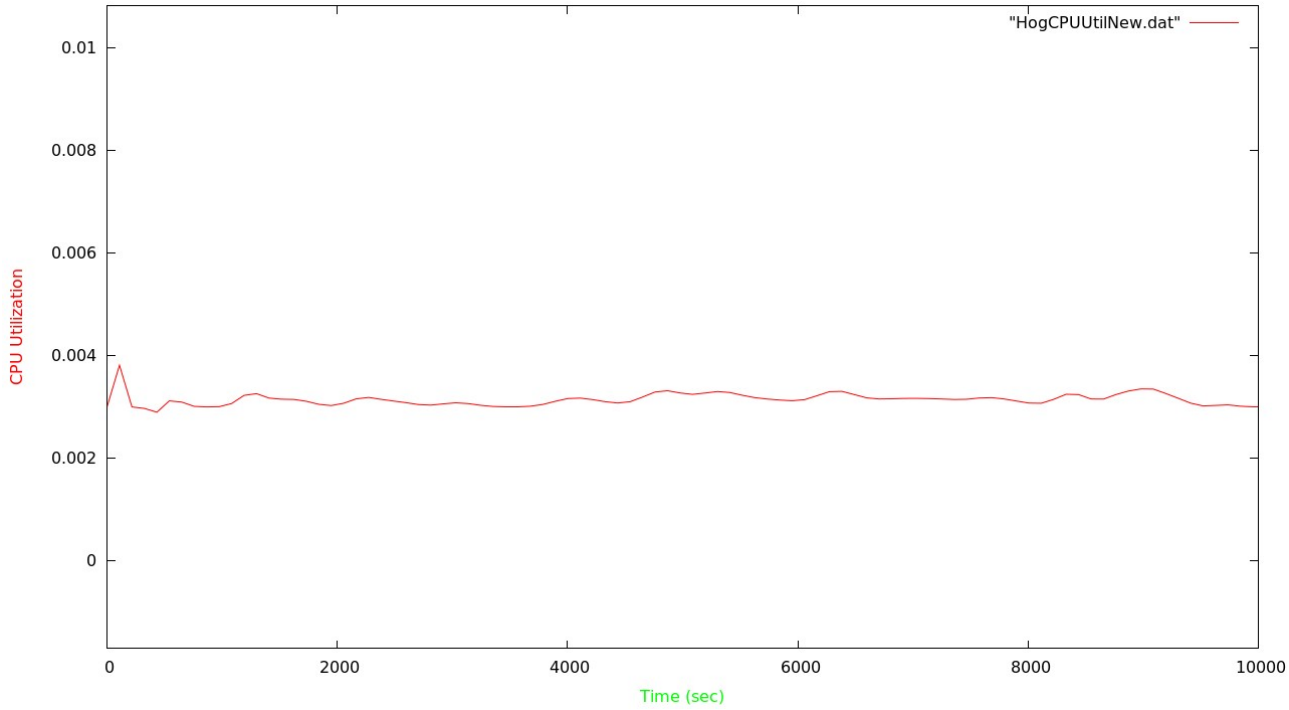
# PERFORMANCE REPORT: PERF TESTING ON DC++

Testing has been done using stress command in 3 situations – CPU intensive, IO intensive and CPU & IO intensive.
Test Machine – AMD A8 6410 (4 cores, 2 Ghz), 3.4 GB usable RAM. Running Ubuntu 15.04
Graphing – GNUPlot
Scripting – BASH

