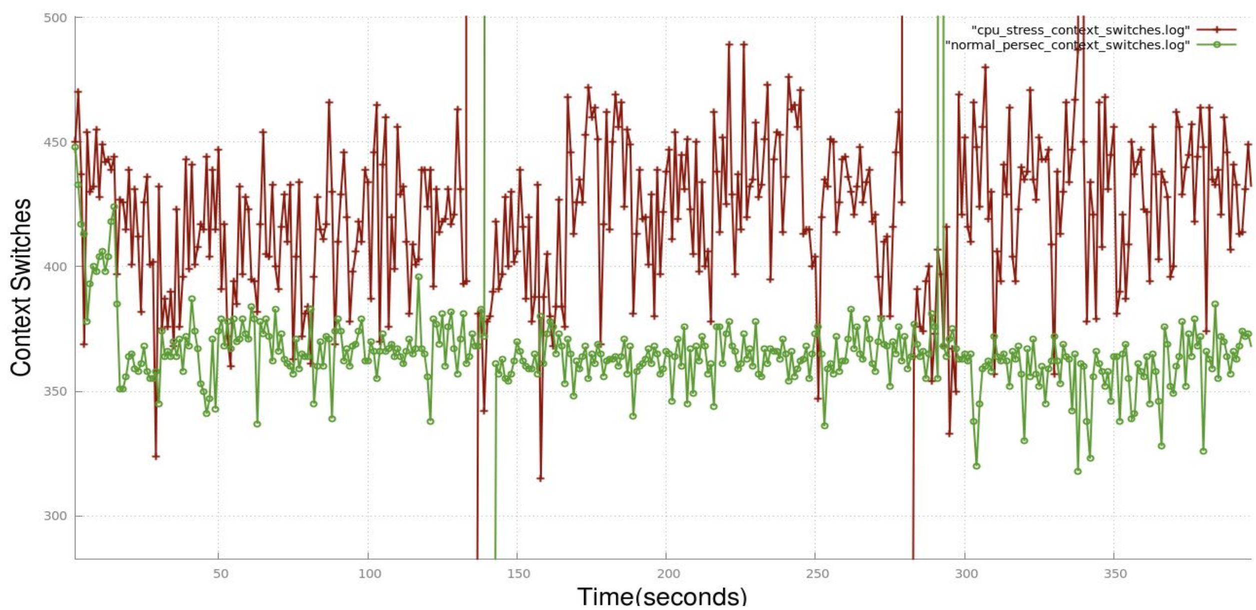
OS

Performance Analysis

Analysing the performance of Linux DC ++ in regular,CPU intensive and IO intensive environments



Setup

In our regular case we run Linux DC ++ on Ubuntu and we do not run any CPU or I/O intensive tasks in this case.We profile the application by executing perf stat after every 15 sec for about 3 hours and collect the data in a log file.

In our CPU intensive case we run Linux Dc ++ under heavy CPU intensive stress.This heavy CPU stress is simulated by the stress utility. Using this stress utility we were able to generate 40 CPU stressors.Each CPU stressor is nothing but a process spinning on an sqrt().

In our I/O intensive case we run Linux DC ++ under heavy I/O stress which again was simulated using stress utility.

