CS 423 MP3 Group 9

Laurynas Tamulevicius (tamulev2)
Pranava Aditya (paditya2)
Vassil Mladenov (mladeno2)
David Lipowicz (lipowic2)

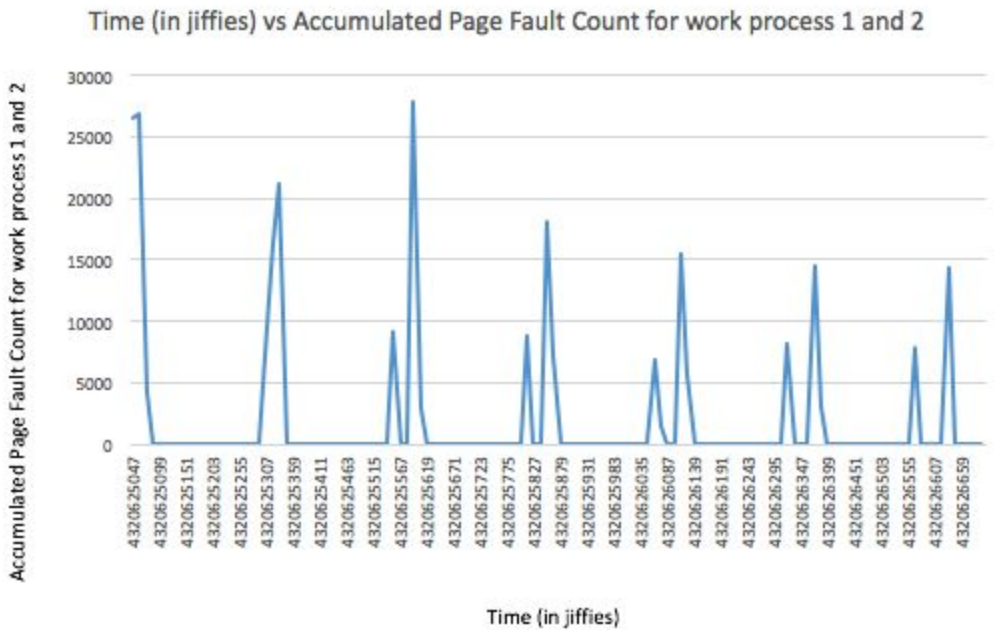
Design and Implementation:

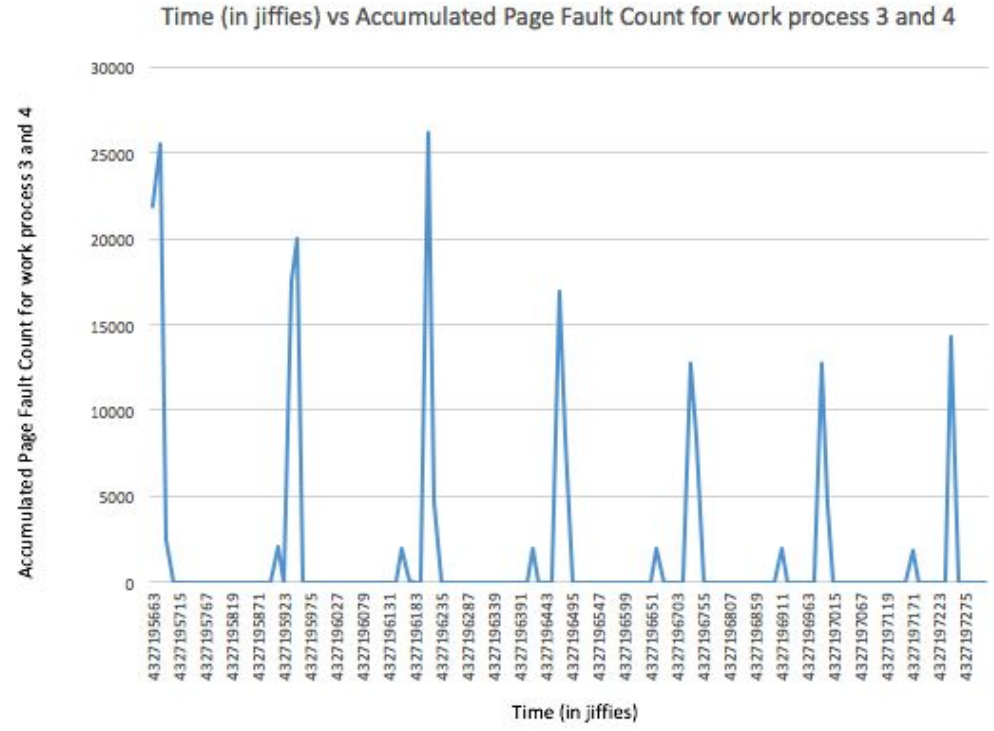
For our design and implementation, we followed the following steps:

