

# AIX I/O Performance Tuning

Jaqui Lynch

Mainline Information Systems

jaqui.lynch@mainline.com

<http://www.circle4.com/jaqui/papers/aixioperf.pdf>

Session 6829



1

## Agenda

- Logical Volume Manager
- Vmtune
- Iostat
- Monitor or top
- File caching
- filemon/fileplace
- Policies
- I/O Pacing
- RAID Levels

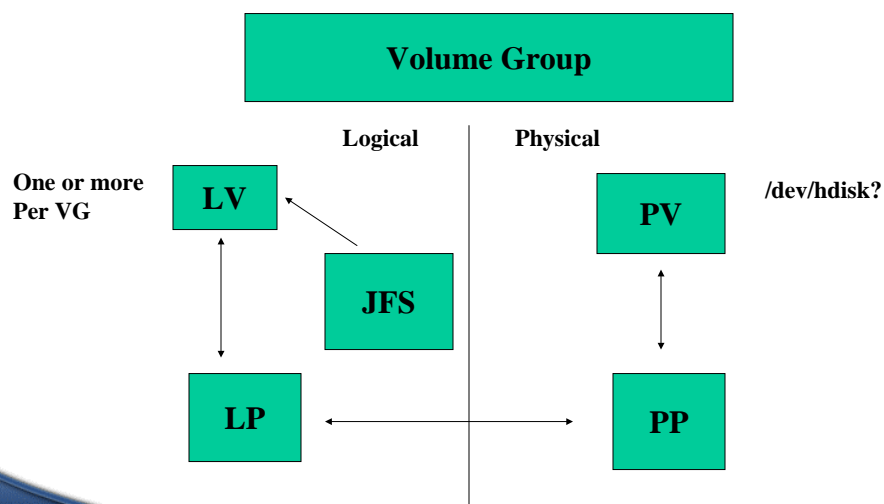


2

## LVM – Logical Volume Manager

- Commands, library subroutines and tools
- Controls disk resources
- LVDD – logical volume device driver
  - Manages and process all I/Os
  - Translates logical addresses to physical addresses
- Disk read or write
  - Involves filesystem, VMM and LVM
- Physical Volume
- Logical Volume
- Volume Group

## LVM Layout



# JFS

- Journal Filesystem
  - Pool of 4k blocks (frags)
- Fragsize
  - 512, 1024, 2048 or 4096
  - Smaller fragsize allows small files to be stored more efficiently
- Fragmentation and Extents
  - Increases seeks and longer I/O response
- JFS Logs

# Filesystems

- Stored in blocks of the fragsize (4096 def)
- Inode has 8 pointers to data blocks if file <32kb
  
- <=32kb Inode to 1-8 x 4k blks
- <=4mb Inode to indirect blk to 1024 x 4k blks
- >4mb <=2gb
  - Inode to dblIndirect (512) to indirect ...
- AIX 4.2 BF filesystem
  - Double indirects point to 128kb blocks instead of 4kb ones
  - First indirect is 4kb
  - Max size is 64gb
- Great description in Certification Guide for Perf.

## JFS Logs

- Max size = 256mb
- Default is 1 x PP
- All filesystems in a VG use the samelog by default
- If sync I/O or many files created and deleted then this file will be hot
- `mklv -t jfslog -y JFSLOGdb2 testvg 2 hdisk4`
- `logform /dev/JFSLOGdb2`
- `chfs -a log=/dev/JFSLOGdb2 /filesystem`
- Umount and mount filesystem

## vmstat (AIX)

**IGNORE FIRST LINE - average since boot**  
**Run vmstat over an interval (i.e. vmstat 2 30)**

Kthr		memory		Page				Faults				CPU				
r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa
2	3	253292	5	0	1	262	1783	2949	0	941	13281	1877	12	29	0	60
2	3	255503	0	0	0	260	1110	1879	0	911	14725	1049	22	25	0	53
1	2	273017	487	0	0	0	0	0	0	507	12627	422	12	20	65	3

fre should be between minfree and maxfree  
fr:sr ratio 1783:2949 means that for every 1783 pages freed 2949 pages had to be examined.