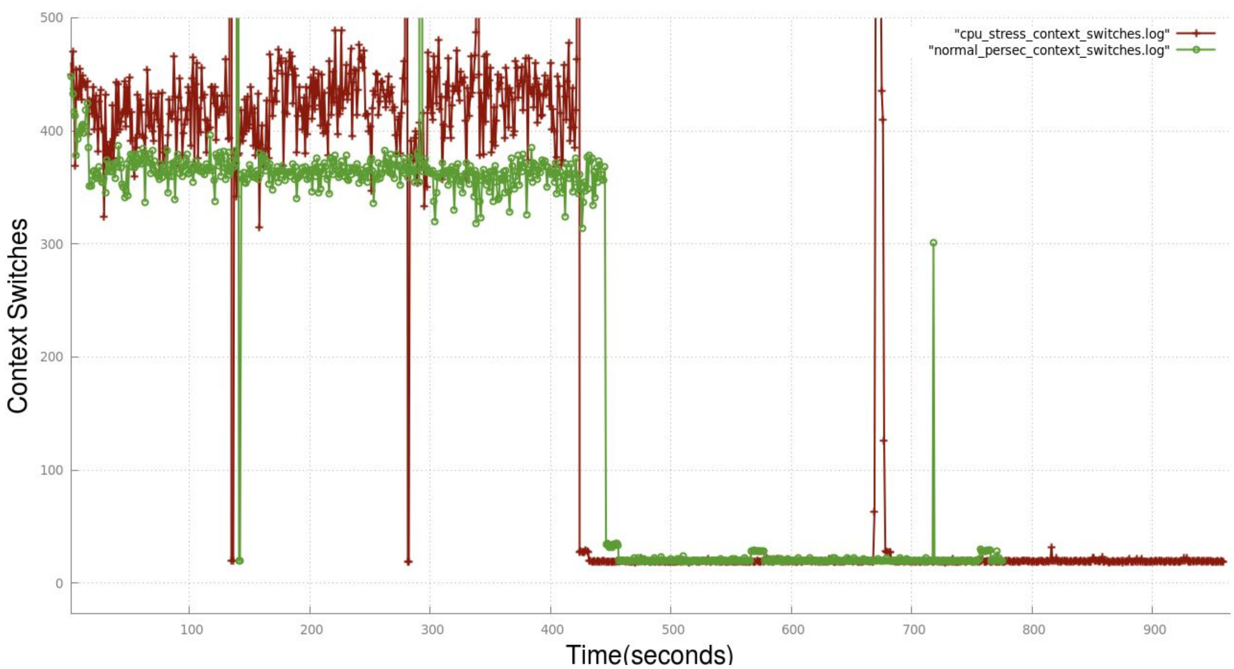
After obtaining the data we extract and clean the data which is relevant to us(context switches and CPU utilization) using *grep* and *sed* commands , in this report we plot a graph of CPU usage vs time and Context-switches vs time using gnuplot

In both the cases (CPU usage and Context switches)for the starting half of the time period we download some file from some peer pc and for the latter half we keep the Linux DC ++ idle i.e. we don't download any file.

Regular environment vs CPU intensive environment

Context Switches

Red - CPU Intensive   Green - Regular

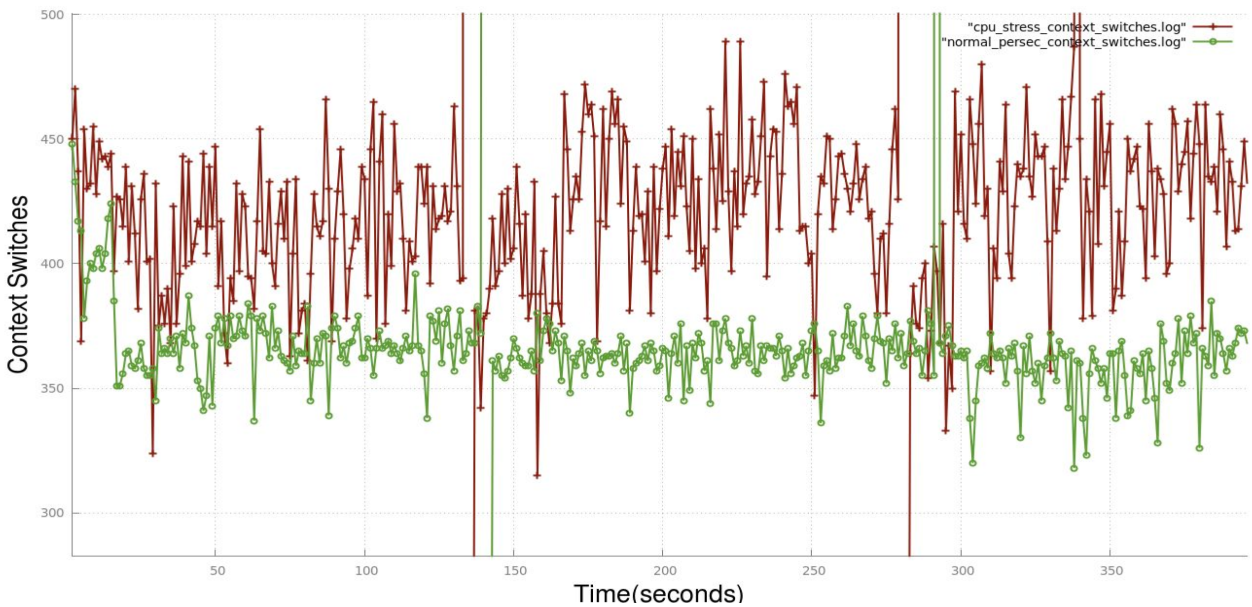


During starting half of the time period ,i.e the time period in which we download some files using Linux DC ++ the number of context switches in CPU intensive case is consistently higher as compared to normal case.