1. For this MP, we followed the steps provided in the document very closely. We started off by getting a general kernel module set up, similar to MP1 and MP2. We then worked on adding in the proc filesystem entry (implementing write and read callbacks). This allows PIDs to be registered and unregistered.
2. Next, we created a new PCB containing the major and minor fault counts, along with time spent in CPU (time spent in user space + time spent in kernel space).
3. We followed this by working on the registration and unregistration functions that get called when a PID is registered/unregistered. These functions add to the work queue and remove from the work queue respectively when they are invoked. As well as adding/removing to the work queue, we also add/remove the relevant process in the PCB list.
4. In the next step, we allocated the memory buffer using the vmalloc() function as specified in the MP document. We then set the PG_reserved bits in order to disable management of allocated pages by the virtual memory system.
5. We used a delayed workqueue in order to measure the major page fault counts, the minor page fault counts and the CPU Utilization of all the registered user processes and stored these data samples in the memory buffer. In order to store these samples, we created a defined a new struct that allowed us to store the data so that we could profile the page faults and the CPU Utilization later. In order to get these statistics, we used the provided get_cpu_use function. We sampled the data 20 times per second.
6. We then used a character device driver to allow user-level processes to map the shared memory buffer to their address space. We use the open(), close() and mmap() callbacks of the Linux character device driver in this step. In order to access this device driver from a user process, we create a file in the init function. We then created a node using the steps provided in the MP document.
7. The mmap() callback maps the buffer memory into a virtual address space on the request of a user process. In order to do this, we map the physical pages of the buffer to the virtual address space of a user process. We use two kernel functions for each page. These are the vmalloc_to_pfn(virtual_address) function which gets the physical page address of a virtual page of the buffer and the remap_pfn_range() which maps a virtual page of a user process to a physical page. By doing this, we map the range of address space that is passed in as a parameter at the request of the user process.

Case Study 1: Thrashing and Locality

Graphs:





Analysis:

From the above graphs, we can see that both the graphs have a small peak followed by a large peak, followed by a gap, which was then repeated over and over. However, the first graph has much higher small peaks than the second graph. The small peaks in the first graph are between 5000 and 10000 whereas the small peaks in the second graph are between 0 and 5000. The difference in the graphs is because of the difference in the memory locality of the two programs. In the first case, the instance of work process 1 and the instance of work process 2 use Random Locality Access whereas in the second case, the instance of work process 3 uses Random Locality Access and the instance of work process 4 uses Locality based access. This results in lower small peaks in the second graph. Therefore, due to this difference in locality, the **page fault rate is lesser in the second case**. The completion time for the first case is 1612 jiffies or 16120 milliseconds and the completion time for the second case is 1712 jiffies or 17120 milliseconds. Here we see that the **second case is slightly slower** due to the difference in memory locality.

