Understanding and Optimizing Disk I/O - Strategies, Tools, Hardware, and Applications or Winning over Disk I/O worries

Jeff Kubler

Kubler Consulting

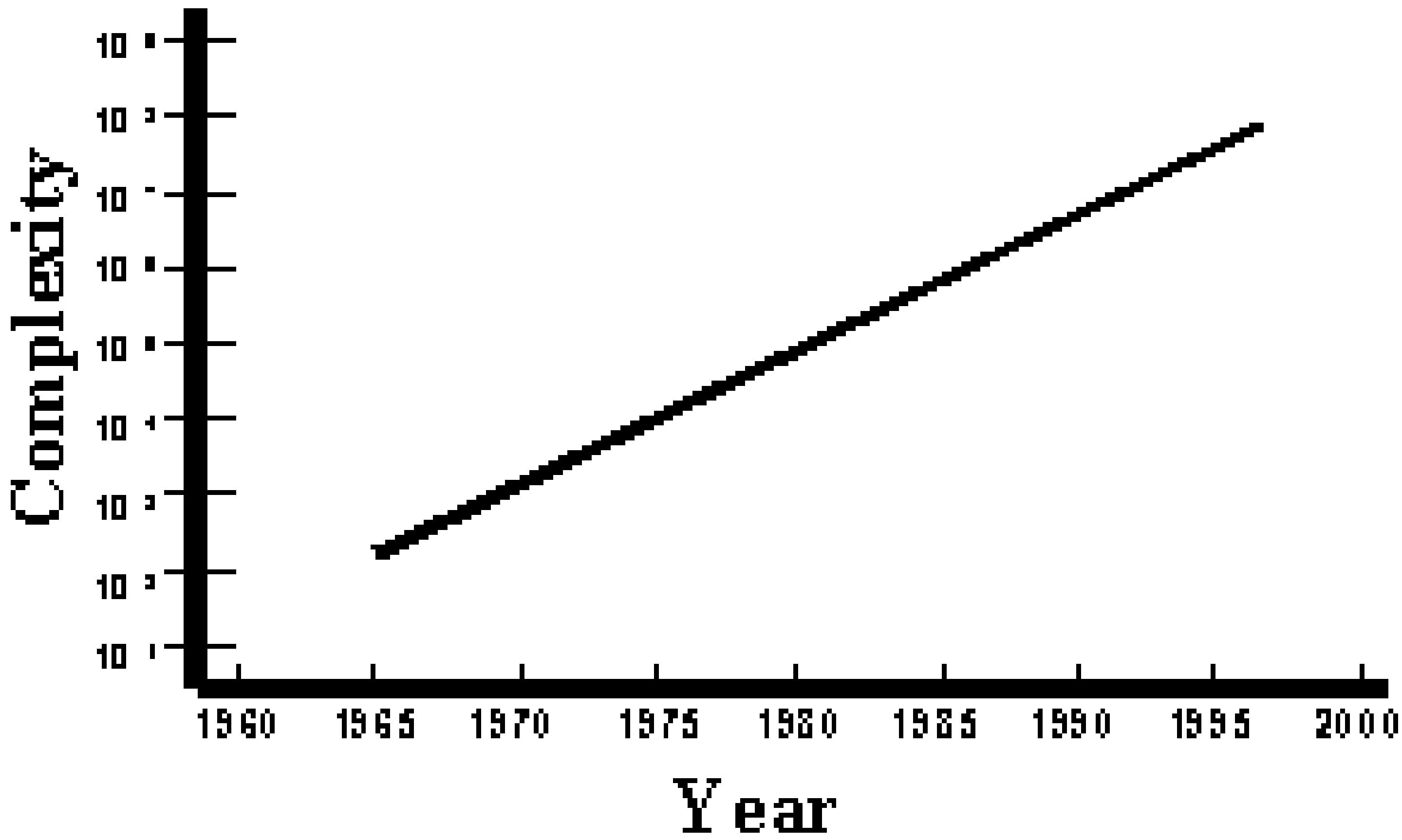
jrkubler@proaxis.com

www.proaxis.com/~jrkubler

Introduction

- CPU processor speeds drastically increases! Moore's Law states that it doubles every 18 months.
- Result: Disk I/O efficiency lags behind.
- More attention to disk needed to preserve efficiency of system.

