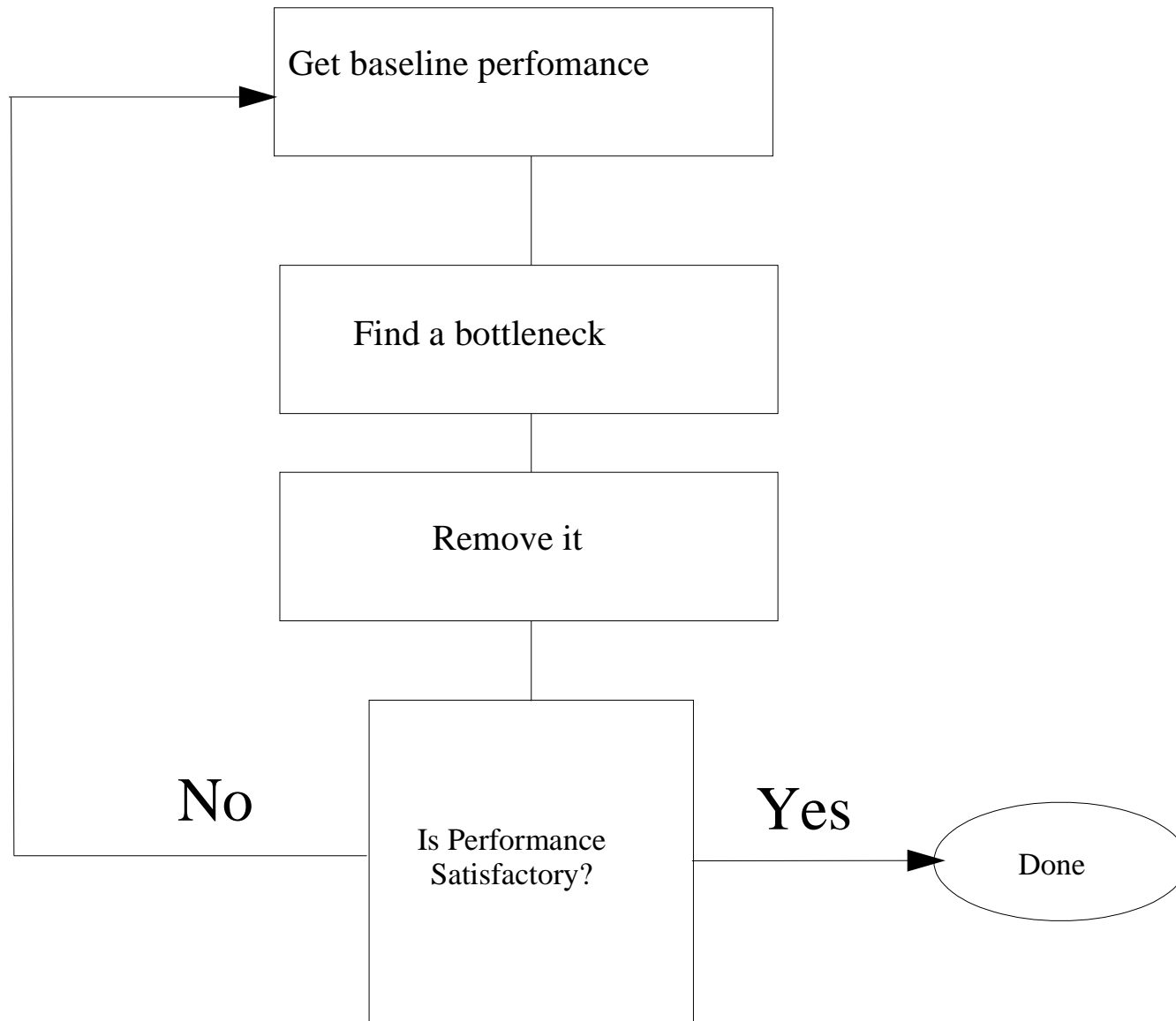


OS Performance Analysis Basic Procedure



Potential Bottlenecks

- Disk
- Disk
- Disk
- Network
- Memory
- CPU

Disks Bottlenecks

- Not enough capacity
- Slow response time
- Poor Layout
- RAID configurations
- File system issues
- Database Issues

Identifying Disk bottlenecks

- Use sar, iostat
- look for service time, disk utilization, queue length, uneven distribution
- know whether array caching is being used