Data:

Part 1 (work processes 1 and 2):

4321335350 22993 0 11
4321335363 26141 0 13
4321335376 731 0 0
4321335389 0 0 0
4321335402 0 0 0
4321335415 0 0 0
4321335428 0 0 0
4321335441 0 0 0
4321335454 0 0 0
4321335467 0 0 0
4321335480 0 0 0
4321335493 0 0 0
4321335506 0 0 0
4321335519 0 0 0
4321335532 0 0 0
4321335545 0 0 0
4321335558 0 0 0
4321335571 0 0 0
4321335584 0 0 0
4321335597 2017 0 1
4321335610 0 0 0
4321335623 19308 0 10
4321335636 18223 0 8

4321335649 0 0 0
4321335662 0 0 0
4321335675 0 0 0
4321335688 0 0 0
4321335701 0 0 0
4321335714 0 0 0
4321335727 0 0 0
4321335740 0 0 0
4321335753 0 0 0
4321335766 0 0 0
4321335779 0 0 0
4321335792 0 0 0
4321335805 0 0 0
4321335818 0 0 0
4321335831 0 0 0
4321335844 770 0 1
4321335857 1174 0 0
4321335870 0 0 0
4321335883 2279 0 2
4321335896 27097 0 13
4321335909 1503 0 0
4321335922 0 0 0
4321335935 0 0 0
4321335948 0 0 0
4321335961 0 0 0
4321335974 0 0 0
4321335987 0 0 0
4321336000 0 0 0
4321336013 0 0 0
4321336026 0 0 0
4321336039 0 0 0
4321336052 0 0 0
4321336065 0 0 0
4321336078 0 0 0
4321336091 0 0 0
4321336104 1942 0 1
4321336117 0 0 0
4321336130 0 0 0
4321336143 0 0 0
4321336156 19937 0 10
4321336169 5552 0 2
4321336182 0 0 0
4321336195 0 0 0

4321336208 0 0 0
4321336221 0 0 0
4321336234 0 0 0
4321336247 0 0 0
4321336260 0 0 0
4321336273 0 0 0
4321336286 0 0 0
4321336299 0 0 0
4321336312 0 0 0
4321336325 0 0 0
4321336338 0 0 0
4321336351 1951 0 1
4321336364 0 0 0
4321336377 0 0 0
4321336390 0 0 0
4321336403 0 0 0
4321336416 16656 0 8
4321336429 4638 0 2
4321336442 0 0 0
4321336455 0 0 0
4321336468 0 0 0
4321336481 0 0 0
4321336494 0 0 0
4321336507 0 0 0
4321336520 0 0 0
4321336533 0 0 0
4321336546 0 0 0
4321336559 0 0 0
4321336572 0 0 0
4321336585 0 0 0
4321336598 1933 0 1
4321336611 0 0 0
4321336624 0 0 0
4321336637 0 0 0
4321336650 0 0 0
4321336663 0 0 0
4321336676 14962 0 8
4321336689 2612 0 1
4321336702 0 0 0
4321336715 0 0 0
4321336728 0 0 0
4321336741 0 0 0
4321336754 0 0 0

4321336767 0 0 0
4321336780 0 0 0
4321336793 0 0 0
4321336806 0 0 0
4321336819 0 0 0
4321336832 0 0 0
4321336845 0 0 0
4321336858 1864 0 1
4321336871 0 0 0
4321336884 0 0 0
4321336897 0 0 0
4321336910 0 0 0
4321336923 0 0 0
4321336936 14392 0 7
4321336949 0 0 0
4321336962 0 0 0
4321336975 0 0 0
4321336988 0 0 0
4321337001 0 0 0

Part 1 (work processes 3 and 4):

4327195663 21869 0 11
4327195676 25544 0 13
4327195689 2454 0 1
4327195702 0 0 0
4327195715 0 0 0
4327195728 0 0 0
4327195741 0 0 0
4327195754 0 0 0
4327195767 0 0 0
4327195780 0 0 0
4327195793 0 0 0
4327195806 0 0 0
4327195819 0 0 0
4327195832 0 0 0
4327195845 0 0 0
4327195858 0 0 0
4327195871 0 0 0
4327195884 0 0 0
4327195897 0 0 0
4327195910 2017 0 2
4327195923 0 0 0