When DC++ lies idle the number of context switches is more or less same.

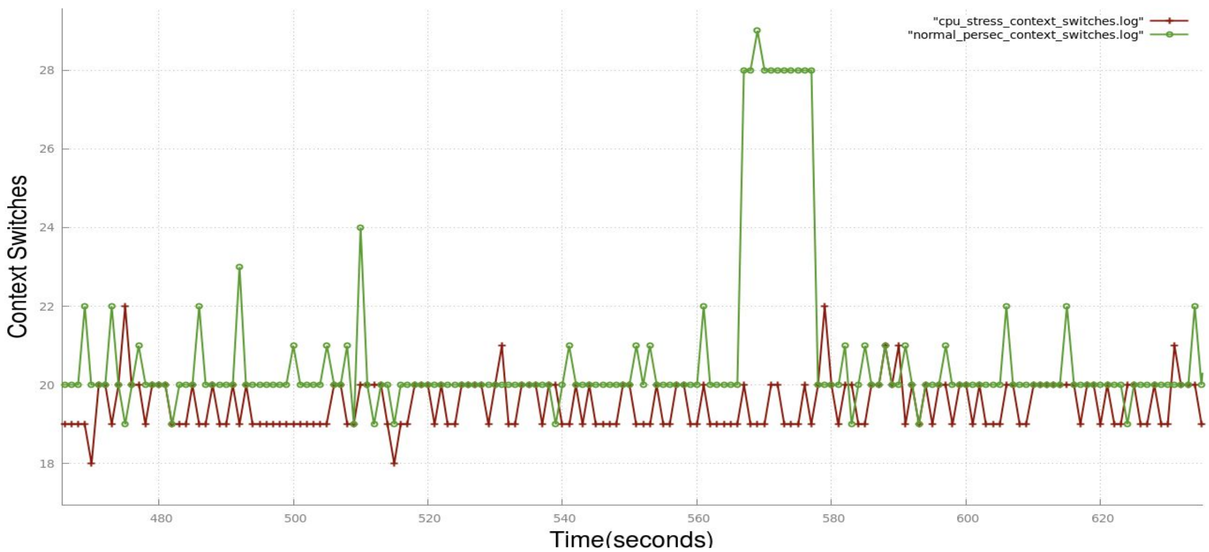
Deductions:

Graph zoomed showing context switches while a file is being downloaded/uploaded.



Since in the CPU intensive cases there are around 40 more process' which hog the CPU we have to context switch each of them in and out inorder to run the 40 process' + Linux DCpp fairly resulting in greater context switches.

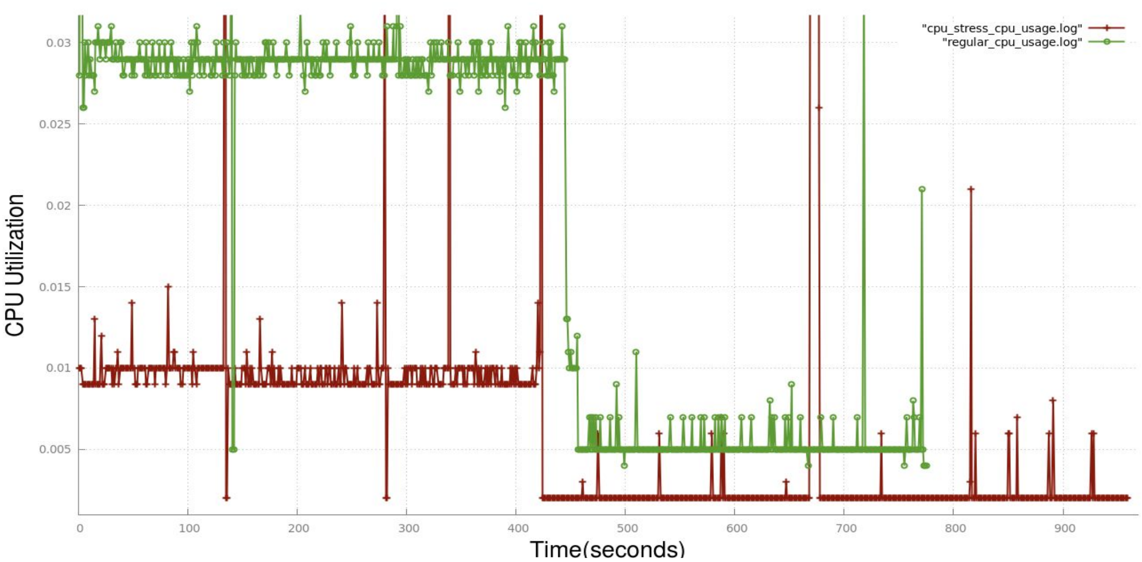
Graph zoomed showing context switches when Linux DC ++ is ide



In the latter half when no downloading is happening the result is more or less same because only those many programs of Linux DC ++ are executing which are required to keep the system up and running.Since less programs of Linux Dc ++ are running when it is idle the no. of context switches is lesser as compared to former case.