Relative Access times

Device	Real time (1s - 1 ns)	Seconds	Relative time
CPU Registers	2 nsec	2×10^{-9}	2 seconds
CPU cache	20 nsec	20×10^{-9}	20 second
Main Memory	200 nsec	20×10^{-8}	2-3 minutes
Disk	20 msec	20×10^{-3}	7 months

iostat -x 30 information

Wait - queue length, number of entries waiting for disk

svc_t - service time in millisecc. >50 constantly is bad

%w - percent of time queue occupied

%b - percent busy <30% good >60% bad

```
                extended device statistics
device    r/s    w/s    kr/s    kw/s  wait  actv  svc_t  %w  %b
fd0       0.0    0.0    0.0    0.0   0.0   0.0  279.0   0   0
sd0       0.0    0.0    0.0    0.0   0.0   0.0   0.0   0   0
sd1       0.1    0.9    0.7    6.3   0.0   0.1  72.1   0   1
sd4       0.0    0.0    0.0    0.0   0.0   0.0   0.0   0   0
sd6       0.0    0.0    0.0    0.0   0.0   0.0   0.0   0   0
nfs1      0.0    0.0    0.0    0.0   0.0   0.0   0.7   0   0
nfs2      0.0    0.0    0.1    0.4   0.0   0.0  298.1   0   0
nfs3      0.0    0.0    0.2    0.0   0.0   0.0  35.4   0   0
```

iostat -x -n -p Example

```
extended device statistics
  r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
46.0    6.6   102.0    7.0  1.8  2.0    33.8   37.2   28 100 c0t0d0
46.0    6.6   102.0    7.0  1.8  2.0    33.8   37.2   28 100 c0t0d0s0
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s1
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s2
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s3
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s7

extended device statistics
  r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
44.8   23.2   221.2   41.7  6.5  2.0    95.1   29.1   63 100 c0t0d0
44.8   23.0   221.2   40.1  6.4  2.0    94.6   29.1   63 100 c0t0d0s0
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s1
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s2
  0.0    0.2    0.0    1.6  0.0  0.0   237.9   26.9    5  1  c0t0d0s3
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s7

extended device statistics
  r/s    w/s    kr/s    kw/s wait actv wsvc_t asvc_t  %w  %b device
50.8    0.0   105.4    0.0  0.0  1.9     0.3   38.2    2 100 c0t0d0
50.8    0.0   105.4    0.0  0.0  1.9     0.3   38.2    2 100 c0t0d0s0
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s1
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s2
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s3
  0.0    0.0    0.0    0.0  0.0  0.0     0.0    0.0    0  0  c0t0d0s7
```

What is Service time?

- Time waiting on queue
- SCSI commands
- Head seek (1-15 msec)
- Rotational latencies (0-10 msec)
- Data transfer time
- Interrupt response

What can we do about Disk Bottlenecks

- Balance the load (striping, partitioning)
- More disks
- UFS Logging eliminates fsck
- Distribute swap
- Put related data on same partitions
- Don't fill up the disk
- Add memory (Array Cache/UFS/DB SGA)

Databases and file systems

- Default **newfs** parameters are NOT appropriate for DB
- UFS single writer lock prevents multiple writes to the same DB file
- use multiple DB files or ...
- use raw disk
- Mount file systems using concurrent direct I/O option

Raw

vs.

UFS

- Fast
- Eliminates single writer issue
- Difficult to manage
- Difficult to backup
- No cache or buffering

- Slower
- Easier to manage
- Easier to backup
- Listed in vfstab, mount, df, ls
- Cached in RAM

Measuring Network Throughput

netstat -i 5

Collision rate = $\text{colls} / \text{output packets}$
should be <5%

input	hme0	output		input	(Total)	output			
packets	errs	packets	errs	colls	packets	errs	packets	errs	colls
7111436	0	6947310	0	385458	7111436	0	6947310	0	385458
13	0	0	0	0	13	0	0	0	0
21	0	1	0	0	21	0	1	0	0
32	0	18	0	0	32	0	18	0	0

Alleviating Network problems

- Subnets
- Switched Ethernet
- Fast, Gigabit Ethernet
- ATM 155 or 622 for WANS
- Bandwidth Manager
- ndd tuning

What is NDD?