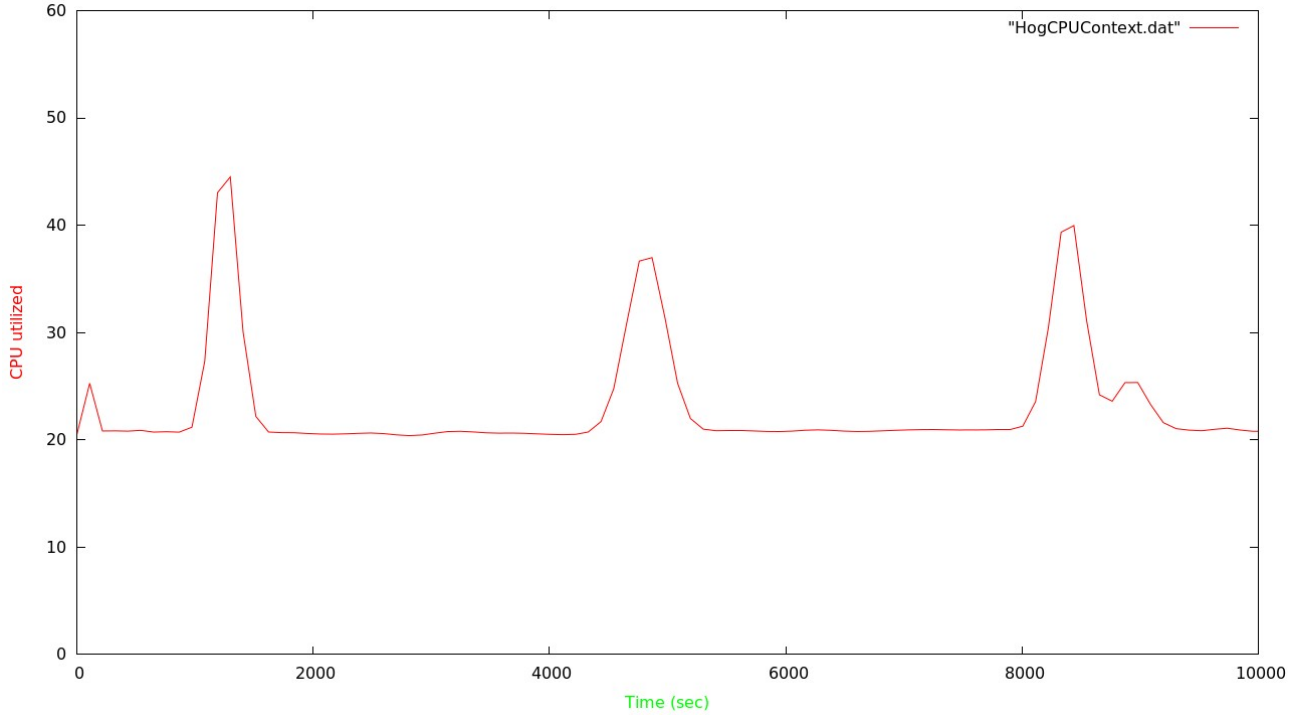
## I) CPU Intensive

50 workers were run using stress command to stress the CPU and perf was run to monitor LinuxDC++ every 10 seconds for 3 hours. Results were taken and graphs were plotted as follows :

