**init.d tweaker script details**

**Battery tweak**

**dirty\_expire\_centisecs**

This tunable is used to define when dirty data is old enough to be eligible for writeout by the kernel flusher threads. It is expressed in 100'ths of a second. Data which has been dirty in-memory for longer than this interval will be written out next time a flusher thread wakes up.

Default value is 100, that means it clears the data that is no longer used by any process in every 100 second. We change it to 500 for saving power.

Default value: 100

Changed value: 500

**dirty\_writeback\_centisecs**

The kernel flusher threads will periodically wake up and write `old' data out to disk. This tunable expresses the interval between those wakeups, in 100'ths of a second. Setting this to zero disables periodic writeback altogether.

Default value is 100 second that denotes the kernel flasher thread wake up in every 100 second and write the data to secondary memory. We change it to 1000 second.

Default value: 100

Changed value: 1000

**dirty\_background\_ratio**

Contains, as a percentage of total system memory, the number of pages at which the background kernel flusher threads will start writing out dirty data.

Default value: 10

Changed value: 60

**dirty\_ratio**

Contains, as a percentage of total system memory, the number of pages at which a process which is generating disk writes will itself start writing out dirty data.

Default value: 40

Changed value: 95

**Defragment Database**

Defragmentation means rearrange the data of any disk or file. It is actually a indexing technique. We use this script for defragment the SQLite database on every boot. It may take a few second to defragment for the first time.

**Disable Normalize Sleeper**

Normalize Sleeper is a property of kernel which calls the normalize script on every period defined by the script. Normalize sleeper is a scheduler indeed. As we defragment the database on every boot, so we actually don’t need to normalize the database.

Default value: AFFINE\_WAKEUPS/ SYNC\_WAKEUPS

Changed value: NO\_NORMALIZED\_SLEEPER

Other most used values:

* NEW\_FAIR\_SLEEPERS
* NORMALIZED\_SLEEPER
* WAKEUP\_PREEMPT
* START\_DEBIT
* DOUBLE\_TICK
* LAST\_BUDDY

**EXT4 tweak perfect mount options**

Default file system of linux kernel is EXT4. We remount the file system, /data directory and /cache directory with noauto\_da\_alloc (automatic allocation off), nosuid (SU id disable), nodiratime (redirection time off) and barrier=0.

**EXT4 tweak removes journalism**

Journaling is an ext4 filesystem log service and it involves using memory and data.So disabling journaling means that memory and data used for maintaining this service are no longer used and are available for other processes. Journaling is required in computer hard drives as a precaution for data loss in situations like unclean shutdown in power outage. And I don't think mobile phones which use flash drives ever require journaling.

Journaling ensures the integrity of the filesystem by keeping a log of the ongoing disk changes. However, it is known to have a small overhead. Some people with special requirements and workloads can run without a journal and its integrity advantages. In Ext4 the journaling feature can be disabled, which provides a small performance improvement.

We remove journalism in /system, /data and /cache directory.

**CPU Governor Changer**

We already benchmark for the best CPU Governor for battery performance. With this script we change it to the best.

Default value: Ondemand/ Smartass

Changed value: Ondemand

**Increase Cache Size**

Depending on what raid level you have and stripe settings, Linux may be fetching a lot of data for a small random read. It may be helpful to set a larger read-ahead. So we Increase the cache size of the disk.

Default value: 512

Changed value: 2048

**Internet Speed**

The TCP/IP parameters for tweaking a Linux-based machine for fast internet connections are located in /proc/sys/net/... (assuming 2.1+ kernel). This location is volatile, and changes are reset at reboot.

This hack just make the time default values for TCP/IP connection lower so that more connections can be handled by time on your TCP/IP protocol. The following will decrease the amount of time your Linux box will try take to finish closing a connection and the amount of time before it will kill a stale connection. This will also turn off some IP extensions that aren't needed.

**IO Scheduler Changer**

Like the CPU Governor we also benchmark for the best IO Scheduler for battery performance. With this script we change it to the best.

Default value: Deadline

Changed value: Noop

**MicroSD Card Speed**

This script is to speed up the read and write capacity of MicroSD card. We only change the read ahead block size.

Default value: 1024

Changer value: 2048

**Misc Tweak**

**page-cluster**

page-cluster controls the number of pages which are written to swap in a single attempt. The swap I/O size. It is a logarithmic value - setting it to zero means "1 page", setting it to 1 means "2 pages", setting it to 2 means "4 pages", etc. The default value is three (eight pages at a time). There may be some small benefits in tuning this to a different value if your workload is swap-intensive.