Moore’s Law


Complexity
1960
1965
1970
1975
1980
1985
1990
1995
2000
Year

Disk I/O Importance

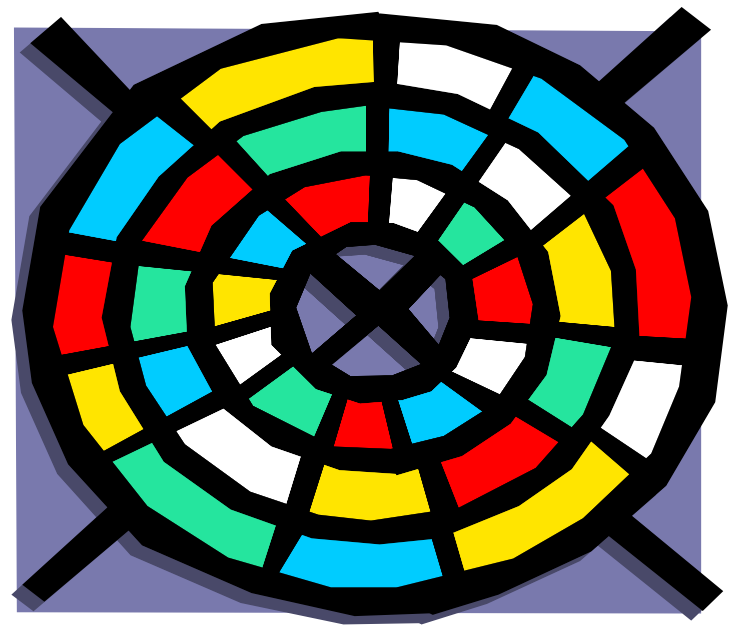
- Mike Loukides said of disk I/O “This is the single most important aspect of I/O performance.” From System Performance Tuning By Mike Loukides O’Rielly & Associates, Inc.

Data Locality

- Describes the location of data on disk (it is sometimes referred to as locality of reference)
- Data Locality encompasses both the issue of the placement of files on disk or on multiple disks and the issue of records within the files placed on disk.

What is Disk I/O?

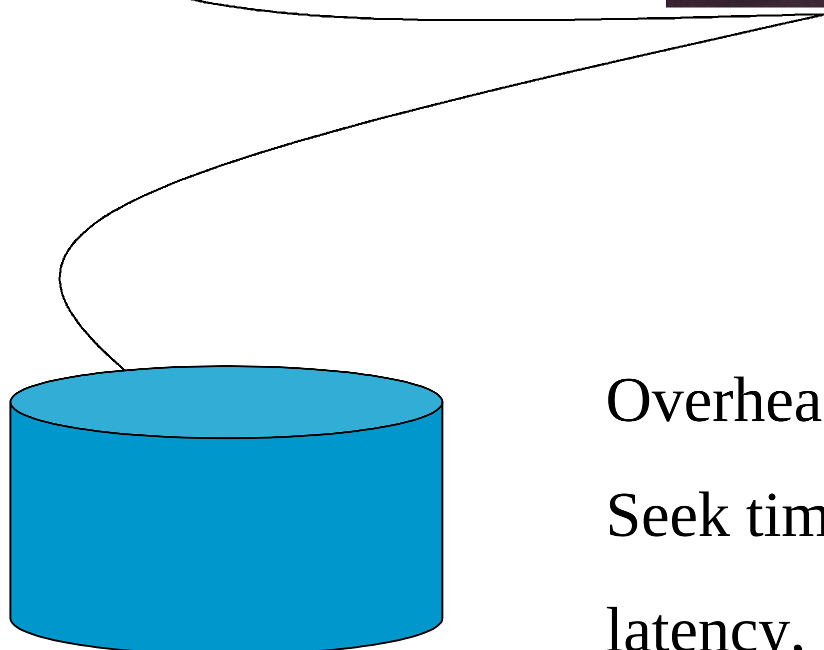
- Act of retrieving and/or updating information stored on a disk drive or in a disk environment.



