

CS 423 MP3 README and Analysis

Hongpeng Guo *hg5*

How to run the code.

1. `./prerun.sh <major number>`
(This step will make all the kernel files, install the kernel, as well as make node corresponding to the device major number. Note that you need to know the major number before run this code. You can follow the instructions in MP3.pdf to check the major number.)
2. We have three test scenarios according to the assignment. The first two are for thrashing and locality. The third is for multi-programming.

```
./run1.sh  
./monitor > profile1.data
```

```
./run2.sh  
./monitor > profile2.data
```

```
./run3_1.sh  
./monitor > profile3_1.data
```

```
./run3_2.sh  
./monitor > profile3_2.data
```

```
./run3_3.sh  
./monitor > profile3_3.data
```

Analysis of the results.

1. Experiment 1

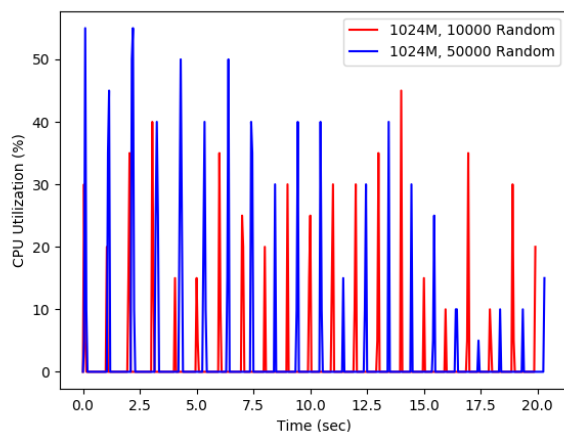


fig1. cpu utilization

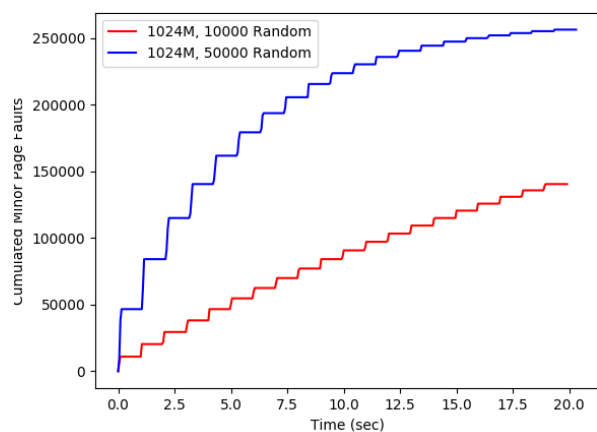


fig2. cumulated minor page faults

2. Experiment 2

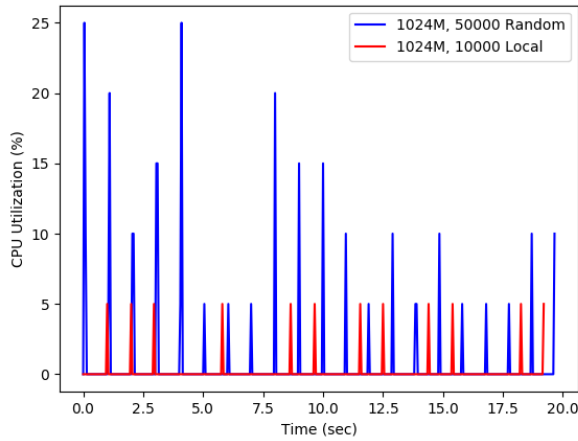


fig3. cpu utilization

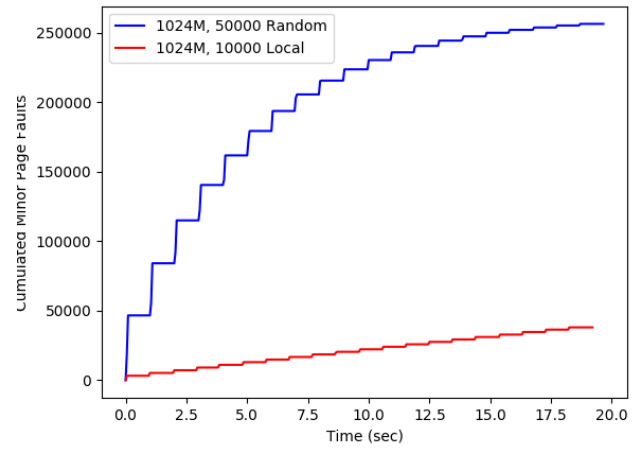


fig4. cumulated minor page faults

3. Experiment 3

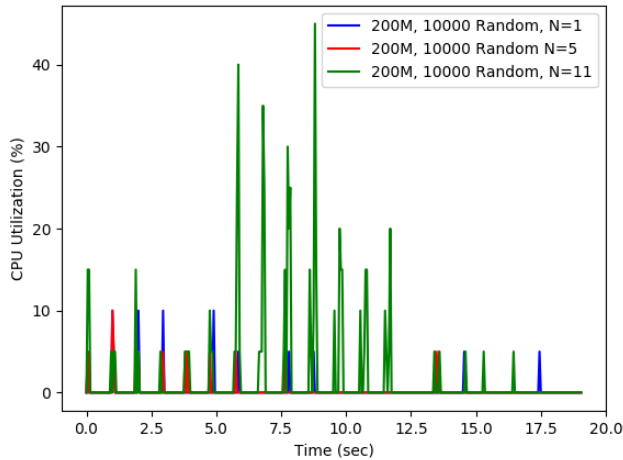


fig5. cpu utilization

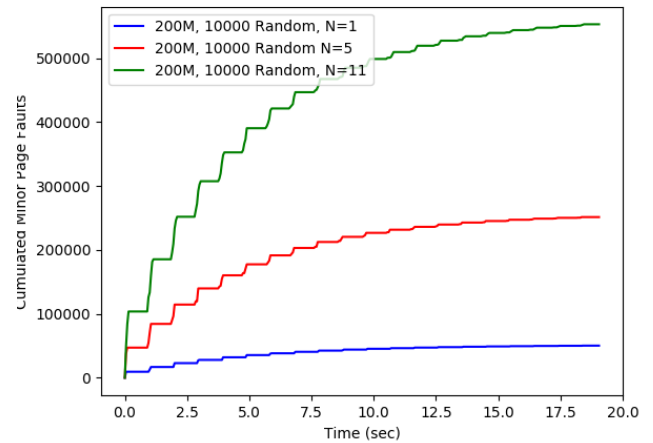


fig6. cumulated minor page faults

From the above experiment results, we observe that:

- (1) More access in each iteration will result in higher cpu utilization and cumulated page faults. Page faults accumulated fast at the beginning and slow in the end.
- (2) Locality memory access will greatly reduce cpu utilization ratio as well as page faults number. Because the locality increase the memory hit rate.
- (3) Multi-programming will increase both cpu utilization as well as the number of page faults.