Assignment 3 Write Up

Daniel Urrutia (deurruti@ucsc.edu)
Justin Wong (jujwong@ucsc.edu)
Joshua Mora (jommora@ucsc.edu)
CMPS 111, Winter 2016

For testing, we will be making use of the VMSTAX utility for graphing the number of page faults and total paging activity in a given test with the stress FreeBSD package. We assigned 256MB of RAM to the FreeBSD VM.
The stress package is installed on FreeBSD by running the command

pkg install stress

To run the stress program after installing the package, you run the command

stress -m 3 –vm-bytes [number]MB -t [time]s

The -m flag specifies the number of processes that will be allocating memory during the stress test. The –vm-bytes flag specifies the amount (in megabytes) of memory assigned to each of those memory allocating processes. The -t flag specifies the length of time to run the stress test for (in seconds).

IMPORTANT NOTE: Our charts that display the mod. kernel's page faults and paging activity have incorrect time scales (labeled on their respective x-axis). The correct time scale begins at 0 seconds. Gen. kernel charts and mod. kernel charts are graphed on the same time scales (so please look at generic kernel time scales), it's just labeled incorrectly on the mod. kernel charts.

Stress Test Logs

Figure 1: Generic kern stress @ 32MB

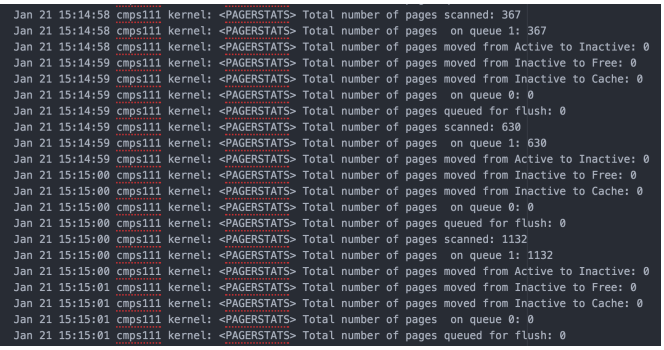

Figure 2: Generic kern stress @ 64MB

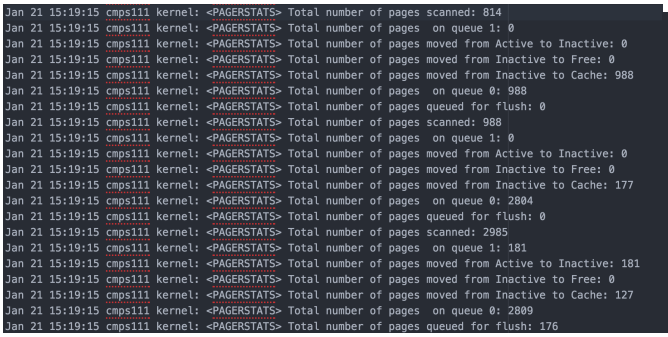

Figure 3: Generic kern stress @ 128MB

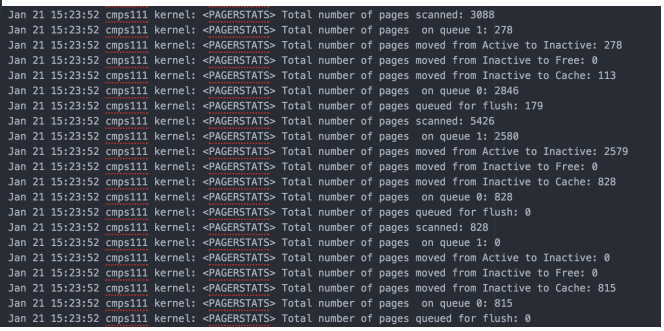

Figure 4: Generic kern stress @ 256MB

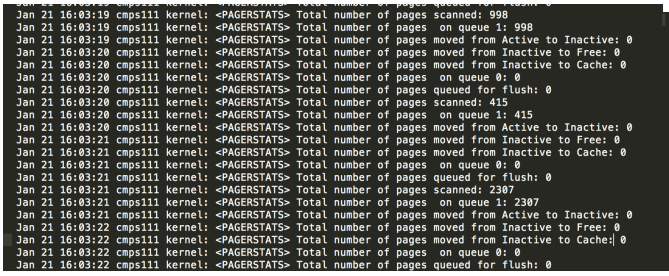

Figure 5: Modified kern stress @ 32MB

Figure 6: Modified kern stress @ 64MB

Figure 7: Modified kern stress @ 128MB

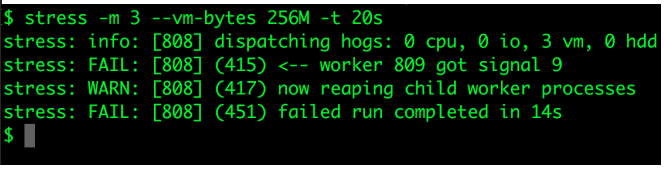

Figure 8: Modified kern stress @ 256MB

Page faults and activity

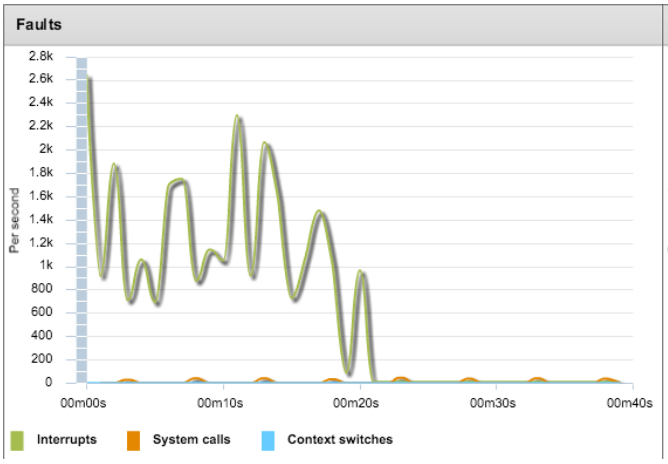

Figure 9: Number of page faults 32MB stress (gen.)

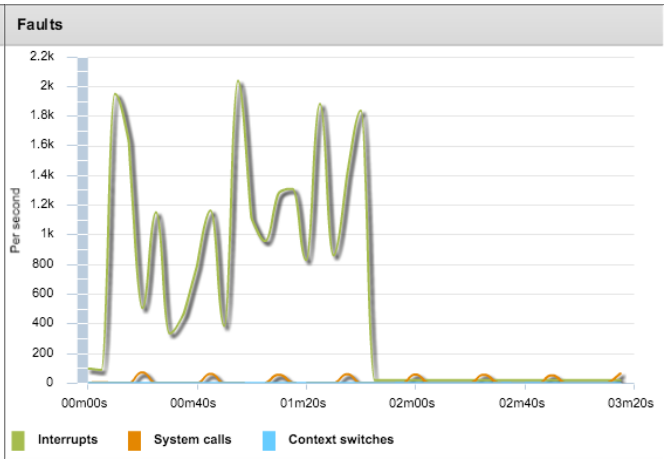

Figure 10: Number of page faults 32MB stress (mod.)

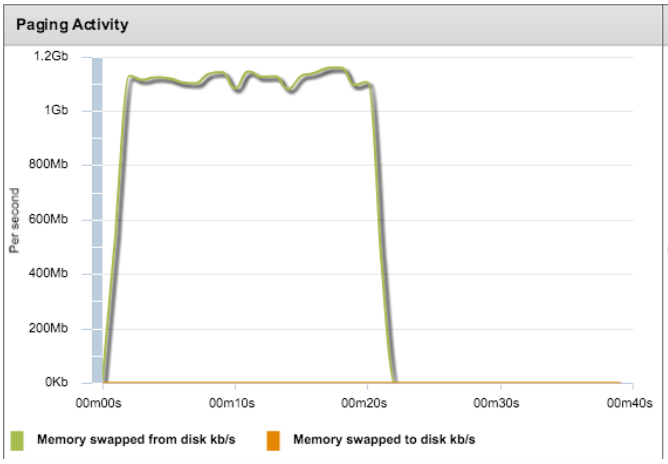

Figure 11: Paging activity 32MB stress (gen.)