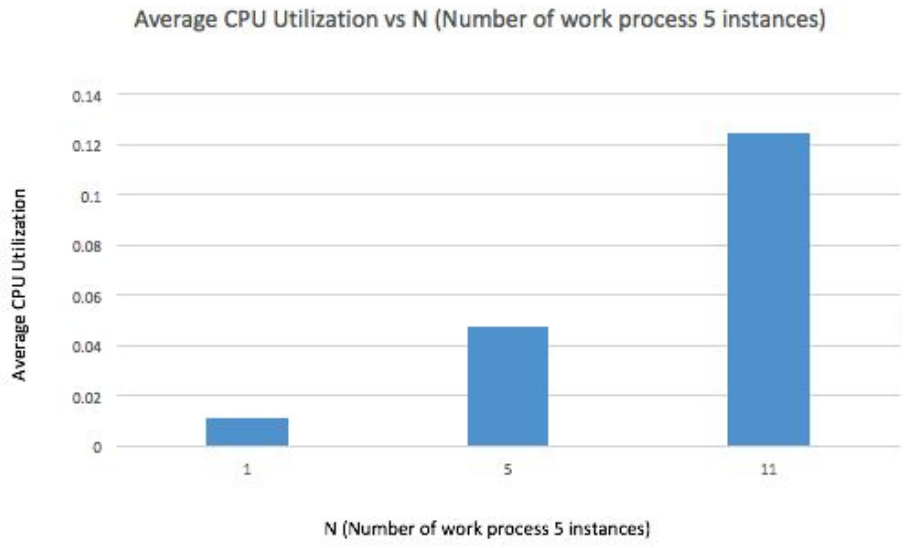
4327195936 17478 0 9
4327195949 20053 0 10
4327195962 0 0 0
4327195975 0 0 0
4327195988 0 0 0
4327196001 0 0 0
4327196014 0 0 0
4327196027 0 0 0
4327196040 0 0 0
4327196053 0 0 0
4327196066 0 0 0
4327196079 0 0 0
4327196092 0 0 0
4327196105 0 0 0
4327196118 0 0 0
4327196131 0 0 0
4327196144 0 0 0
4327196157 1908 0 1
4327196170 36 0 0
4327196183 0 0 0
4327196196 0 0 0
4327196209 26141 0 13
4327196222 4738 0 2
4327196235 0 0 0
4327196248 0 0 0
4327196261 0 0 0
4327196274 0 0 0
4327196287 0 0 0
4327196300 0 0 0
4327196313 0 0 0
4327196326 0 0 0
4327196339 0 0 0
4327196352 0 0 0
4327196365 0 0 0
4327196378 0 0 0
4327196391 0 0 0
4327196404 0 0 0
4327196417 1942 0 1
4327196430 0 0 0
4327196443 0 0 0
4327196456 0 0 0
4327196469 16891 0 8
4327196482 8598 0 3

4327196495 0 0 0
4327196508 0 0 0
4327196521 0 0 0
4327196534 0 0 0
4327196547 0 0 0
4327196560 0 0 0
4327196573 0 0 0
4327196586 0 0 0
4327196599 0 0 0
4327196612 0 0 0
4327196625 0 0 0
4327196638 0 0 0
4327196651 0 0 0
4327196664 1951 0 0
4327196677 0 0 0
4327196690 0 0 0
4327196703 0 0 0
4327196716 0 0 0
4327196729 12738 0 7
4327196742 8404 0 3
4327196755 0 0 0
4327196768 0 0 0
4327196781 0 0 0
4327196794 0 0 0
4327196807 0 0 0
4327196820 0 0 0
4327196833 0 0 0
4327196846 0 0 0
4327196859 0 0 0
4327196872 0 0 0
4327196885 0 0 0
4327196898 0 0 0
4327196911 1933 0 1
4327196924 0 0 0
4327196937 0 0 0
4327196950 0 0 0
4327196963 0 0 0
4327196976 0 0 0
4327196989 12785 0 7
4327197002 4647 0 2
4327197015 0 0 0
4327197028 0 0 0
4327197041 0 0 0