CPU Utilization

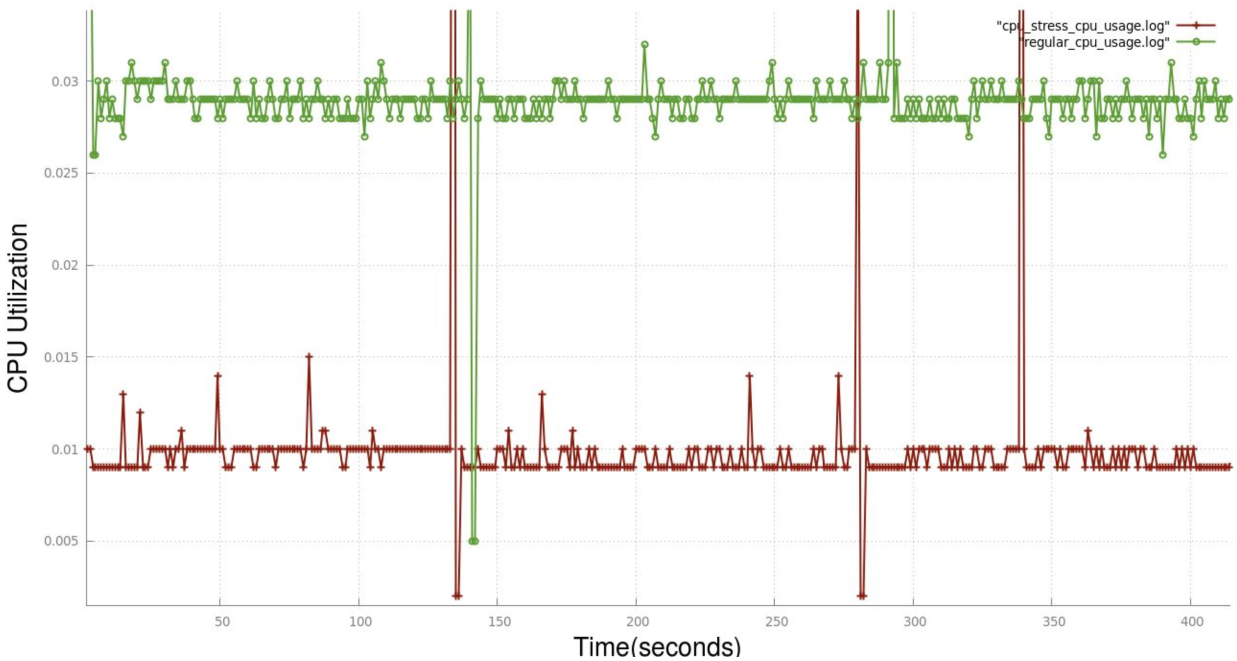
Red - CPU Intensive Green - Regular



In the starting half the CPU utilization in regular case is consistently higher as compared to CPU intensive case.Here eved when in the latter half when Linux DC++ is idel the CPU utilization in regular case is higher as compared to CPU intensive case.