To get a 60 second average try: `vmstat 60 2`

## vmstat (AIX)

kthr		memory			page				faults			cpu				
r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa
0	0	308376	165	0	0	0	2	11	0	121	1150	133	8	4	77	10
1	5	309809	465	0	2	580	1001	2033	0	1174	83870	1203	12	32	1	55
1	3	310054	321	0	0	76	55	83	0	665	98714	619	20	30	1	48
3	2	310066	318	0	0	33	44	97	0	677	99889	686	36	36	4	24
2	3	309657	339	0	2	0	0	0	0	613	145959	899	28	32	9	32
1	2	308326	1648	0	1	0	0	0	0	486	92649	811	13	27	43	17
1	2	308326	1646	0	0	0	0	0	0	450	90150	155	4	23	72	1
1	2	308276	1707	0	0	0	0	0	0	463	90456	326	8	23	64	5
1	2	308276	1701	0	0	0	0	0	0	453	91140	507	10	24	63	2
0	2	308256	1718	0	0	0	0	0	0	466	4708	655	21	4	69	6

## /usr/samples/kernel/vmtune

### AIX Specific

In AIX this is part of the bos.adt.samples fileset

Run vmtune and look at the following:

minperm default is 20%

maxperm default is 80%

numperm This is what percent of real memory is currently being used for caching files - if it is high reduce maxperm to 30 to 50%

Reducing maxperm stops file caching affecting programs that are running

Also maxrandwrt - default is 0 - try 32

## vmtune output

vmtune: current values:

-p	-P	-r	-R	-f	-F	-N	-W		
minperm	maxperm	minpgahead	maxpgahead	minfree	maxfree	pd_npages	maxrandwrt		
13104	65524	2	8	120	128	524288	32		
-M	-w	-k	-c	-b	-B	-u	-l	-d	
maxpin	npswarn	npskill	numclust	numfsbufs	hd_pbuf_cnt	lvm_bufcnt	lrubucket	defps	
104839	4096	1024	1	93	208	9	131072	1	
-s		-n	-S	-h					
sync_release_ilock		nokilluid	v_pinshm	strict_maxperm					
0		0	0	0					

number of valid memory pages = 131048  
maximum pinable=80.0% of real memory  
number of file memory pages = 100060

maxperm=50.0% of real memory  
minperm=10.0% of real memory  
numperm=76.4% of real memory

## rc.tune 1/2

```
#
# rc.tune is called by inittab to tune the network and system parameters
#
# To make no changes permanent use the no -opp option
/usr/sbin/no -rp -o rfc1323=1
/usr/sbin/no -rp -o sb_max=1310720
/usr/sbin/no -rp -o subnetsarelocal=1
/usr/sbin/no -rp -o tcp_sendspace=262144
/usr/sbin/no -rp -o tcp_recvspace=262144
/usr/sbin/no -rp -o udp_sendspace=65536
/usr/sbin/no -rp -o udp_recvspace=655360
/usr/sbin/no -rp -o tcp_mssdflt=1448
#
/usr/sbin/no -rp -o thewall=131072
/usr/sbin/no -rp -o ipqmaxlen=256
#
# If this works well for you try setting maxperm to 30 instead of 50
```

## rc.tune 2/2

```
# /usr/samples/kernel/vmtune -p 10 -P 50 -W 32
# vmtune is being phased out and replaced by ioo and vmo
#
/usr/sbin/vmo -p -o minperm%=10
/usr/sbin/vmo -p -o maxclient%=50
/usr/sbin/vmo -p -o maxperm%=50
/usr/sbin/ioo -p -o maxrandwrt=32
/usr/sbin/nfso -p -o nfs_rfc1323=1
#
# If this works well for you try setting maxperm to 30 instead of 50
```

## ioo Output

```
minpgahead = 2
maxpgahead = 8
pd_npages = 65536
maxrandwrt = 0
numclust = 1
numfsbufs = 186
sync_release_ilock = 0
lvm_bufcnt = 9
j2_minPageReadAhead = 2
j2_maxPageReadAhead = 8
j2_nBufferPerPagerDevice = 512
j2_nPagesPerWriteBehindCluster = 32
j2_maxRandomWrite = 0
j2_nRandomCluster = 0
hd_pvs_opn = 1
hd_pbuf_cnt = 320
```