Overhead - Negotiating the controller.
Seek Time - find data
Latency - wait for data spin.
Xfr (transfer of data) - bring data over.

Anatomy of a Process

- All activity exists a process.
- Processes usually rely on data. Data in one of two places, in memory or on disk.
- If on disk then if updated it must be posted back to disk.
- Disk access is the slowest link.



Overhead - talking to controller

Seek time - looking for data

latency, settling

Transfer - moving data

General Measurements of Disk I/O

- Disk I/O Queue Length
- Pause or Wait for I/O
- Disk Service time
- Disk Utilization
- Total I/O count
- Buffer Cache efficiency
- Response times

Measurement of Disk I/O - vmstat

```
Vmstat -d 5
|procs|  |-memory-| |--------page------|  |-faults-|  |-cpu--|
r  b  w  avm   free re at pi po fr de sr  in  sy  cs  us sy id
1 46  0 2469    466  0  0  0  0  0  0  0 108  37  25   3  2 95
0 47  0 2140    500  1  1  0  0  0  0  0 113  65  30   2  1 97
device    xfer/sec
c0t6d0        0
c0t1d0        0
```

- **Procs: Running, Blocked, Swapped**
- **Memory: Active Virtual Pages; size of memory free**
- **re: Re-claims; Page Freed but Referenced Again**
- **pi/po: Page In/ Out Rates (per second)**
- **fr: pages freed rate**

Measurement of Disk I/O - iostat

- Tin and tout-show char read and written
- CPU metrics - us, ni, sy, id
- bps - kilobytes per second, sps - seeks per second, msps - milliseconds per seek.

Optimal Disk I/O

- Ideal: None at all
- Newest technology
- One channel per drive
- Fully optimized database engine
- Ideally programmed app, no full table scans, etc.

Causes of Disk I/O Inefficiency

- Other priorities
- Fragmentation
- Low disk space
- Short on memory
- Disk I/O imbalance
- Configuration issues

Other Priorities: Data Integrity vs Performance

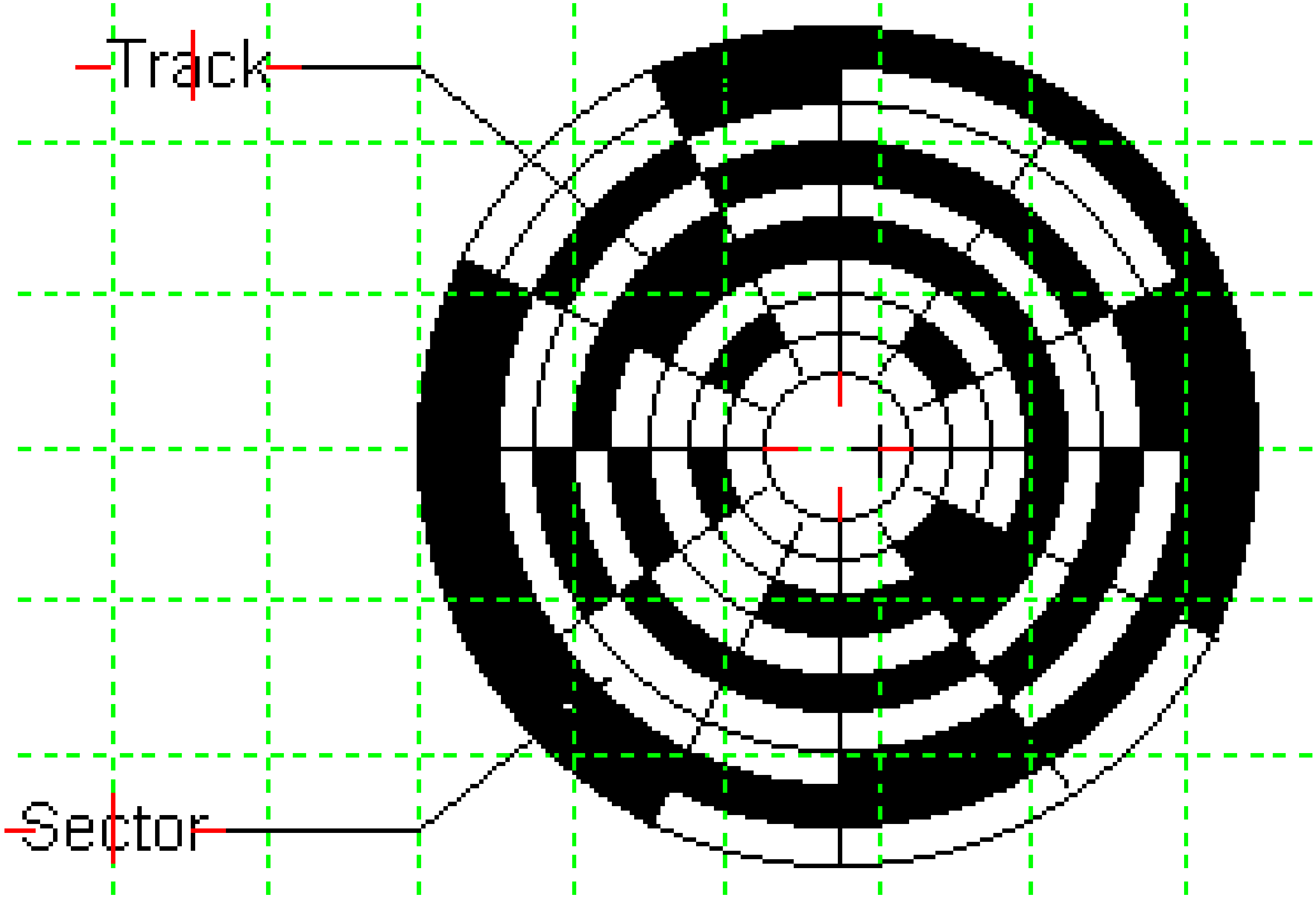
- **High availability vs. fast I/O (mutually exclusive?)**
  - **Mirroring - can be faster to read, however slower to write.**
  - **Raid vs. JBOD - highly write oriented apps may suffer.**
  - **EMC vs JBOD - certain situations have suffered performance issues.**