Deductions

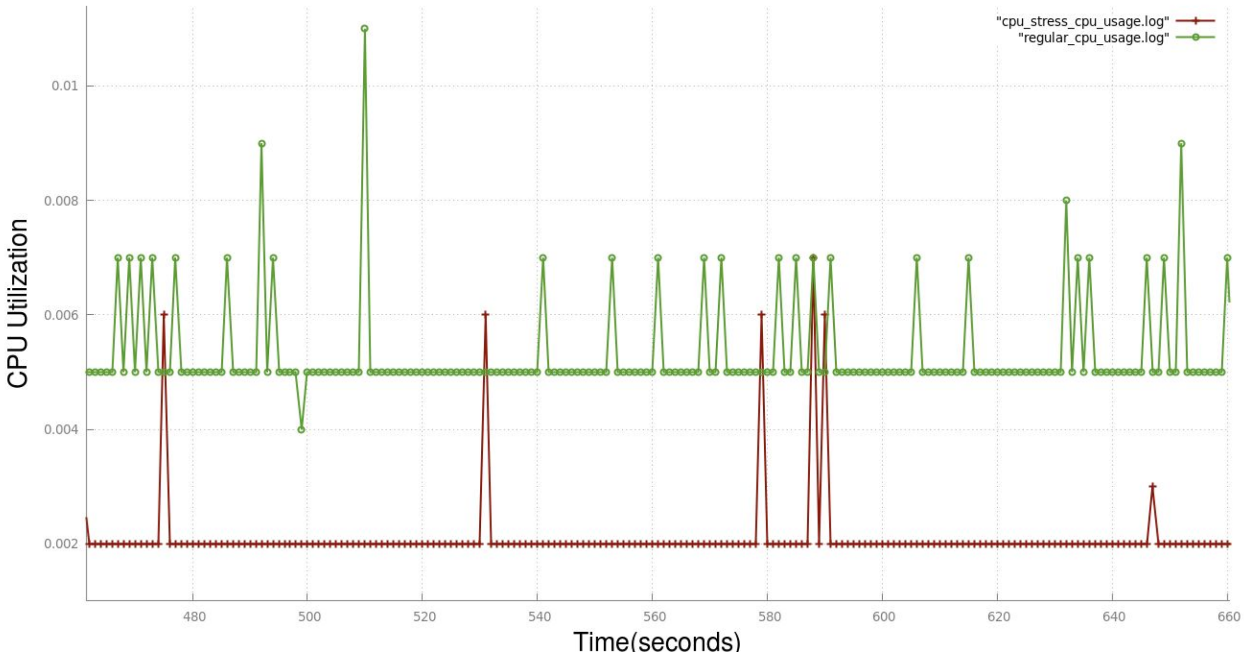
Graph zoomed showing CPU Utilizations when some file is being downloaded or uploaded



In the former half the CPU utilization in regular case is much higher s compared to CPU intensive.It is because in the CPU intensive case due to 40 extra CPU intensive process the CPU time is divided into those 40 process' + Linux DC ++ resulting in a lesser overall CPU utilization.

In the latter case half also the CPU utilization is higher because of the same fact.

Graph zoomed showing CPU Utilizations when linux DC ++ is idle



Regular environment vs I/O intensive environment

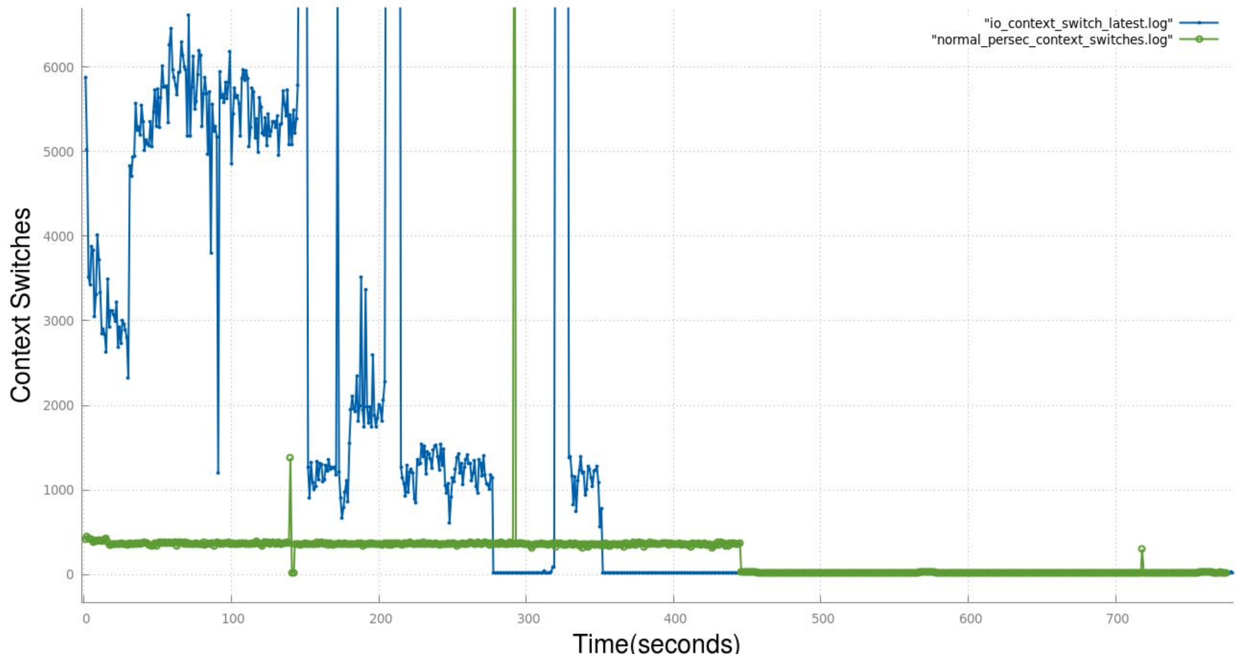
Context Switches

In the starting half of the time period during which a file is download/uploaded using Linux DC ++ the number of context switches in I/O intensive environments are abnormally high as compared to regular environment.

