 1 as protect_flag for 0x7d0000 size 4096
: Invalid read attempt from 0x7cf000
setting 3 as protect_flag for 0x7cf000 size 4096
setting 1 as protect_flag for 0x7cf000 size 4096
a = 1 and b = 1
```

Distributed Sorting-

This is a stress testing of our DisSys where all the addresses being accessed are not page aligned, here from each address we were trying to extract the page number and trying to copy the whole page. We also encountered an issue where other undesired global variables being allocated on the same virtual page as ours and causing us extra interruption. In order to avoid that we installed a guard band from the user side, which is page aligned this ensure the very next allocation happens on the page which is not shared by others.



Word Count

Using our DSM and distMutex as library we developed a MapReduce framework for our distributed system. Our framework exposes the user with an option where he can scale his multi-threaded software to distributed system. We provided an mpi, where the programmer can register the map and reduction function and register the shared memory either in data segment or in heap and work on the same memory space concurrently.

```
Word : hello  Occurance : 273
setting 0 as protect_flag for 0x7e3fd4 size 4096
: Invalid write attempt to 0x7df000
setting 3 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7e3fd4 size 4096
setting 1 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
[amk7371@e5-cse-135-01 word_count]$
```

```
Word : happy  Occurance : 230
setting 0 as protect_flag for 0x7e3fd4 size 4096
setting 0 as protect_flag for 0x7e3fd4 size 4096
setting 1 as protect_flag for 0x7e1000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7e3fd4 size 4096
: Invalid read attempt from 0x7df000
setting 1 as protect_flag for 0x7e1000 size 4096
setting 3 as protect_flag for 0x7df000 size 4096
setting 1 as protect_flag for 0x7df000 size 4096
: Invalid write attempt to 0x7df000
setting 1 as protect_flag for 0x7e2008 size 4096
setting 3 as protect_flag for 0x7df000 size 4096
setting 1 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
[amk7371@e5-cse-135-10 word_count]$
```

```
Word : world  Occurance : 242
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7e3fd4 size 4096
setting 0 as protect_flag for 0x7e3fd4 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
: Invalid read attempt from 0x7df000
setting 3 as protect_flag for 0x7df000 size 4096
setting 1 as protect_flag for 0x7df000 size 4096
: Invalid write attempt to 0x7df000
setting 3 as protect_flag for 0x7df000 size 4096
setting 1 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
[amk7371@e5-cse-135-13 word_count]$
```

```
setting 1 as protect_flag for 0x7e2008 size 4096
Word : psuni  Occurance : 275
setting 0 as protect_flag for 0x7df000 size 4096
setting 0 as protect_flag for 0x7df000 size 4096
: Invalid read attempt from 0x7df000
setting 3 as protect_flag for 0x7df000 size 4096
setting 1 as protect_flag for 0x7df000 size 4096
: Invalid write attempt to 0x7df000
setting 3 as protect_flag for 0x7df000 size 4096
[amk7371@e5-cse-135-08 word_count]$
```

Kmeans

We also implemented an application based on the system we develop which provides us k cluster means. We used our distributed MapReduce programming model to allow different machines to work on the same set of data.

```
[amk7371@e5-cse-135-08 k_means]$ ./app4 Testcases/input2.txt
setting 0 as protect_flag for 0x7e1000 size 4096
setting 0 as protect_flag for 0x7e3000 size 392
setting 0 as protect_flag for 0x7e3000 size 4096
setting 0 as protect_flag for 0x7e3000 size 4096
setting 0 as protect_flag for 0x7e3000 size 4096
setting 0 as protect_flag for 0x7e1000 size 4096
: Invalid read attempt from 0x7e3000
setting 3 as protect_flag for 0x7e3000 size 4096
setting 1 as protect_flag for 0x7e3000 size 4096
: Invalid write attempt to 0x7e3000
setting 3 as protect_flag for 0x7e3000 size 4096
setting 0 as protect_flag for 0x7e1000 size 4096
setting 0 as protect_flag for 0x7e1000 size 4096
: Invalid read attempt from 0x7e1000
setting 3 as protect_flag for 0x7e1000 size 4096
setting 1 as protect_flag for 0x7e1000 size 4096
: Invalid write attempt to 0x7e1000
setting 3 as protect_flag for 0x7e1000 size 4096
setting 1 as protect_flag for 0x7e1000 size 4096
setting 1 as protect_flag for 0x7e3004 size 4096
setting 0 as protect_flag for 0x7e3004 size 4096
: Invalid write attempt to 0x7e1000
setting 3 as protect_flag for 0x7e1000 size 4096
setting 0 as protect_flag for 0x7e3004 size 4096
New centroid. 0.173750 0.156250
New centroid. 0.616154 0.663077
New centroid. 0.605000 0.270000
New centroid. 0.963333 0.213333
[amk7371@e5-cse-135-08 k_means]$
```