4327197054 0 0 0
4327197067 0 0 0
4327197080 0 0 0
4327197093 0 0 0
4327197106 0 0 0
4327197119 0 0 0
4327197132 0 0 0
4327197145 0 0 0
4327197158 0 0 0
4327197171 1864 0 1
4327197184 0 0 0
4327197197 0 0 0
4327197210 0 0 0
4327197223 0 0 0
4327197236 0 0 0
4327197249 14281 0 7
4327197262 0 0 0
4327197275 0 0 0
4327197288 0 0 0
4327197301 0 0 0
4327197314 0 0 0

Case Study 2: Multiprogramming

Graph:



Analysis:

We can see from the graph that the average CPU Utilization is higher when the number of instances are higher. This behavior is expected because as the number of processes running at the same time increases, so does the CPU utilization. This is mainly due to an increase in processing that needs to be done, and is caused by thrashing. As processes are constantly switched between, there will be a higher rate of thrashing as the number of processes increases, hence the graph looks like it does. When N = 1, the average CPU Utilization time is 0.01150818, when N = 5, the average CPU Utilization is 0.04724409 and when N = 11, the average CPU Utilization is 0.12477286. The data is in accordance with the expected behavior and we can see that **both the CPU Utilization the completion time increase as the number of instances of work process 5 increase.**

Data:

For N = 1:

4337977759 9445 0 4
4337977772 0 0 0
4337977785 0 0 0
4337977798 0 0 0
4337977811 0 0 0
4337977824 0 0 0
4337977837 0 0 0
4337977850 0 0 0
4337977863 0 0 0
4337977876 0 0 0
4337977889 0 0 0
4337977902 0 0 0
4337977915 0 0 0
4337977928 0 0 0
4337977941 0 0 0
4337977954 0 0 0
4337977967 0 0 0
4337977980 0 0 0
4337977993 0 0 0
4337978006 7424 0 3
4337978019 0 0 0
4337978032 0 0 0
4337978045 0 0 0
4337978058 0 0 0
4337978071 0 0 0
4337978084 0 0 0

4337978097 0 0 0
4337978110 0 0 0
4337978123 0 0 0
4337978136 0 0 0
4337978149 0 0 0
4337978162 0 0 0
4337978175 0 0 0
4337978188 0 0 0
4337978201 0 0 0
4337978214 0 0 0
4337978227 0 0 0
4337978240 0 0 0
4337978253 0 0 0
4337978266 6047 0 3
4337978279 0 0 0
4337978292 0 0 0
4337978305 0 0 0
4337978318 0 0 0
4337978331 0 0 0
4337978344 0 0 0
4337978357 0 0 0
4337978370 0 0 0
4337978383 0 0 0
4337978396 0 0 0
4337978409 0 0 0
4337978422 0 0 0
4337978435 0 0 0
4337978448 0 0 0
4337978461 0 0 0
4337978474 0 0 0
4337978487 0 0 0
4337978500 0 0 0
4337978513 5055 0 3
4337978526 0 0 0
4337978539 0 0 0
4337978552 0 0 0
4337978565 0 0 0
4337978578 0 0 0
4337978591 0 0 0
4337978604 0 0 0
4337978617 0 0 0
4337978630 0 0 0
4337978643 0 0 0

4337978656 0 0 0
4337978669 0 0 0
4337978682 0 0 0
4337978695 0 0 0
4337978708 0 0 0
4337978721 0 0 0
4337978734 0 0 0
4337978747 0 0 0
4337978760 940 0 1
4337978773 3160 0 1
4337978786 0 0 0
4337978799 0 0 0
4337978812 0 0 0
4337978825 0 0 0
4337978838 0 0 0
4337978851 0 0 0
4337978864 0 0 0
4337978877 0 0 0
4337978890 0 0 0
4337978903 0 0 0
4337978916 0 0 0
4337978929 0 0 0
4337978942 0 0 0
4337978955 0 0 0
4337978968 0 0 0
4337978981 0 0 0
4337978994 0 0 0
4337979007 0 0 0
4337979020 3438 0 2
4337979033 0 0 0
4337979046 0 0 0
4337979059 0 0 0
4337979072 0 0 0
4337979085 0 0 0
4337979098 0 0 0
4337979111 0 0 0
4337979124 0 0 0
4337979137 0 0 0
4337979150 0 0 0
4337979163 0 0 0
4337979176 0 0 0
4337979189 0 0 0
4337979202 0 0 0