a) CPU vs Time



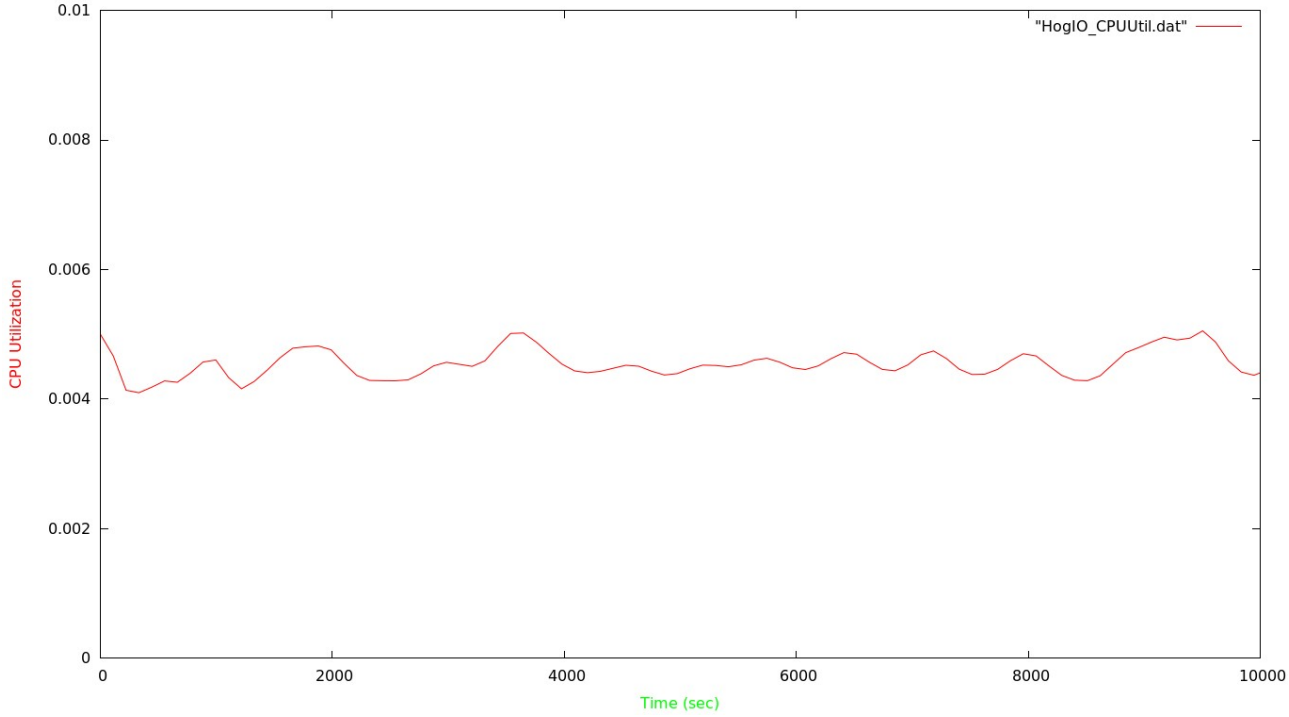
b) Context-Switches




DC++ was not set to download anything in this 3 hour time frame. We see spikes due to get requests for downloads by other users of DC. This increased context-switches momentarily as process needed to be swapped in and out to accommodate request of other users. CPU utilization also spiked around that area but is not visible in the graph as smooth bezier was used to plot the graph to get unified curved plot.