Memory vs Disk

- Symbiotic relationship, inefficiencies in one will cause the other to work harder.
  – Since disk data must be moved to memory the efficiency of the locality plays a big part in how much I/O must take place to find requested data.

Fragmentation

- Defined as "The propensity of the component disk blocks of a file or memory segments of a kernel data structure to become separated from each other."
  - **Disk Fragmentation**
  - **File Fragmentation**


Track
Sector

Disk I/O Imbalance

- Causes I/O "hot spots"
- Hot spots cause higher disk I/O queue length
- Higher disk utilization levels

Inadequate Disk Space

- This can severely impact system performance.
- Can also stop applications from running.

File System Optimization

- HFS, JFS, NFS.
- Suggestions:
  - Distribute the workload evenly
  - Keep similar files on the same file system
  - Give file systems a block size appropriate to activity expected.
  - Don’t use file system paging.

HFS vs JFS

- HFS
- Older of the two, not as patched.
- Fsck can take a long time to process during recovery

- JFS
- Fast on recovery
- With patches speed has increased.
- Don’t turn on many logging options

Configuration issues

- Too few controllers or too many drives per controller.
- Too small or inappropriately placed swap space.

IOSCAN

```
#  ioscan
H/W Path    Class                      Description
==============================================
8/4                 ext_bus                  GSC add-on Fast/W
SCSI Interface
8/4.5                  target
8/4.5.0                   disk               SEAGATE ST32550W
8/4.7                  target
8/4.7.0                   ctl                Initiator
8/4.8                  target
```

