CSE 542 Operating Systems Home Work Project

Performance turning of a differentiated QoS Web Service

HWP1 due: Sep 18, 11:00 AM   HWP2 due: Tues, Oct 16, 11:00AM

Late submissions will not be accepted

Our primary goal for this project is to experience the innards of operating systems. We will implement some of the techniques that we will read in the research papers. We will also practice on how to write "good" technical papers.

For the homework projects, we will tune the performance of a web service that wishes to provide differentiated Quality of Service using the technologies that we will learn in this class. The web service will run on a single machine. Users can access the web service either on ports XX or YY (choose the ports that are convenient for you). Users accessing the service through port XX expect "better" service than users accessing the service at port YY. Presumably, users pay differently depending on the level of service.

You will choose the necessary machines, operating systems, web services, benchmarks and other tools. I will provide two machines, niue.cse.nd.edu and tahiti.cse.nd.edu for experimentation. These are 1.8 GHz Pentium machines with 256 MB of main memory connected by dual NICs with a dedicated Gigabit link. These machines run FreeBSD 5.1_CURRENT. They can be dual-booted to RedHat Linux 9.0. You will also have access to a raw partition to install any "optimized" file system that you choose. Since you will modify the kernel for the second project, you have to use an operating that you have access to the kernel source code. I think FreeBSD is a great operating system for our purposes. The recommended book (I have a copy that you can borrow) should give you information about FreeBSD. You are also free to use your own machines for the experiments; choose machines that are not "too fast" so that it is easier to show improvements. You can install two copies of the same web server – say apache – and configure it to run on different ports. You can use the second machine to drive your workload/benchmark.

Home work project 1: Performance tuning

For this project, you will analyze the base system and then choose three unique optimizations that you think will make your system perform "better" (of course, you will define what better means for your system). The optimizations need not modify anything within the kernel, they can be simple tweaks that you can do to improve system performance. For example, you may change process priorities, change VM parameters, tweaks kernel parameters using sysctl etc.

Home work project 2: Kernel hacking

For this project, you will implement three techniques that we learnt in class (e.g. lottery scheduling) in the kernel in order to improve your system.

Submission Guidelines

You will report on your experience with the original system and the optimizations – why you chose them, whether they worked as expected etc. Your reports should be in PDF (adobe distiller is installed in wizard.cse.nd.edu to convert postscript documents to PDF) and anonymous for double blind reviewing. Your reports should be self contained (you have to motivate the problem and describe everything "important about your system" and is limited to 6 papers in either ACM (http://www.acm.org/sigs/pubs/proceed/template.html), USENIX (http://www.usenix.org/events/usenix03/instrux/details.html#specs), IEEE (http://www.ieee.org/portal/index.jsp?pageID=corp_level1&path=pubs/transactions&file=stylesheets.xml&xsl=generic.xsl) formats. Choose the performance graphs wisely so that they prove your point.

Start early. This is a graduate level work, going beyond the call of duty is required! If you want to implement something cool – go for it. I will email paper submission instructions to the class mailing list.

Interesting reading

- **Brittle metrics in operating systems research** Jeffrey C. Mogul In Proc. 7th IEEE Workshop on Hot Topics in Operating Systems (HotOS-VII)
- **NFS Tricks and Benchmarking Traps** Daniel Ellard and Margo Seltzer . In Proceedings of the 2003 Freenix Track, USENIX Technical Conference (http://www.eecs.harvard.edu/~syrah/papers/freenix-03/)