System 성능 분석 보고서

(대상 장비: XXXX)

1 개요

㈜ XXX XXX의 주 데이터베이스 서버인 Ultra Enterprise 5000 시스템의 성능 진단 요청에 의해 다음과 같은 성능 진단을 수행 하였다. 이 문서는 현재 발생하고 있는 성능 상의 문제를 올바로 규명하고 그 해결책을 제시한다.

1.1 데이터 수집 기간

2002년 7월 9일 ~ 2002년 7월 15일 (1주일간)

1.2 수행 인원

차주현(XXX System Engineer)

1.3 진단 대상 서버 시스템

- 시스템 모델: Sun Ultra Enterprise 5000 (s/n: 651f08ae)
- 용도: Oracle RDBMS
- Hostname: kkdi
- OS: SunOS 5.6 generic_105181-08 sun4u
- CPUs: 168MHz UltraSparc-I (sparcv9) x 6
- Memory: 1GB
- 주 응용 프로그램: Oracle 7.3.4

1.4 진단 대상 서버 하드웨어 구성

- CPUs

Board	CPU ID	Module	Run MHz	E-cache(MB)	Impl.
0	0	0	168	1.0	US-I
0	1	1	168	1.0	US-I
2	4	0	168	1.0	US-I
2	5	1	168	1.0	US-I
5	10	0	168	1.0	US-I
5	11	1	168	1.0	US-I

- Memory

Board	Bank	MB	Status	Condition	Speed
0	0	256	Active	OK	60ns
0	1	256	Active	OK	60ns
2	0	256	Active	OK	60ns

5	0	256	Active	OK	60ns

- IO Cards

Board	Bus Type	Freq. MHz	Slot	Name	Model
1	SBus	25	0	Cgsix	TGX120,170-0006
1	SBus	25	1	QLGC,isp/sd (block)	QLGC,ISP1000
1	SBus	25	2	QLGC,isp/sd (block)	QLGC,ISP1000U
1	SBus	25	3	SUNW,hme	
1	SBus	25	3	SUNW,fas/sd (block)	
1	SBus	25	13	SUNW,soc	501-2069
3	SBus	25	0	QLGC,isp/sd (block)	QLGC,ISP1000
3	SBus	25	1	qec/qe (network)	SUNW,595-3198
3	SBus	25	2	HSI	SUNW,501-1725-01
3	SBus	25	3	SUNW,hme	
3	SBus	25	3	SUNW,fas/sd (block)	
3	SBus	25	13	SUNW,soc	501-2069

1.5 진단 영역

- 이 문서에 포함 되어 있는 내용
  - CPU 병목 진단
  - Memory Shortage 진단
  - I/O 병목 진단
  - 기타 시스템 파라메터 진단
  - 향후를 위한 제언
- 이 문서에 포함 되어 있지 않은 내용
  - 어플리케이션 성격의 분석
  - 오라클 성능

1.6 수집 데이터

다음의 각 명령들을 30초 간격으로 1주일 동안 실행한 결과

- vmstat: CPU, Memory 등 시스템 전반에 걸친 통계 자료
- iostat: 각 디스크들의 read/write 등에 대한 통계
- mpstat: 각 CPU 에 대한 사용량등에 대한 통계
- netstat: 네트웍 인터페이스 및 네트웍 전반에 대한 통계

다음의 기본 정보를 수집

- sar: System Activity Report 로 cron 에 의해 수행되는 시스템의 총괄적 통계

- ps: 시스템 상에서 운영 중인 프로세스들의 상황
- prtconf/prtdiag: 시스템의 하드웨어 설정에 대한 정보
- vmstat: 시스템 인터럽트 통계와 기타 통계
- netstat: TCP/IP 프로토콜에 대한 통계
- ipcs: 시스템의 IPC(Inter Process Communication) 설비에 대한 정보
- pmap: 시스템에서 운영 중인 프로세스 메모리 맵에 대한 정보

1.7 문서 요약

진단 대상 장비는 현재 몇몇(md0, md1, md2, md5)의 디스크에 과도한 I/O 부하가 (%b>90, svc_t>100ms) 걸려 있는 상태이며, 이로 인해 OS의 file buffer cache 또한 커져 있는 상태이다. 이 때문에 메모리 page scan이 더욱 심해지고 (sr > 200) 이것은 다시 I/O 증가의 원인이 되고 있다.