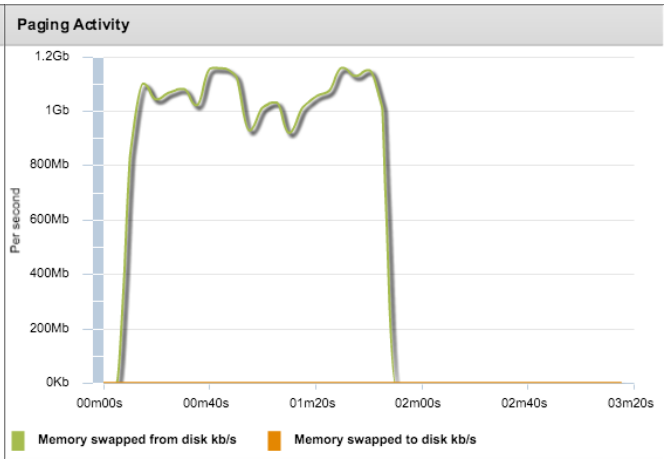

Figure 12: Paging activity 32MB stress (mod.)

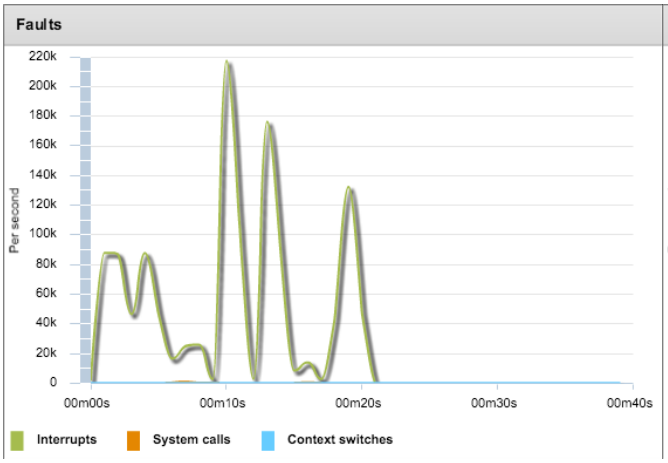

Figure 13: Number of page faults 64MB stress (gen.)

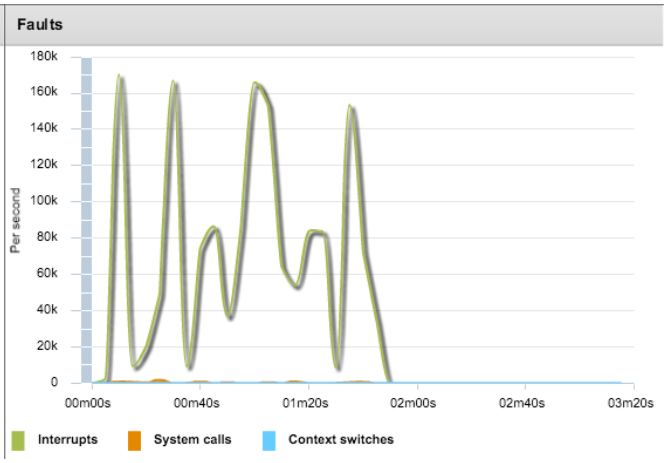

Figure 14: Number of page faults 64MB stress (mod.)

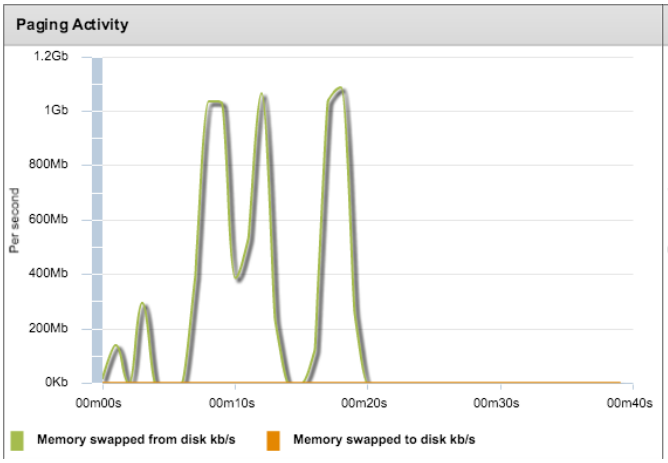

Figure 15: Paging activity 64MB stress (gen.)