8/4 - bus 8, converter 4

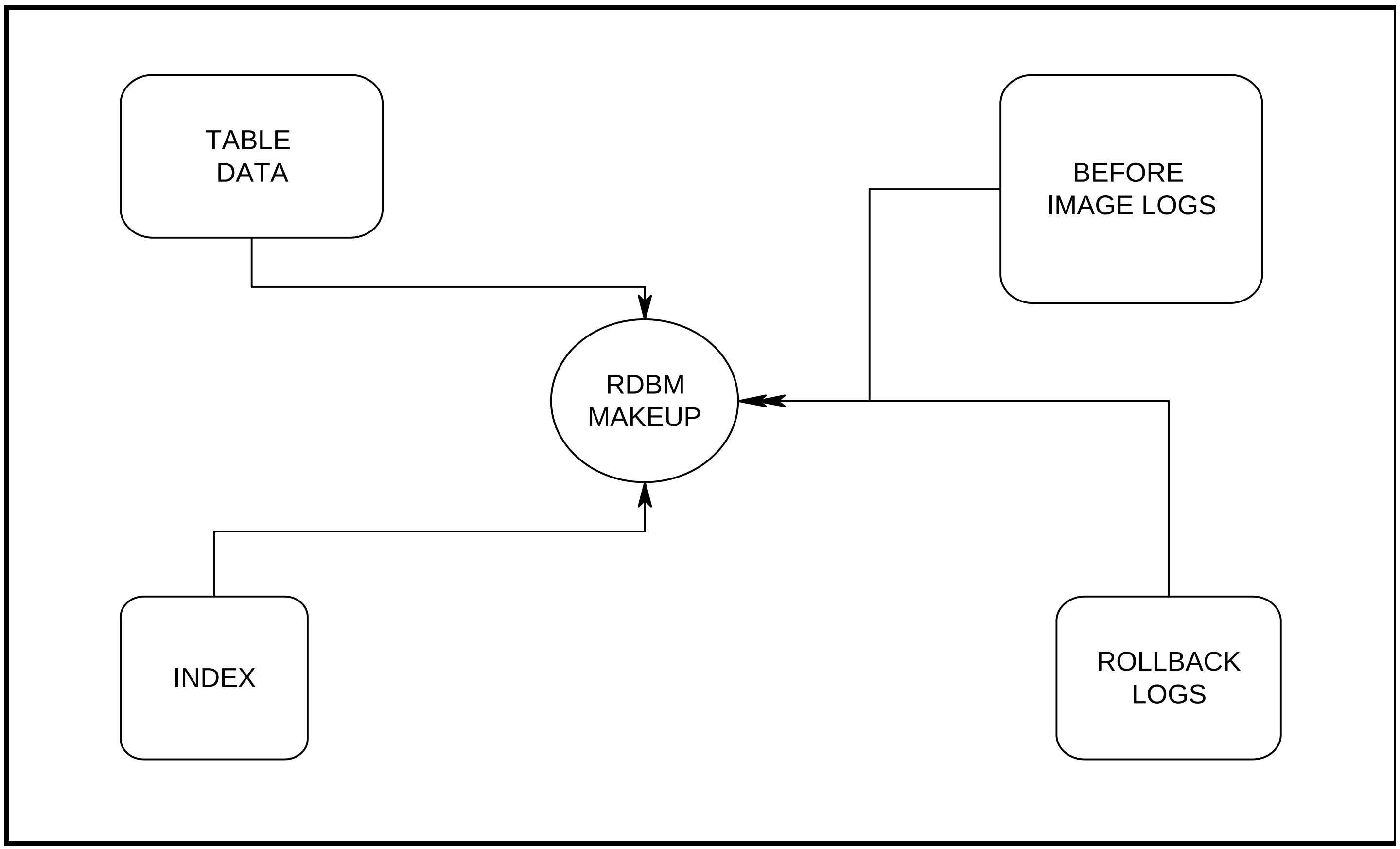
8/4.5 - bus 8, FW SCSI bus 4, target 5

8/4.5.0 bus 8, FW SCSI bus 4, target 5 whole disk

Relational Database inefficiencies

- **Example: ORACLE, INGRESS, INFORMIX, PROGRESS**
- **Consist of: Tables, Indexes, Rollback logs, and Before Image Logs**
- **Suggestions:**
  - Optimize placement of Tables and Indexes
    - Place table files, indexes, and logs on separate disk drives.
  - Use supplied optimization tools

Relational Database Inefficiencies



Strategies

- **Memory**
- **Buffer Cache**
- **JBOD**
  - **balance I/O, work on fragmentation,**
- **Striping**
- **Raw I/O Vs. File System I/O**