In the latter half during which Linux DC ++ is idle and no file is downloaded/uploaded using it the number of context switches in both the environments are more or less same.

Before deducing anything from the generated data we must first make a note as to how I/O load is generated on the system.The *stress* utility which is used to generate I/O load uses the sync() system call to repeatedly copy buffered data in memory onto the disk.We are using 20 process which are running their own instances of sync() so as to generate I/O intensive environment.

Blue - I/O Intensive Green - Regular

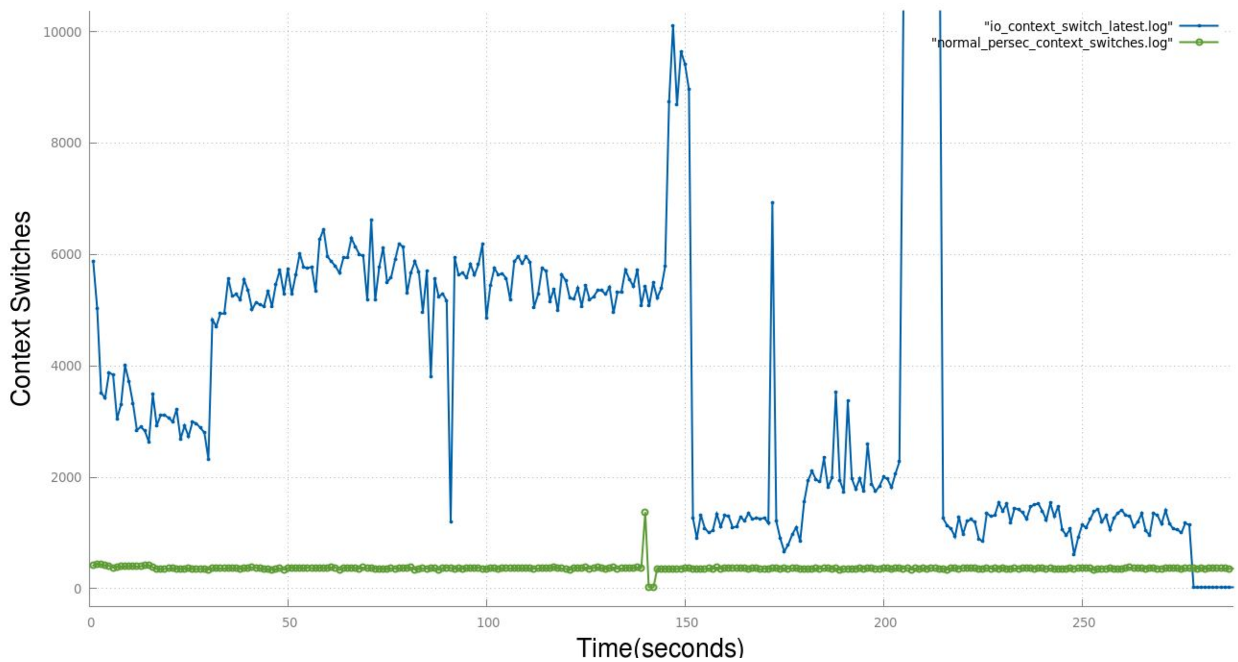


Deductions

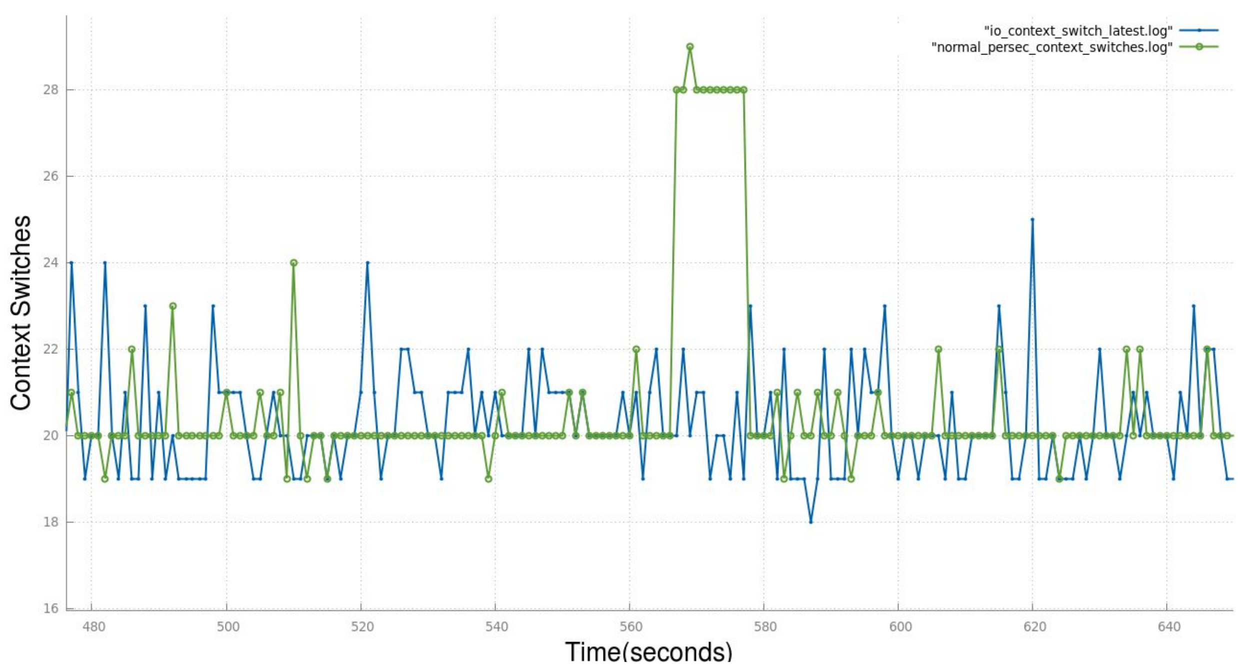
In the earlier half,during which a file is downloaded/uploaded the number of context switches in I/O intensive case is approximately 10 times higher as compared to regular case. One of actions due to which a context switch is triggered is when a process makes itself unrunnable due to a memory-disc synchronization operation.Since we have 20 such process which are performing such sync operations the number of context switch operations automatically goes high.

In the latter half when no downloading is happening the result is more or less same because only those many programs of Linux DC ++ are executing which are required to keep the system up and running.Since less programs of Linux Dc ++ are running when it is idle the no. of context switches is lesser as compared to former case.