4337979215 0 0 0
4337979228 0 0 0
4337979241 0 0 0
4337979254 0 0 0
4337979267 2807 0 2
4337979280 0 0 0
4337979293 0 0 0
4337979306 0 0 0
4337979319 0 0 0
4337979332 0 0 0
4337979345 0 0 0
4337979358 0 0 0
4337979371 0 0 0
4337979384 0 0 0
4337979397 0 0 0
4337979410 0 0 0

For N = 5:

4338119835 30850 1 13
4338119848 16373 0 6
4338119861 0 0 0
4338119874 0 0 0
4338119887 0 0 0
4338119900 0 0 0
4338119913 0 0 0
4338119926 0 0 0
4338119939 0 0 0
4338119952 0 0 0
4338119965 0 0 0
4338119978 0 0 0
4338119991 0 0 0
4338120004 0 0 0
4338120017 0 0 0
4338120030 0 0 0
4338120043 0 0 0
4338120056 0 0 0
4338120069 0 0 0
4338120082 0 0 0
4338120095 20760 0 9
4338120108 16360 0 6
4338120121 0 0 0
4338120134 0 0 0

4338120147 0 0 0
4338120160 0 0 0
4338120173 0 0 0
4338120186 0 0 0
4338120199 0 0 0
4338120212 0 0 0
4338120225 0 0 0
4338120238 0 0 0
4338120251 0 0 0
4338120264 0 0 0
4338120277 0 0 0
4338120290 0 0 0
4338120303 0 0 0
4338120316 0 0 0
4338120329 0 0 0
4338120342 6047 0 2
4338120355 16119 0 7
4338120368 8069 0 3
4338120381 0 0 0
4338120394 0 0 0
4338120407 0 0 0
4338120420 0 0 0
4338120433 0 0 0
4338120446 0 0 0
4338120459 0 0 0
4338120472 0 0 0
4338120485 0 0 0
4338120498 0 0 0
4338120511 0 0 0
4338120524 0 0 0
4338120537 0 0 0
4338120550 0 0 0
4338120563 0 0 0
4338120576 0 0 0
4338120589 3146 0 2
4338120602 1909 0 0
4338120615 20220 0 8
4338120628 0 0 0
4338120641 0 0 0
4338120654 0 0 0
4338120667 0 0 0
4338120680 0 0 0
4338120693 0 0 0

4338120706 0 0 0
4338120719 0 0 0
4338120732 0 0 0
4338120745 0 0 0
4338120758 0 0 0
4338120771 0 0 0
4338120784 0 0 0
4338120797 0 0 0
4338120810 0 0 0
4338120823 0 0 0
4338120836 0 0 0
4338120849 4100 0 2
4338120862 4754 0 2
4338120875 11646 0 4
4338120888 0 0 0
4338120901 0 0 0
4338120914 0 0 0
4338120927 0 0 0
4338120940 0 0 0
4338120953 0 0 0
4338120966 0 0 0
4338120979 0 0 0
4338120992 0 0 0
4338121005 0 0 0
4338121018 0 0 0
4338121031 0 0 0
4338121044 0 0 0
4338121057 0 0 0
4338121070 0 0 0
4338121083 0 0 0
4338121096 3438 0 1
4338121109 0 0 0
4338121122 13752 0 6
4338121135 0 0 0
4338121148 0 0 0
4338121161 0 0 0
4338121174 0 0 0
4338121187 0 0 0
4338121200 0 0 0
4338121213 0 0 0
4338121226 0 0 0
4338121239 0 0 0
4338121252 0 0 0