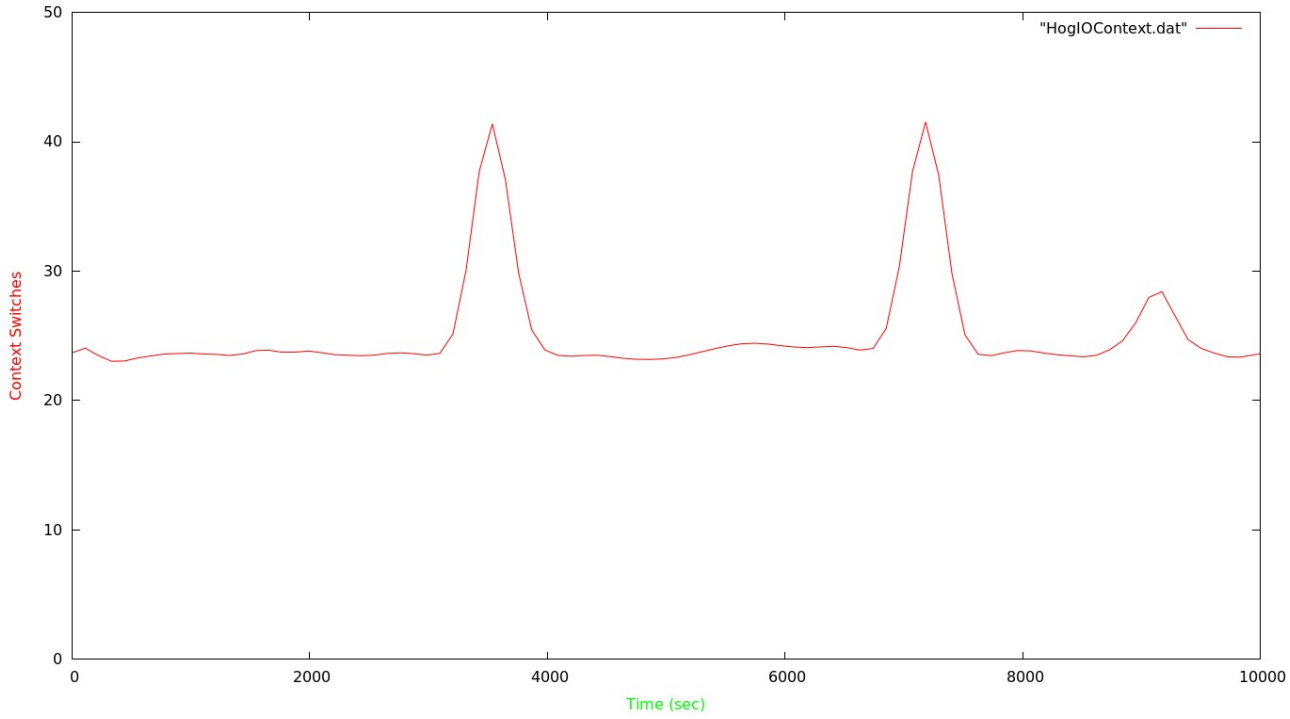
## II) IO Intensive

a) CPU vs Time



50 workers were set using -i flag which called sync() using stress command. Perf was run to monitor the LinuxDCPP in a three hour time frame.

b) Context-Switches vs Time

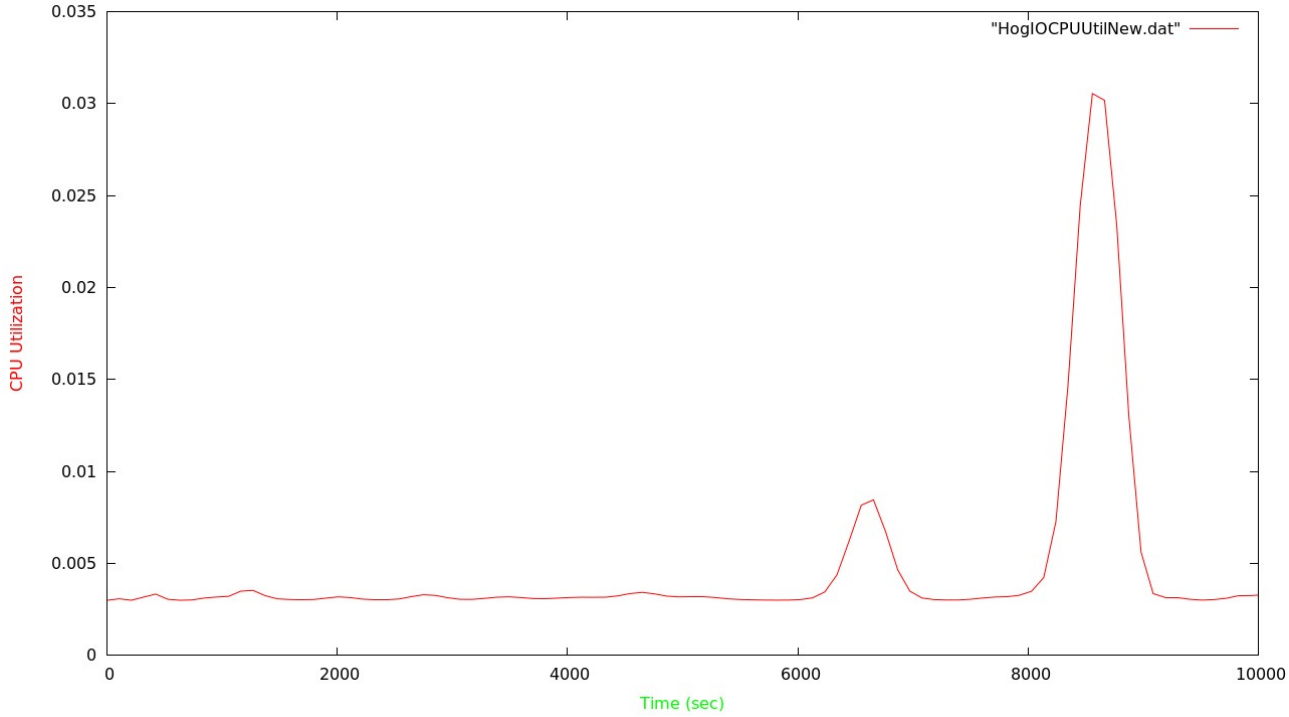



Again similar. Overall the CPU usage by the process was larger compared to when CPU intensive process was running. This was expected as CPU intensive processes totally hog the CPU and in comparison, DC++ is just switched out and doesn't take much CPU unless a download is initiated. The spikes in the graphs are because of get requests by DC users. So when the system is serving a file to a DC user the values shoot up (spikes) to indicate that it uses system resources and is getting switched out more often as it is scheduled more often.