2 현상 분석

2.1 CPU

- key factor

factor	출처	설명	일반적 관리 기준
%sys	vmstat, sar	CPU 가 kernel mode 에서 동작 하고 있는 비율	< 30
%user	vmstat, sar	CPU 가 user mode 에서 동작 하고 있는 비율	< 70
%idle	vmstat, sar	CPU 가 아무것도 하고 있지 않은 상태의 비율[1]	> 10
%wio	vmstat, sar	CPU 가 disk 등의 I/O 가 완료 될길 기다리고 있는 비율	< 20
r	vmstat	실행 가능한 쓰레드가 CPU 의 run-queue 에서 대기하고 있는 프로세스의 수[2]	CPU 갯수 * 3[3]
B	Vmstat	Device 의 I/O 종료를 기다리고 있는 쓰레드의 수	< r

2.1.1 CPU 사용 현황

- 각 일자 별 평균 CPU 사용률

일자	%user	%sys	%wio	%idle
7 월 9 일	18	9	37	36
7 월 10 일	16	8	35	41
7 월 11 일	15	8	31	45
7 월 12 일	15	7	31	47
7 월 13 일	14	6	29	51
7 월 14 일	13	5	27	55

[1] vmstat에서는 %idle이 %wio를 포함하나 sar에서는 포함 되지 않는다. 즉 vmstat에서는 %idle = 100 – (%sys + %user)이고 sar에서는 %idle = 100 – (%sys + %user + %wio)이다.
[2] CPU는 CPU마다 각각의 run-queue가 있다.
[3] 만약 대기하고 있는 쓰레드가 CPU bound job이라면 CPU * 2의 수치도 높다. 즉 job의 성격에 따라 기준이 다르다.

7 월 15 일	22	9	31	39
평균	16.1	7.4	31.5	44.8

표 1 일자별 평균 CPU 사용률

표 1은 각 일자 별 CPU사용률을 보이고 있다. 표에서도 알 수 있듯이 CPU에 걸리는 부하는 적당 하거나 오히려 낮다고 할 수 있으나 %wio의 비율이 높은 것을 알 수 있다. %wio는 CPU가 디스크나 네트웍등의 I/O작업이 완료 되길 기다리고 있는 비율로 과도한 I/O작업으로 인해 CPU자원 중 많은 비율이 낭비 되고 있으며 따라서 전체적인 CPU 사용률이 낮아 지는 주 원인이 된다.

- 시간 대 별 CPU 사용률

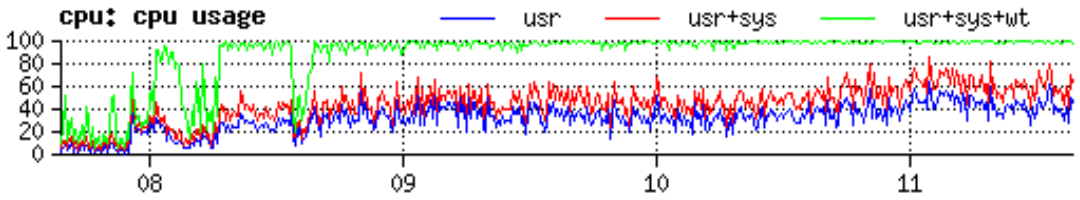


그림 1 7월 9일 mpstat

그림 1은 7월 9일 7시 30분부터 12시까지의 CPU사용률의 변화 이다. 업무가 시작되는 9시 전후로 CPU사용률이 급격히 증가 하는 것을 알 수 있고 이에 따라 %wio비율 역시 높아 지는 것을 알 수 있다.[4]