Default value: 3

Changed value: 8

**msgmni**

msgmni- maximum number of system-wide System V IPC message queues(IDs) allowed.

The tunable specifies the maximum number of system-wide System V IPC message queue identifiers (one per queue). Each message queue created has an identifier (ID) and there is an upper limit of identifiers.

Applications use the system call to create new queues. If all IDs have been used, will return.

If a process acquires a message queue, but the process terminates with- out removing it, the queue and all its messages remain. Abandoned mes- sage queues can be removed with the command. Message queue status can be monitored using the command.

Default values: 48000

Changed value: 64000

**msgmax**

msgmax- maximum System V IPC message size in bytes(OBSOLETE).

The tunable is obsolete and has been removed. This tunable specified the maximum allowable size, in bytes, of any single message in a System V message queue. This tunable is implied by the tunable msgmnb(5) (the size of a queue).

Default values: 48000

Changed value: 64000

**lease-break-time**

This file specifies the grace period (in seconds) that the kernel grants to a process holding a file lease after it has sent a signal to that process notifying it that another process is waiting to open the file. If the lease holder does not remove or downgrade the lease within this grace period, the kernel forcibly breaks the lease.

Default value: 5

Changed value: 10

**Move Dalvik Cache**

This script is for moving the dalvik cache from system to external microSD card (if inserted). This will help to reduce battery drain by stop mounting the entire dalvik cache on the start of every process.

**Remove Logger**

This script removes the log file.

**VM Management**

**min\_free\_kbytes**

Depending on setup and workload (eg. within a virtual machine with little memory and much I/O) we can get into the situation that the kernel has little memory left, so wants to write some dirty pages to disk, but cannot, because for that it would need some memory free. Now, while that cannot happen with DRBD alone (as this has some small, reserved memory pool to guarantee progress without needing extra memory allocated), we can get into this with eg. iSCSI and md, especially with too low values for the sysctl vm.min\_free\_kbytes (which is set to 128 on some installations, ie. only 128kByte reserved!).

Default value: 128

Changed value: 4096

**oom\_kill\_allocating\_task**

This enables or disables killing the OOM-triggering task in out-of-memory situations. If this is set to zero, the OOM killer will scan through the entire tasklist and select a task based on heuristics to kill. This normally selects a rogue memory-hogging task that frees up a large amount of memory when killed. If this is set to non-zero, the OOM killer simply kills the task that triggered the out-of-memory condition. This avoids the expensive tasklist scan. If panic\_on\_oom is selected, it takes precedence over whatever value is used in oom\_kill\_allocating\_task. The default value is 0 (1 for a very view setup).

Default value: 0

Changed value: 0

**panic\_on\_oom**

This enables or disables panic on out-of-memory feature.

If this is set to 0, the kernel will kill some rogue process, called oom\_killer. Usually, oom\_killer can kill rogue processes and system will survive. If this is set to 1, the kernel panics when out-of-memory happens. However, if a process limits using nodes by mempolicy/cpusets, and those nodes become memory exhaustion status, one process may be killed by oom-killer. No panic occurs in this case. Because other nodes' memory may be free. This means system total status may be not fatal yet. If this is set to 2, the kernel panics compulsorily even on the above-mentioned. Even oom happens under memory cgroup, the whole system panics.

The default value is 0.

1 and 2 are for failover of clustering. Please select either according to your policy of failover. panic\_on\_oom=2+kdump gives you very strong tool to investigate why oom happens.

**laptop\_mode**

If non-zero, this sysctl disables the new 32-bit mmap layout - the kernel will use the legacy (2.4) layout for all processes.

**swappiness**

This control is used to define how aggressive the kernel will swap memory pages. Higher values will increase agressiveness, lower values decrease the amount of swap.

Default value: 60

Changed value: 0

**vfs\_cache\_pressure**

Controls the tendency of the kernel to reclaim the memory which is used for caching of directory and inode objects.

At the default value of vfs\_cache\_pressure=100 the kernel will attempt to reclaim dentries and inodes at a "fair" rate with respect to pagecache and swapcache reclaim. Decreasing vfs\_cache\_pressure causes the kernel to prefer to retain dentry and inode caches. When vfs\_cache\_pressure=0, the kernel will never reclaim dentries and inodes due to memory pressure and this can easily lead to out-of-memory conditions. Increasing vfs\_cache\_pressure beyond 100 causes the kernel to prefer to reclaim dentries and inodes.