- Network driver configuration
- Controls TCP/IP parameters for data flow
- Highly recommended for web servers
- Also important for security and DoS attacks
- Tutorials
 - http://www.sun.com/sun-on-net/performance/TCP_tutorial.pdf
 - <http://www.sean.de/Solaris/tune.html>

Memory Bottle necks

- Not enough real RAM
- Not enough system virtual memory
- Not enough process virtual space
- Unnecessary/unused processes
- User process memory allocation

What are swapping and paging? When do they occur? Who is responsible?

- Paging is the process of freeing or writing to disk (swap or ufs) pages not currently needed
- When memory gets low (pageout)
- When disk writes are required (fsflush, sync)
- Swapping removes entire processes out of memory (sched)
- When memory extremely low

Determining memory status

vmstat 5 (partial)

- w processes waiting to be swapped in
- swap, free - available space in KB
- sr - scan rate 0 is good, nonzero is bad

procs			memory		page						
r	b	w	swap	free	re	mf	pi	po	fr	de	sr
0	0	0	11584	10824	0	136	1	0	1	0	0
0	0	0	56496	3192	0	297	0	4	4	0	0
0	0	0	56464	3160	0	243	0	4	4	0	0
0	0	0	56304	3144	0	252	0	1	1	0	0
0	0	0	56416	3112	0	252	0	4	4	0	0

Determining Virtual Memory Usage

```
swap -s  
total: 147560k bytes allocated  
+ 13800k reserved = 161360k used, 73968k available
```

Alleviating Memory problems

- Resource Managers can control VM usage.
- Memory bottleneck is also CPU, Disk bottleneck!
- Add memory or eliminate processes if scan rate too high
- Add swap if swap low (swap -a)
- Exit unused processes, control pigs, move some to other computers.
- Adjust page algorithms (kernel tuning)

CPU Bottlenecks

- Too much system time
- Process priorities
- Lock contention

Determining CPU Utilization

mpstat

smtx - number of times a mutex lock not gotten
(>200/CPU is bad)
usr - CPU executing for user
sys - CPU executing for system (>35% is bad)
wt - idle time waiting for I/O to complete
idl - idle CPU time

CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
10	136	0	614	300	100	297	51	20	15	0	827	4	4	19	73
11	71	0	105	111	5	384	102	17	4	0	645	2	6	19	74
14	78	0	126	221	75	387	90	17	13	0	585	3	2	32	64
15	112	0	71	444	337	314	73	19	4	0	756	2	4	10	84

Determining CPU Utilization

sar -q 5 30

NOTE: Zeros are blank filled.

runq-sz - Size of the system-wide run queue
>4/CPU is bad

%runocc - % of time run queue occupied

16:27:40	runq-sz	%runocc	swpq-sz	%swpocc
16:27:45				
16:27:50				
16:27:55				
16:28:00	1.0	20		
16:28:50	1.0	40		
16:28:55	2.0	20		
16:29:00	2.0	20		
16:29:25	2.0	20		

CPU usage example

Vmstat output

procs			memory		page							disk				faults			cpu		
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	m1	m2	m3	m4	in	sy	cs	us	sy	id
0	0	0	724616	32464	0	0	0	0	0	0	0	0	0	0	0	628	1393	1095	0	0	99
0	0	0	724616	32464	0	22	0	0	0	0	0	0	0	0	0	666	20472	1134	3	3	95
0	0	0	724640	32488	0	22	0	0	0	0	0	13	13	13	0	787	116734	1131	12	14	73
0	0	0	724648	32496	0	21	0	0	0	0	0	0	0	0	0	649	118208	1122	13	14	74

mpstat output

CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
10	8	0	325	300	100	313	99	3	1	0	239	0	0	0	100
11	8	0	28	109	7	349	98	6	1	0	665	0	0	0	99
14	5	0	2	239	90	331	99	5	1	0	266	0	1	0	99
15	0	0	13	114	10	134	8	6	1	0	447	2	1	0	97
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
10	11	0	338	301	100	262	83	10	4	0	23226	10	11	0	79
11	0	0	3	117	2	213	65	18	17	0	60267	27	25	6	42
14	5	0	4	193	56	366	99	20	8	0	1892	1	5	21	73
15	5	0	20	266	156	294	82	18	11	0	20119	8	10	0	82
CPU	minf	mjf	xcal	intr	ithr	csw	icsw	migr	smtx	srw	syscl	usr	sys	wt	idl
10	0	0	324	300	100	309	97	6	2	0	392	0	0	0	99
11	0	0	0	118	2	172	59	16	7	0	66116	27	30	0	43
14	0	0	0	208	60	234	73	19	7	0	50734	22	23	0	55
15	0	0	0	125	22	385	104	23	1	0	638	0	1	0	99

prstat (Solaris)

