Time Bomb

Aaron Morgenegg

09/14/18

OpenMPI Time Bomb Program

This is a simple time bomb program. To start, process 0 will initialize a timer with a random value. It will pass this timer to a random process, and output this passing to the console. Then, the program loops. While the timer is positive, the process with the timer will decrement it, and pass it to another random process. If it hits zero, the process announces it has blown up and will alert the other processes to end the loop.

```
// Time Bomb OpenMPI c++ implementation
#include <iostream>
#include <time.h>
#include <stdlib.h>
#include <mpi.h>
#include <unistd.h>

void Kaboom(int world_rank, int world_size){
std::cout << world_rank << " went KABOOM! Somebody set up us the bomb\n";
int terminate = -1;
for(unsigned int i = 0; i < world_size; i++){
MPI_Send(&terminate, 1, MPI_INT, i, 0, MPI_COMM_WORLD);
}
}

int getTarget(int world_size){
return rand()%world_size;
}
```

```
int main(int argc, char** argv) {
// initialize MPI
MPI_Init(&argc, &argv);

// Initialize rng
srand(time(NULL));

// stores number of processes in world_size
int world_size;
MPI_Comm_size(MPI_COMM_WORLD, &world_size);

// Get the rank of this process
int world_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

int timer = 1;

if(world_rank == 0){
timer = rand() % 15 + 1;
std::cout << "Lighting the fuse for a " << timer << " second bomb\n";
int target = getTarget(world_size);
std::cout << "Passing the bomb from " << world_rank << " to " << target << std::en
MPI_Send(&timer, 1, MPI_INT, target, 0, MPI_COMM_WORLD);
}

while(1){
MPI_Recv(&timer, 1, MPI_INT, MPI_ANY_SOURCE, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE)
sleep(1);
if(--timer == 0){
Kaboom(world_rank, world_size);
} else if(timer < 0) break;
else {
```