## vmo Output

```
memory_frames = 65536
  maxfree = 128
  minfree = 120
  minperm% = 20
  minperm = 11813
  maxperm% = 80
  maxperm = 47255
strict_maxperm = 0
  maxpin% = 80
  maxpin = 52429
  maxclient% = 80

lrubucket = 131072
  defps = 1
  nokilluid = 0
  numpsblks = 131072
  npskill = 1024
  npswarn = 4096
  v_pinshm = 0
pta_balance_threshold = 50
  pagecoloring = 0
  framesets = 0
  mempools = 0
  lgpg_size = 0
  lgpg_regions = 0
  num_spec_dataseg = n/a
  spec_dataseg_int = n/a
  memory_affinity = n/a
```

## lsps -a (similar to pstat)

- Ensure all page datasets the same size although hd6 can be bigger - ensure more page space than memory
- Only includes pages allocated (default)
- Use lsps -s to get all pages (includes reserved via early allocation (PSALLOC=early))
- Use multiple page datasets on multiple disks

# lsps output

```
lsps -a
Page Space Physical Volume Volume Group Size %Used Active Auto Type
paging05   hdisk9       pagvg01    2072MB  1  yes  yes  lv
paging04   hdisk5       vgpaging01 504MB   1  yes  yes  lv
paging02   hdisk4       vgpaging02 168MB   1  yes  yes  lv
paging01   hdisk3       vgpagine03 168MB   1  yes  yes  lv
paging00   hdisk2       vgpaging04 168MB   1  yes  yes  lv
hd6        hdisk0       rootvg     512MB   1  yes  yes  lv
```

```
lsps -s
Total Paging Space Percent Used
3592MB             1%
```

Bad Layout above  
Should be balanced  
Make hd6 the biggest by one lp or the same size as the others

# iostat (AIX)

Run iostat ver an interval (i.e. iostat 2 30)

tty:	tin	tout	avg cpu: %user	%sys	%idle	%iowait
(kp)	17.5	2.9	0.0	75.4	2.9	23.4
(te)	0	0	9	8	24	59

Disks:	%tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk1	3.5	32	3.5	0	64
hdisk0	59	27.6	59.5	44	508
hdisk2	12	64	16	0	128
hdisk3	3.5	32	3.5	0	64

%user + %sys <= 80%  
Similar info from ps au or sar -u  
%iowait – time waiting on disk I/O – includes local and nfs  
Historical disk I/O collection is off by default on AIX 5.1

## More Iostat

```
tty:  tin   tout  avg-cpu: % user  % sys  % idle  % iowait
      0.0   0.0    10.2  20.4   11.1   58.2
```

```
Disks:  % tm_act  Kbps  tps  Kb_read  Kb_wrtn
hdisk0  50.0    284.0  59.0   16    552
hdisk1   8.0    526.0   6.0  1052     0
hdisk27  30.5    178.0  34.0   4    352
hdisk36  59.0    330.0  71.5  108    552
hdisk37  29.5    176.0  33.5   0    352
hdisk38   0.0     0.0   0.0   0     0
hdisk2   30.5    200.5  37.5   1    400
hdisk8   1.5     6.0   1.5   0    12
hdisk11  31.0    200.5  37.5   1    400
hdisk21  30.5    210.5  37.0   1    420
hdisk10  31.0    210.5  38.0   1    420
hdisk6   0.0     0.0   0.0   0     0
hdisk7   3.5   1048.5  12.0  2097     0
cd0      0.0     0.0   0.0   0     0
```

%tmact time physical disk was active - >40% means processes probably waiting  
Adapter throughput – add up the kbps per drive on each adapter

## Intra and Inter Policies

- Intra Policy
  - How data is laid out on the disk
  - Outer edge, outer middle, center, inner middle, inner edge
  - If  $\leq 4$ gb disk then center is fastest seek
  - If  $> 4$ gb then outer edge
- Inter Policy
  - How data is laid out between/across disks
  - Minimum as few disks as possible
  - Maximum as many disks as possible