__Graph zoomed showing context switches while a file is being downloaded/uploaded.__

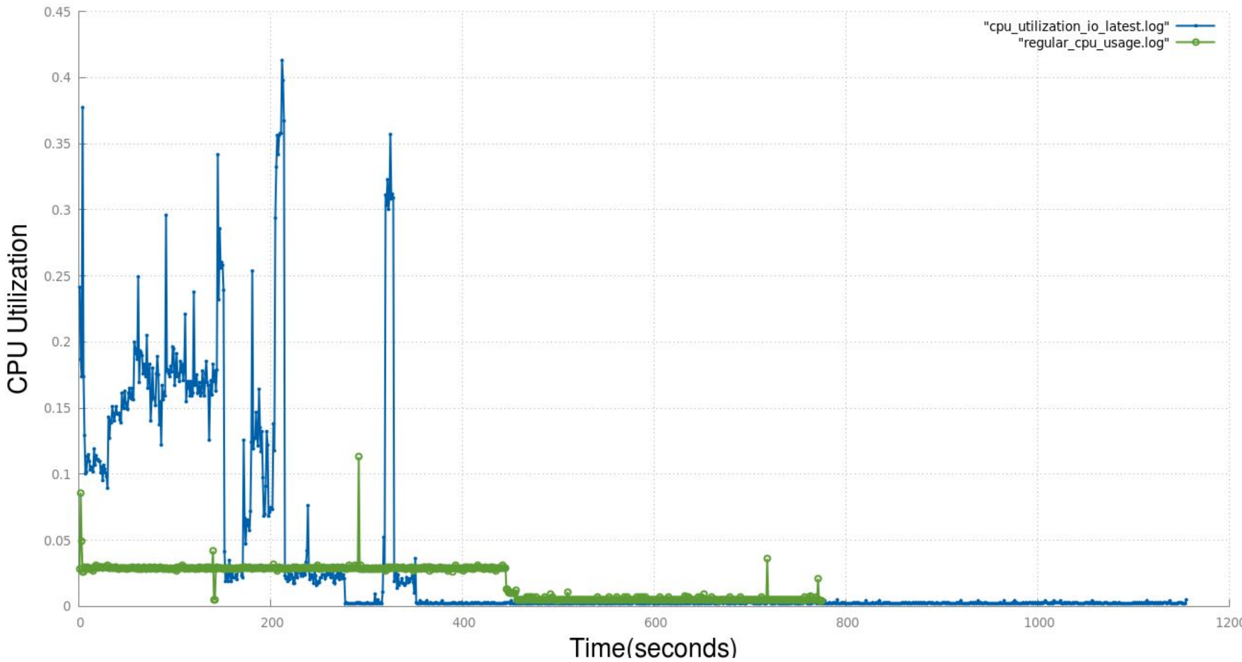


__Graph zoomed showing context switches while Linux DC ++ is idle.__



.

CPU Utilization



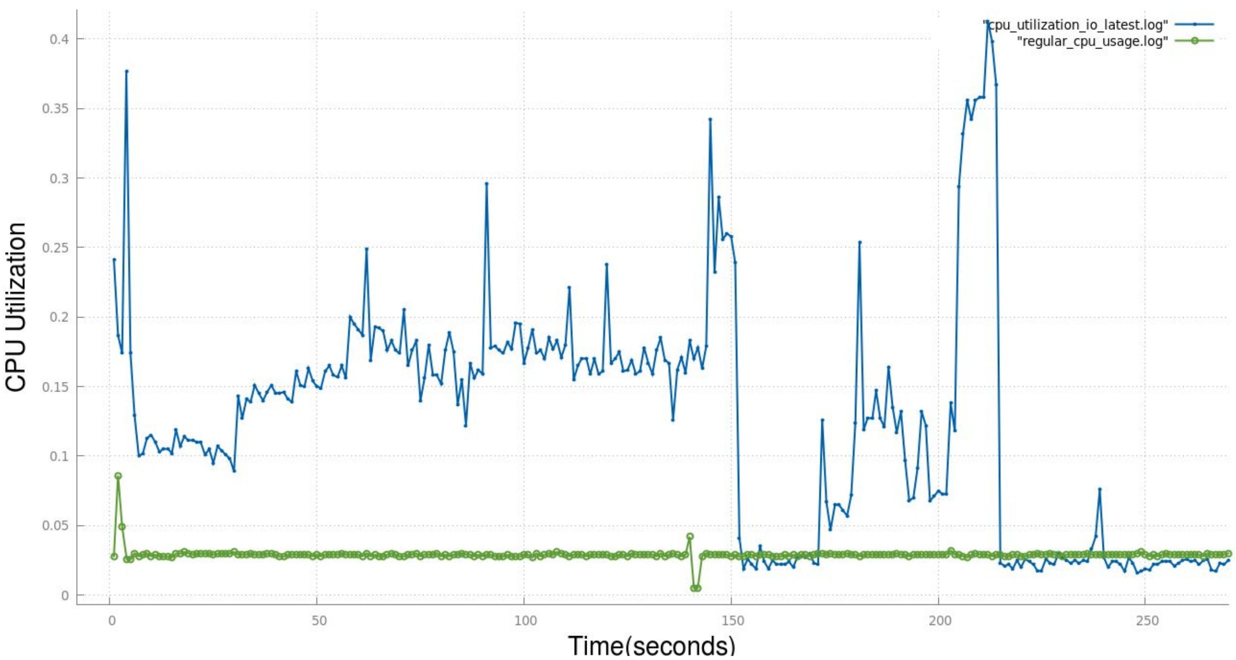
In the earlier half of the time period during which we either upload/download a file through Linux DC++ the CPU utilization is abnormally high in I/O intensive case whereas in the latter case where Linux DC ++ is idle the CPU utilization is very slightly better in regular case.

Deductions

In the earlier half the CPU utilization in I/O intensive case is much higher as compared to regular case because in I/O intensive case the 40 process which are used for increasing the I/O load of the system are not CPU intensive and are dumping buffered data from memory onto disc.Thus most of the CPU time is available to Linux DC ++ resulting in greater CPU utilization.

In the latter case ,as shown in previous cases due to Linux DC ++ being idle it does not use much of the CPU time resulting in a more or less the same result in both the cases.

<u>Graph zoomed showing CPU Utilization while a file is being downloaded/uploaded.</u>



<u>Graph zoomed showing CPU Utilization while Linux DC++ is idle</u>