- 시간 대 별 Run-Queue 및 Wait-Queue

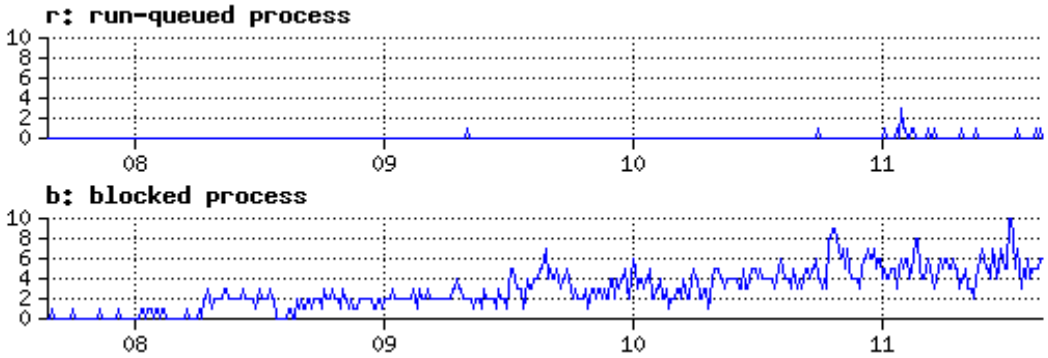


그림 2 7월 9일 vmstat

그림 2는 7월 9일 7시 30분부터 12시까지의 CPU Run-queue및 Wait-queue

[4] 전체적인 그래프는 지면 관계 온라인으로 조회 가능 하도록 한다.
(http://technet.hst.co.kr/~zoo11/parmlog/)

의 변화 이다. Run-queue는 전체적으로 낮게 유지 되고 있은나, I/O작업의 완료를 기다리고 있는 Wait-queue는 업무와 함께 급격 하게 높아 지는 것을 볼 수 있다.

- CPU 간의 부하 균형

```
CPU minf mjf xcal  intr ithr csw icsw migr smtx  srw syscl  usr sys wt idl
  0   22  35 2132  183  151 460   37   42   71    1  4834   20  10 61   9
  1   34  34 1753   47    1 470   45   46   39    0  5170   32  10 53   5
  4   33  37 3985  192  146 497   46   44   69    1  4598   23  12 58   6
  5   33  32 1555  317  100 396   38   41   48    0  6801   40   9 46   5
 10   19  28 2725   50    1 473   49   47   88    2  4677   31   9 53   6
 11   16  37 4822  201  146 510   61   55   52    1  5733   27  12 55   6
CPU minf mjf xcal  intr ithr csw icsw migr smtx  srw syscl  usr sys wt idl
  0    4  15 1528  131  115 320   17   25   19    0  1363   15   3 61  21
  1    3  24 5846   25    0 312   24   32   21    0  2866   20   5 57  18
  4    7  24 1606  161  136 316   23   30   24    0  1252   22   4 56  18
  5    3  29 1890  308  100 275   23   29   20    0  1686   33   4 49  13
 10    5  24 2720   29    0 335   28   34   22    0  1452   24   3 56  17
 11    4  26 2664  121   84 319   37   38   26    0  1813   32   5 50  13
CPU minf mjf xcal  intr ithr csw icsw migr smtx  srw syscl  usr sys wt idl
  0    4  13 2362  180  161 308   20   29   19    0  6359   16   4 61  19
  1   10  23 5330   30    0 318   29   34   20    0  3617   23   7 55  15
  4    8  17 4579  160  134 306   25   33   23    1  3290   20   6 58  17
  5   12  27 2999  310  100 261   28   31   21    0  5747   32   7 49  11
 10    5  20 2740   35    0 319   33   38   20    0  3315   22   4 59  15
 11    8  23 3086  175  134 275   41   37   19    0  5933   37   7 46  10
```

위의 결과는 7월 9일 업무 시작 후 부하가 많은 때인 10시경에 mpstat를 30초 간격으로 실행 한 결과 이다. Csw[5]값이 비교적 6개의 CPU간에 고르게 분포 함을 알 수 있다. 또한 user나 sys의 비율이 각 CPU사이에 고르게 분포 되어 있다.

- 주요 프로세스들

순위	PID	%CPU	command
1	5445	16.0	oracleKKDI (DESCRIPTION=(LO...
2	1924	13.4	ora_dbwr_KKDI
3	1924	12.4	ora_dbwr_KKDI
4	9809	11.9	ora_s001_KKDI
5	1924	11.6	ora_dbwr_KKDI

[5] Context Switch: CPU가 여러 쓰레드를 수행 하기 위해 쓰레드 사이의 CPU상태 변화의 수

6	1924	11.6	ora_dbwr_KKDI
7	23935	9.6	ora_s000_KKDI
8	1924	8.8	ora_dbwr_KKDI
9	1924	8.6	ora_dbwr_KKDI
10	26462	7.6	ora_s000_KKDI

표 2 CPU점유 순위

위의 표는 ps(/usr/ucb/ps) 명령어를 매 4시간 마다 실행 하여 나온 결과 중 CPU점유 상위 10개를 보여 준다. CPU점유 상위 프로세스들은 대부분 오라클 데이터베이스 서버 백그라운드 프로세스들이고 이 중에서도 I/O양이 많은 오라클 DB Writer 프로세스가 그 주를 이루고 있다. 즉 CPU bound job은 없고 대부분이 I/O에 집중 하는 프로세스이다.

