

# PERFORMANCE TUNING

Getting that extra bogomips

# ACTUALLY...

- The focus for this section will be more on the process of performance tuning...
  - Collecting meaningful benchmarks
  - Establishing a baseline
  - Understanding how to compare benchmarks

# BENCHMARKS

- A benchmark is a specific measure of performance, taken in a repeatable fashion such that outside influences are minimized and operational characteristics of the machine and operating system are matched for every measurement.
  - In other words, every time a benchmark is taken, it's taken in the same manner and under the same conditions
  - This allows for meaningful comparison of benchmarks

# BENCHMARK BEST PRACTICES

- Unless the benchmark mandates otherwise, it's generally best to collect in single user mode. This will help to isolate the system from outside users and influence, such as network requests and nosy users.
- Furthermore, if possible, shut down all services that won't be needed ( single user mode will go a long way towards accomplishing this )
- Run the benchmark at least 5 times in a row, and average the results. For better accuracy, run the test 10-20 times in a row and throw out the top and bottom 10% metrics. Then average the resultant set.
- Document everything! Conditions, commands, sequences, timing, every individual result and how the final benchmark was calculated.

# BENCHMARKING TOOLS

# HDPARM

- Great way to test hard drive performance
- Using the -t option of hdparm, disk subsystem read times can be accurately measured without any filesystem overhead or cache inconsistencies.
- The -T option measures cache read performance.

# COMPILING

- One of the most common ‘real world’ benchmarks is to compile some software and time how long it takes. This covers cpu, io, memory and operating system.
- The kernel is a great example
  - Obtain source code for kernel ( must always use same version of kernel for meaningful results )
  - Configure with default configuration
  - `time` the compile step ( `make` )

# LMBENCH

- lmbench is a well-known tool with a large selection of benchmark tests available
- See <http://lmbench.sourceforge.net>



# IOZONE

- A very nice filesystem benchmarking suite.
- Covers many different file and IO system operations.
- Produces excellent reports which can be imported into a spreadsheet applications to create outstanding graphical representations.
- See <http://www.iozone.org>

# SOME TUNABLE FEATURES

Play with caution

# SHUTDOWN UNUSED SERVICES

- Why run apache if you aren't serving a website?
- Review running services and shut down unnecessary ones

# HDPARM

- Lots of parameters available for tuning.
- In-depth knowledge of disk IO subsystems required.
- Very complex command - see man page.

# SYSCTL

- Kernel parameters
- `sysctl -a`: produce a complete list of all tunable kernel parameters
- Examples include networking, kernel, filesystem

# RECOMPILE THE KERNEL

- Custom build the kernel - add/remove the features needed for each particular system.
- Target exact processor family to take advantage of special instructions and abilities

# OTHER TRICKS

- If access times aren't needed, disable them on the filesystem
  - Modify `/etc/fstab` and add "noatime" to options for each filesystem
  - This reduces inode writes every time a directory is visited or a file is viewed
- On multiple CPU/core machines, use `taskset` to bind processes to one processor to help reduce unnecessary cache dumping

# OTHER TRICKS

- Tune physical and virtual memory
  - Spread out swap space across several drives, set priorities to maximize parallelism while avoiding slower drives
- Implement a RAID solution
- Disable SELinux
- Tune nice levels



```
slideshow.end();
```