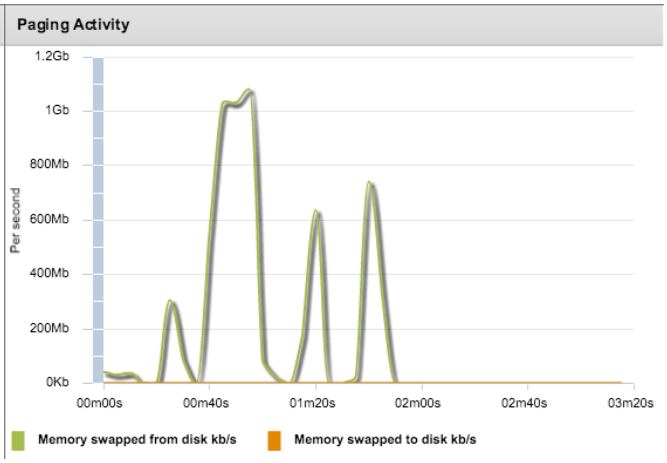

Figure 16: Paging activity 64MB stress (mod.)

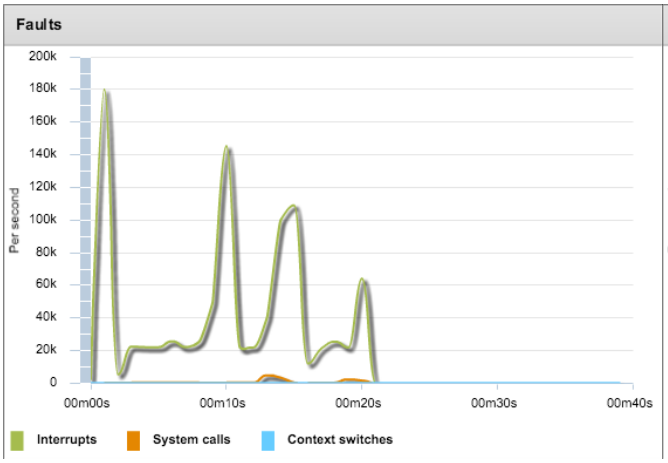

Figure 17: Number of page faults 128MB (gen.)

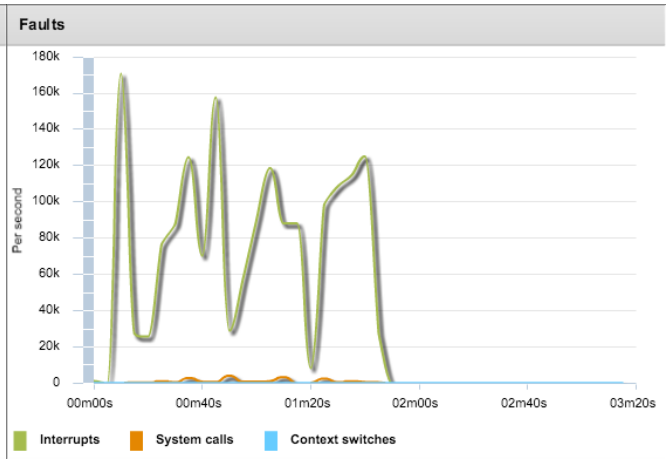

Figure 18: Number of page faults 128MB (mod.)

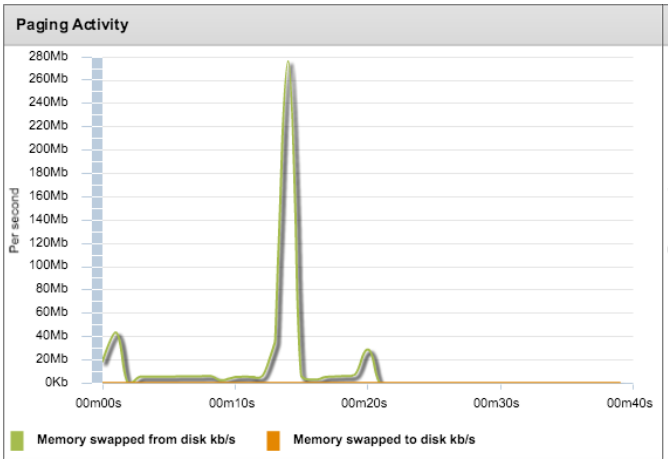

Figure 19: Paging activity 128MB (gen.)