- User, project, processor based CPU utilization
- lightweight thread information
- System/user/sleep time per process
- trap, signal, system call, page fault information
- user lock information

Sample prstat output

NPROC	USERNAME	SIZE	RSS	MEMORY	TIME	CPU
42	jlaurent	559M	251M	69%	2:05:00	76%
34	root	253M	104M	29%	4:50:42	3.4%
1	daemon	2648K	1208K	0.3%	0:00:00	0.0%

PID	USERNAME	USR	SYS	TRP	TFL	DFL	LCK	SLP	LAT	VCX	ICX	SCL	SIG	PROCESS/NLWP
16691	jlaurent	33	33	-	-	-	-	39	-	459	328	12K	0	prstat/1
16561	jlaurent	4.5	2.6	-	-	-	-	92	-	80	120	402	0	soffice.bin/5
9426	jlaurent	1.9	0.8	-	-	-	-	97	-	100	32	2K	23	netscape/1
17863	root	1.3	0.3	-	-	-	-	98	-	291	14	2K	2	Xsun/1
24684	jlaurent	0.1	0.0	-	-	-	-	100	-	10	0	23	0	wish8.1/1

PID	USERNAME	SIZE	RSS	STATE	PRI	NICE	TIME	CPU	PROCESS/LWPID
65	root	2768K	800K	sleep	18	0	0:00:00	0.0%	picld/4
65	root	2768K	800K	sleep	58	0	0:00:00	0.0%	picld/3
65	root	2768K	800K	sleep	58	0	0:00:00	0.0%	picld/2
65	root	2768K	800K	sleep	18	0	0:00:00	0.0%	picld/1
58	root	2072K	928K	sleep	59	0	0:00:00	0.0%	syseventd/11
58	root	2072K	928K	sleep	59	0	0:00:00	0.0%	syseventd/10
58	root	2072K	928K	sleep	52	0	0:00:00	0.0%	syseventd/9
58	root	2072K	928K	sleep	52	0	0:00:00	0.0%	syseventd/8
58	root	2072K	928K	sleep	52	0	0:00:00	0.0%	syseventd/7

CPU Run Queues

- Run queue indicates how far over utilized (or under configured) your system is.
- $\text{load average} = \text{running} + \text{run queue}$

Alleviating CPU problems

- More or Faster CPUs
- Add CPU cache
- Resource Manager
- Add memory if scan rate high
- Adjust process priorities (prioctl, nice)
- Control processor utilization (psrset)
- Adjust time slices (kernel tuning)

Solaris Configuration commands

(see UNIX Rosetta for other OSs)

- `psrinfo -v` - number and speed of CPUs
- `prtconf` - memory and device tree information
- `prtdiag` - CPU, Cache, Board, Memory, SBUS slot info
- `/etc/release` - Solaris HW release info
- `showrev -p` - Installed patch info
- `sysdef` - kernel parameter info
- `swap -l` - swap partition info
- `pkginfo` - installed packages

'sar' (System Activity Report)

available on most UNIX/Linux systems

- sar collects all performance data
- sar includes timestamps
- sar dumps to a binary file for review
- Examples
 - `sar -A -o sarfile 5 30 >/dev/null &`
 - Collect all data in background to file every 5 seconds 30 times and inhibit screen output
 - `sar -u -f sarfile` (Report CPU utilization from file)
- `sar -u 5 30` (Interactive version)

Top Ten

- Look for a disk bottleneck. More than 30% busy or 50 ms service time is a bad sign.
- When the customer says disks are no problem, insist on seeing `iostat -x` output
- After tuning other items, check disks again
- Use `nfsstat -m` to find a busy net or NFS server
- Don't worry about `vmstat` free RAM, it will not go above "lotsfree"

Top Ten (continued)

- Don't worry about pagein, pageout levels, all file I/O is done this way.
- Sustained high scan rates indicate a RAM shortage
- Run queue length $>4/\text{CPU}$ indicates CPU shortage
- If block procs = runnable procs, check again for slow disk
- If sys CPU time $>$ user time find out why. (Other than NFS servers)