Strategies - Memory

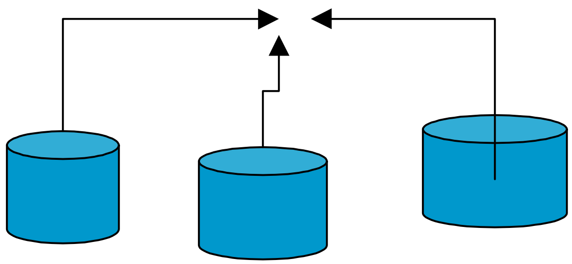
- Scratch pad of all work
- Best strategy - get all the memory you can!
- Use Virtual Memory - usually 2x the size of memory (larger memory systems not the rule).

Strategies - Buffer Cache

- **"The buffer cache is a pool of buffers that provides intermediate storage for data moving to or from the system's disk drives." System Performance Tuning, by Mike Loukides**.
- Too low will cause additional I/O.
- DBC Min/DBC Max - What percentage is best?

Strategies - JBOD

- Stands for “Just a bunch of disks”
- Very straight forward
- Easier to think about in terms of placement of files, etc.
- No data protection



Strategies - Striping

- Writing data to multiple disks to increase throughput.
- Try to achieve parallelism in reads and writes. Requires separate controllers.
- Any one disk goes down, ouch!

Strategies - Raw vs file system I/O

- Is favored by database applications as it bypasses the file system management routines. Reads and writes are made directly from memory to the surface of the disk.
- Had seen comments that this could increase performance by 30 %.
- I/O is not buffered.

Tools

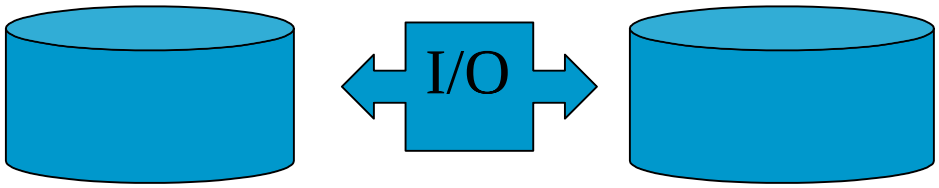
- **Reloads**
- **Online JFS - **Journal File System
  Online defragmenter
- **Diagnostics tools**: iostat, bdf, vmstat, Glance, Sar, SOS, etc.

Hardware

- Mirroring
- Raid (Redundant Array of Inexpensive Disks)
- Autoraid
- Solid State Disk
- SSA Drives
- Large Cached Storage Systems
  - **EMC**

Mirroring

- Copy data to two places, slows writes.
- Reads data from 2 places, speeds read
- Expense, need a duplicate of every disk



Raid & Autoraid

- Provide several levels of redundancy of data.
- Raid levels (Raid 0 = striping, Raid 1 = mirror, Raid 5 = strip data & parity on several drives, etc.)
- AutoRaid - easy to install, redundant, most active data in Raid 1, less Raid 5.

Solid State

- Large cache boxes
- Hot files are kept in ssd device
- Contains intelligence to see busiest files

SSA: New Standard!

- Low Cost/High Perf. Serial Interface
- 96 Disks/Adapter
- Up to 320 MB/sec on a Single Adapter
- Up to 2400m Between Nodes
- Simple Twisted Pair Cable
- 3 vs. 12 SCSI Cmds per I/O Transfer

Applications

- Logical Volume Manager
- Diskpak
- Seekrite
- Syncsort -
- SUPRTOOL -

Solutions

- Optimize your databases and database access
- Spread out the I/O
- Upgrade to the latest technology disk drives
- Avoid configuration problem
- Deal with fragmentation

Solutions

- Avoid disk space problems
  - Use the compress command.
  - Delete core dump files.
  - Configure filesystems with small block size.
  - Configure filesystems with less free space.
  - Use quotas, monitor with cron, etc.

Conclusion

- Seek to understand the nature of I/O
- Try to find ways to reduce I/O
- Practice management of I/O
- Maximize memory/buffer/swap
- Remember disk I/O is the weakest link in the chain!

The End

Thanks for coming!