MP4 Report

Jialu Zhao

1.Measure the performance of the system with varying numbers request channels and sizes of the buffer.

Default: -n 10000 –w 10 –b 100

```
                                    Histogram
Name         0-9  10-19  20-29  30-39  40-49  50-59  60-69  70-79  80-89  90-99  Total
Joe Doe     1008   1013   1022    982    960    983    988    982   1032   1030  10000
Jane Smith 1011    990   1027    993   1015   1014   1001    989    946   1014  10000
John Smith 1028    985    989    961   1010    961   1019    988   1040   1019  10000
 done (control).
New request is quit
The total time is:11.0298s
```

(1) Only increase numbers of worker channels

Number of worker channels	Total time(MP4)	Total time(MP3)
10	11.0298s	10.9409s
15	7.40159s	7.33623s
20	5.5632s	5.52317s
25	4.54797s	5.14366s
30	3.82856s	4.46089s

(2) Only increase the size of the buffer

Size of buffer	Total time
100	11.0298s
200	11.0051s

300	11.0359s
400	11.0402s
500	11.0133s

2.How does the performance compare to your implementation in MP3?

Compared with the implementation in MP3, there is not too much difference between them. When the number of channels get to 30, the implementation which used only one worker thread is faster.

3.Does increasing the number of request channels still improve the performance? By how much? Is there a point at which increasing the request channels does not further improve performance?

Increasing the number of request channels still improves the performance. When the channels increase from 10 to 30, the total time decrease from 10.9409s to 4.46089s.

For my laptop, when the number of channels get to 68, there is no further performance.

4. Bonus point.

Finish all the bonus, histogram can update every 10000 useconds.
list what part of the program you changed to make it resilient against signals:

```
int main(int argc, char * argv[]) {
  struct sigaction sa;
  struct itimerval timer;

  /* Install timer_handler as the signal handler for SIGVTALRM. */
  memset (&sa, 0, sizeof (sa));
  sa.sa_handler = &print_histogram;
  sigaction (SIGVTALRM, &sa, NULL);

  /* Configure the timer to expire after 250 msec... */
  timer.it_value.tv_sec = 0;
  timer.it_value.tv_usec = 10000;
  /* ... and every 250 msec after that. */
  timer.it_interval.tv_sec = 0;
  timer.it_interval.tv_usec = 10000;
  /* Start a virtual timer. It counts down whenever this process is
  executing. */
  setitimer (ITIMER_VIRTUAL, &timer, NULL);
  // read from command line
  int opt;
  // local variables exitsed in main function
  data_request_per_person = 10000;
  size_bounded_buffer = 50;
  number_worker_threads = 10;
```

Just add this in the front of the main function.