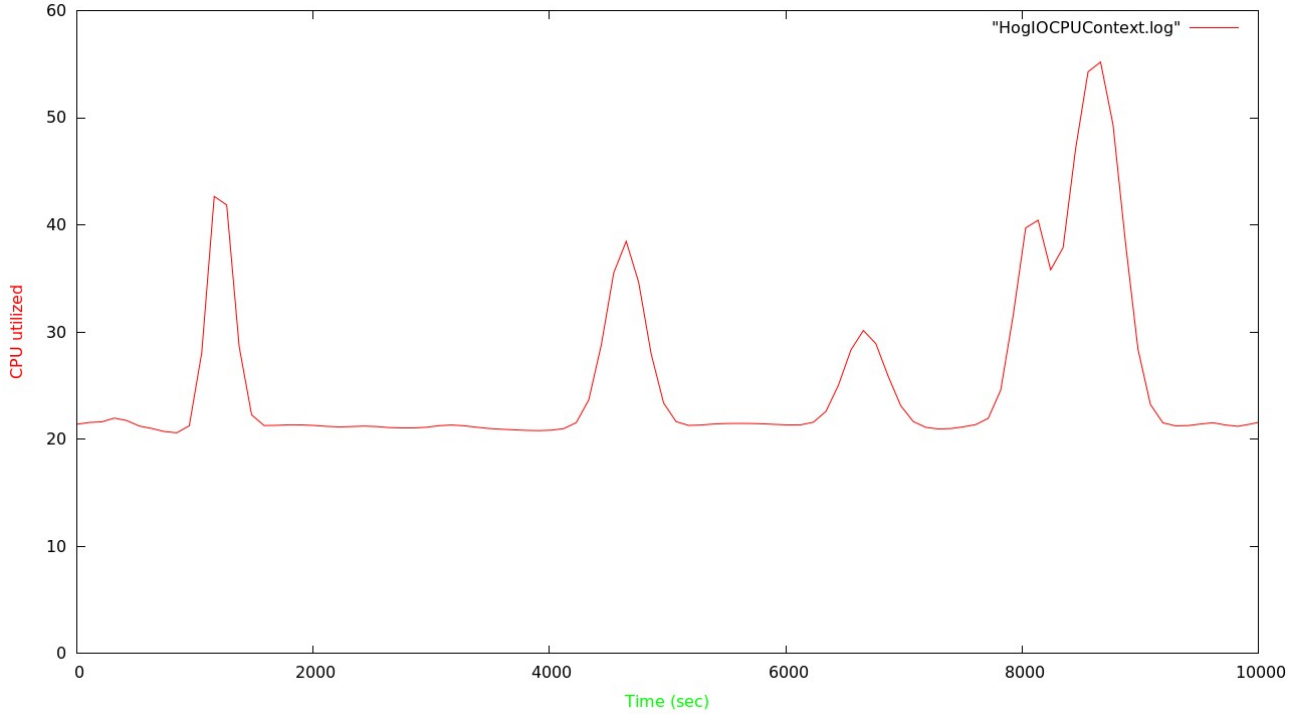
## III) CPU and IO Intensive

50 workers spinning (hogging CPU) and 50 workers spinning on sync() is run using stress command. Perf stats are collected on LinuxDC++ every 10 seconds. The results obtained have been plotted to graph below:

a) CPU vs Time




b) Context-Switches vs Time




The results are again almost similar except in case of CPU usage. The spikes as seen are get requests by DC users trying to download a file. That sparks CPU usage and context-switches by the DC client. The CPU usage is lower than that of the CPU usage during IO intensive operations as CPU is actually hogged here by CPU intensive operations which are also being concurrently run. The average number of Context-Switches are in between as compared to context-switches in case of other scenarios – IO intensive being the highest and CPU intensive being the lowest.