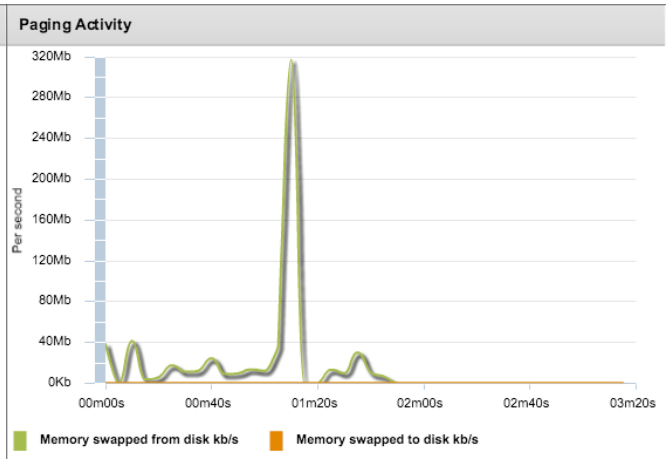

Figure 20: Paging activity 128MB (mod.)

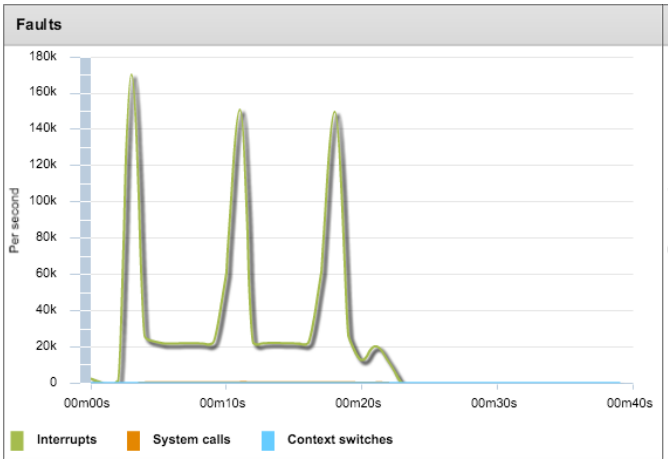

Figure 21: Number of page faults 256MB (gen.)

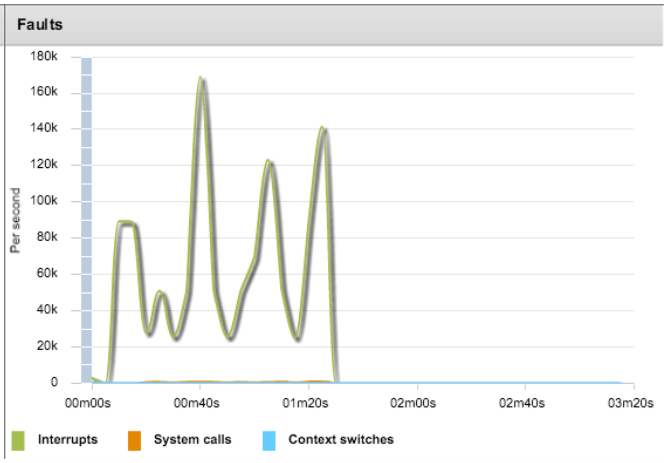

Figure 22: Paging activity 256MB (gen.)

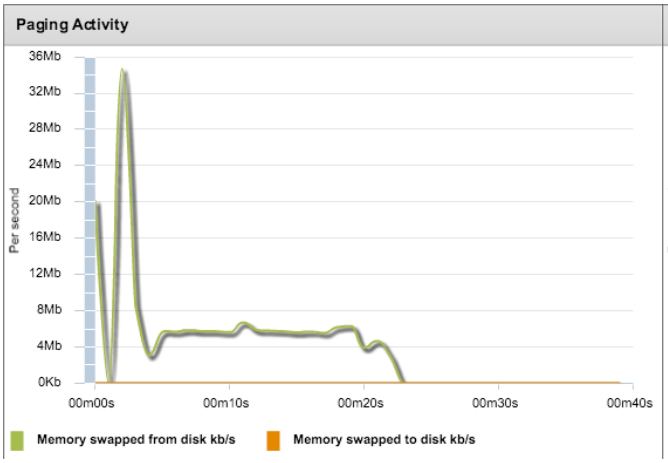

Figure 23: Paging activity 256MB (gen.)