4338121265 0 0 0
4338121278 0 0 0
4338121291 0 0 0
4338121304 0 0 0
4338121317 0 0 0
4338121330 0 0 0
4338121343 221 0 1
4338121356 2586 0 1
4338121369 5614 0 2
4338121382 5614 0 3
4338121395 0 0 0
4338121408 0 0 0
4338121421 0 0 0
4338121434 0 0 0
4338121447 0 0 0
4338121460 0 0 0
4338121473 0 0 0
4338121486 0 0 0

For N = 11:

4338231453 20663 0 14
4338231466 25502 0 13
4338231479 26773 0 13
4338231492 27483 0 13
4338231505 3510 0 1
4338231518 0 0 0
4338231531 0 0 0
4338231544 0 0 0
4338231557 0 0 0
4338231570 0 0 0
4338231583 0 0 0
4338231596 0 0 0
4338231609 0 0 0
4338231622 0 0 0
4338231635 0 0 0
4338231648 0 0 0
4338231661 0 0 0
4338231674 0 0 0
4338231687 0 0 0
4338231700 0 0 0
4338231713 0 0 0
4338231726 7424 0 3

4338231739 0 0 0
4338231752 25137 0 13
4338231765 28635 0 13
4338231778 20468 0 9
4338231791 0 0 0
4338231804 0 0 0
4338231817 0 0 0
4338231830 0 0 0
4338231843 0 0 0
4338231856 0 0 0
4338231869 0 0 0
4338231882 0 0 0
4338231895 0 0 0
4338231908 0 0 0
4338231921 0 0 0
4338231934 0 0 0
4338231947 0 0 0
4338231960 0 0 0
4338231973 0 0 0
4338231986 6047 0 3
4338231999 0 0 0
4338232012 0 0 0
4338232025 15521 0 7
4338232038 31084 0 13
4338232051 13865 0 5
4338232064 0 0 0
4338232077 0 0 0
4338232090 0 0 0
4338232103 0 0 0
4338232116 0 0 0
4338232129 0 0 0
4338232142 0 0 0
4338232155 0 0 0
4338232168 0 0 0
4338232181 0 0 0
4338232194 0 0 0
4338232207 0 0 0
4338232220 0 0 0
4338232233 5055 0 2
4338232246 0 0 0
4338232259 0 0 0
4338232272 0 0 0
4338232285 8702 0 5

4338232298 17924 0 6
4338232311 21978 0 7
4338232324 1946 0 0
4338232337 0 0 0
4338232350 0 0 0
4338232363 0 0 0
4338232376 0 0 0
4338232389 0 0 0
4338232402 0 0 0
4338232415 0 0 0
4338232428 0 0 0
4338232441 0 0 0
4338232454 0 0 0
4338232467 0 0 0
4338232480 0 0 0
4338232493 4100 0 1
4338232506 0 0 0
4338232519 0 0 0
4338232532 0 0 0
4338232545 8200 0 4
4338232558 4100 0 1
4338232571 16370 0 8
4338232584 10219 7 5
4338232597 2134 158 1
4338232610 0 0 0
4338232623 0 0 0
4338232636 0 0 0
4338232649 0 0 0
4338232662 0 0 0
4338232675 0 0 0
4338232688 0 0 0
4338232701 0 0 0
4338232714 0 0 0
4338232727 0 0 0
4338232740 3438 0 1
4338232753 0 0 0
4338232766 0 0 0
4338232779 0 0 0
4338232792 4233 326 1
4338232805 3712 271 3
4338232818 2499 251 1
4338232831 3650 325 6
4338232844 8093 854 6

4338232857 5330 854 6
4338232870 4015 812 3
4338232883 3248 715 2
4338232896 0 0 0
4338232909 0 0 0
4338232922 0 0 0
4338232935 0 0 0
4338232948 0 0 0
4338232961 0 0 0
4338232974 0 0 0
4338232987 324 172 0
4338233000 1570 760 3
4338233013 778 387 2
4338233026 135 57 0
4338233039 0 0 0
4338233052 451 143 1
4338233065 2424 739 3
4338233078 4024 1263 4
4338233091 2037 648 2
4338233104 2127 683 2

For the graph:

N	Average CPU Utilization (over entire time period)
1	0.01150818
5	0.04724409
11	0.12477286