## I/O Pacing

- Set high value to multiple of  $(4*n)+1$
- Limits the number of outstanding I/Os against a file
- Minpout – minimum
- Maxpout – maximum
- If process reaches maxpout then it is suspended from creating I/O until outstanding requests reach minpout

## Other I/O performance notes

- Mirroring of disks
  - Islv to check # copies
  - Mirror write consistency
- Write verify
- BB Policy
- Logical volume scheduling policy
  - Parallel
  - Parallel/sequential
  - Parallel/roundrobin
  - Sequential
- Async I/O

## Disk Technologies

- Arbitrated
  - SCSI 20 or 40 mb/sec
  - FC-AL 100mb/sec
  - Devices arbitrate for exclusive control
  - SCSI priority based on address
- Non-Arbitrated
  - SSA 80 or 160mb/sec
  - Devices on loop all treated equally
  - Devices drop packets of data on loop

## Adapter Throughput - SCSI

Courtesy of <http://www.scsita.org/terms/scsiterms.html>

	100%	70%	Bits	Max
	mby/s	mby/s	Bus	Devs
				Width
➤ SCSI-1	5	3.5	8	8
➤ Fast SCSI	10	7	8	8
➤ FW SCSI	20	14	16	16
➤ Ultra SCSI	20	14	8	8
➤ Wide Ultra SCSI	40	28	16	8
➤ Ultra2 SCSI	40	28	8	8
➤ Wide Ultra2 SCSI	80	56	16	16
➤ Ultra3 SCSI	160	112	16	16
➤ Ultra320 SCSI	320	224	16	16
➤ Ultra640 SCSI	640	448	16	16

- Watch for saturated adapters

## Adapter Throughput - Fibre

100% mbit/s	70% mbit/s
➤ 133	93
➤ 266	186
➤ 530	371
➤ 1 gbit	717
➤ 2 gbit	1434

➤ SSA comes in 80 and 160 mb/sec

## RAID Levels

- Raid-0
  - Disks combined into single volume stripeset
  - Data striped across the disks
- Raid-1
  - Every disk mirrored to another
  - Full redundancy of data but needs extra disks
  - At least 2 I/Os per random write
- Raid-0+1
  - Striped mirroring
  - Combines redundancy and performance

## RAID Levels

- RAID-5
  - Data striped across a set of disks
  - 1 more disk used for parity bits
  - Parity may be striped across the disks also
  - At least 4 I/Os per random write (read/write to data and read/write to parity)
  - Uses hot spare technology

## Striping

- Spread data across drives
- Improves r/w performance of large sequential files
- Can mirror stripe sets
- Stripe width = # of stripes
- Stripe size
  - Set to 64kb for best sequential I/O throughput
  - Any  $N^2$  from 4kb to 128kb

## General Recommendations

- Different hot LVs on separate physical volumes
- Stripe hot LV across disks to parallelize
- Mirror read intensive data
- Ensure LVs are contiguous
  - Use lslv and look at in-band % and distrib
  - reorgvg if needed to reorg LVs
- Writeverify=no
- minpgahead=2, maxpgahead=16 for 64kb stripe size
- Increase maxfree if you adjust maxpgahead
- Tweak minperm, maxperm and maxrandwrt
- Tweak lvm\_bufcnt if doing a lot of large raw I/Os

## >30% I/O going to Paging

- Write more memory efficient code
- Add memory
- Reschedule the process
- Tweak minperm/maxperm
- Be aware of early/late/deferred paging
  - Since AIX 4.3.2 now does deferred instead of late paging
  - Switched back to late at AIX 5.1
- Add more page space
  - More datasets (concurrency)
  - Larger size

## >30% I/O going to normal f/s

- Problem probably user I/O
- Check fragmentation
- Reorganize the filesystem
- Add physical volumes - watch adapter throughput
- Split the data somehow or spread it
- Adding memory may help with caching but don't count on it
- Add jfs logs

## Summary