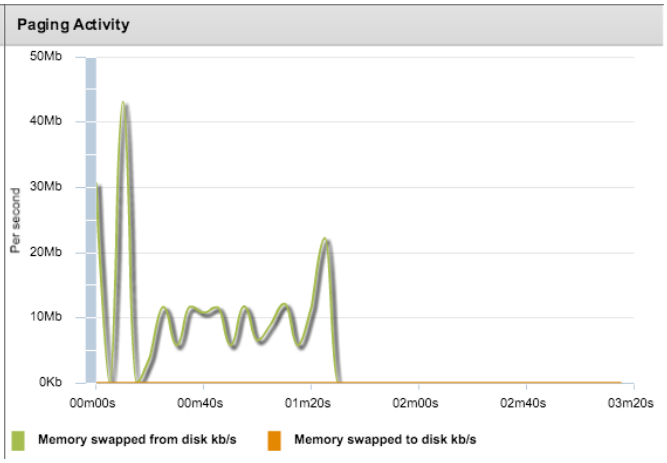

Figure 24: Paging activity 256MB (mod.)

32MB and 64MB Stress Testing

32MB (figures 9-12): In the beginning of the 32MB stress test, the generic kernel had more page faults than the modified one. However, near the end of the stress test, the modified kernel had approximately more page faults than the generic kernel. Between the two kernels, paging activity was about the same with the exception of a noticeable dip in the modified kernel's paging activity graph (figure 12). This means that overall, the modified kernel had slightly less paging activity.

64MB (figures 13-16): In the beginning of the 64MB stress test, the modified kernel had about double the number of page faults and peaked more often. There was one peak in the generic kernel's graph that had a higher value than any of the peaks in the modified kernel's graph, but when viewing the overall state of the modified kernel, the frequency and value of peaks leads to a greater sum of page faults in the modified kernel when compared to the generic kernel's graph. Interestingly enough, there was more paging activity in the generic kernel than in the modified kernel, despite there being more page faults in the modified kernel. The generic kernel had 3 main peaks which were of consistent height; the modified kernel also had 3 main peaks, but the later two peaks in the modified kernel's graph were significantly lower than the first peak. This is why there was less overall paging activity in the modified kernel.

128MB and 256MB Stress Testing

128MB (figures 17-20): Both kernels had beginning peaks which were approximately the same value (about 180,000), but the modified kernel had significantly more peaks. This means that the modified kernel had more page faults overall, even though the modified kernel's y-axis is on a smaller scale. In the paging activity graphs, the generic and modified kernels had similar distributions (both had a single main peak), but the generic kernel had a smoother distribution in the regions outside of the peak. The modified kernel was a bit more sporadic in the regions outside of the single peak. One other thing worth mentioning is that the modified kernel's y-axis is on a higher scale, so its single main peak has a much higher value than the generic kernel's single main peak.

256MB (figures 21-24): The number of page faults in the modified kernel had a more sporadic distribution whereas the generic kernel had fewer spikes and also the regions between the spikes plateaued. This means the generic kernel ran a bit more "smoothly," so it was unnecessary for it to page as often, leading to fewer page faults. The behavior between the two kernels for the page fault graphs is mirrored in the page activity. The modified kernel had a more sporadic distribution of paging activity throughout its entire stress test whereas the generic kernel ran much "smoother". The peak in the generic kernel's graph is significantly less than the peak in the modified kernel's graph. If you pay attention to the y-axes for the two graphs, you will see that the modified kernel has higher overall paging activity. This means that the modified kernel had to page more often.

Summary:

After performing analysis on the several configurations of memory that we provided to the stress test, our findings lead us to conclude that the modified kernel does not scale for larger memory configurations. However, we see that in the case of 32 and 64 megabytes of memory, the modified and default system have very similar performance. In some cases, the modified kernel's paging system was more optimal, specifically when comparing figure 13 to figure 14. Overall, our findings prove that slim-chance paging leads to worse performance when compared to the default kernel's paging system.

As a side note, as you can see in figure 8, modified kernel failed when specifying 256M for the stress test. This is why we provided the image of the failure rather than the logs.