2.1.2 CPU 성능 진단 결과

지금 까지 알아 본 바와 같이 시스템에서 CPU의 부하는 문제가 될 수준은 아니나 과중한 I/O job으로 인해 CPU의 대기 시간이 높다.

2.2 메모리

- Key factor[6]

Factor	출처	설명	일반적 관리 기준
Page in/out	vmstat, sar	메모리 page가 디스크로부터 읽히거나(in) 쓰여지는(out) 것	없음
Swap in/out	vmstat, sar	메모리 page가 스왑 디바이스로 부터 읽히거나(in) 쓰여지는 것(out)	없음
Sr	vmstat, sar	부족한 물리적 메모리를 확보 하기 위해 메모리 page를 scan하는 횟수	< 200
W	vmstat	Swap-out 되어 있는 프로세스의 수	< 1

[6] vmstat나 sar에서의 freemem은 메모리 부족을 판단 하는데 있어서 큰 의미가 없다. Freemem은 단순히 커널 파라메터 lotsfree에 근접하게 유지될 뿐이므로 항상 적은 값을 유지 하게 된다.

2.2.1 메모리 사용 현황

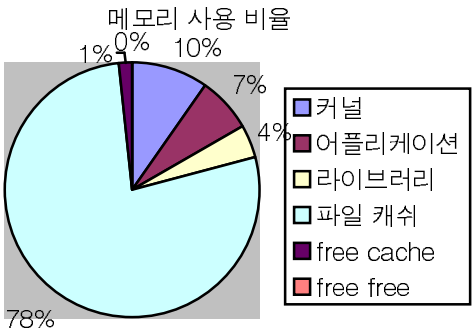
- 메모리 평균 사용량

구분	출처	사용량	설명
전체 메모리	Prtconf 등	968MB	시스템에 설치된 총 물리적 메모리
커널 메모리	Sar -k	95MB	커널이 사용 하고 있는 물리적 메모리[7]
어플리케이션 메모리	Pmap, memps	68MB	어플리케이션이 사용하고 private 메모리[8]
공유 라이브러리	Prtlib, memps –m	40MB	각 프로세스들이 공유 하는 라이브러리[9]
파일 캐쉬	-	754MB	파일 시스템의 내용의 캐쉬[10]

표 3 전체 메로리의 평균 사용량

위의 표와 옆의 그래프에서도 알 수 있듯이 현재 시스템 메모리의 대부분은 파일 시스템을 캐쉬 하기 위한 file buffer cache로 사용 되고 있다.

만약 메모리가 모자란 다면 파일 버퍼 캐쉬의 양 보다는 어플리케이션 메모리의 비율이 더 높아야 하므로 어플리케이션 메모리 부족은 아니다.



[7] 커널이 시스템 운영을 위해 필요한 자료 구조 등을 포함 하는 메모리. Page out되지 않고 항상 물리적 메모리에 존재한다. Sar –k명령으로 계산 된다.

[8] 메모리에 로드된 실행 파일과 그것의 부수적인 메모리들.text, data, heap, stack등. Memps명령으로 private영역을 더한 값.

[9] 각각의 실행 파일들이 메모리로 로드 될 때 널리 쓰여지는 라이브러리들은 다시 로드 되는 것이 아니라 현재 메모리 상에 존재 하는 라이브러리의 주소를 맵핑 할 뿐이다. 이로써 각 실행 파일들은 라이브러리를 공유 할 수 있다.libc, libsocket등이 그 예이다..

[10] 솔라리스는 메모리의 효율적 사용을 위해 최소한의 메모리(lotsfree)만을 남기고 나머지는 모두 파일 시스템의 파일들을 캐쉬하기 위한 용도로 사용 한다.

- Scan rate 와 page in/out

일자	Page in	Pages pg in	Page out	Pages pg out	Scan rate
7월 9일	106.26	261.78	22.55	31.19	619.37
7월 10일	114.75	272.54	15.68	25.07	651.99
7월 11일	99.27	303.59	13.68	22.14	735.68
7월 12일	91.52	206.47	15.40	22.41	475.42
7월 13일	64.68	166.95	13.39	18.81	387.29
7월 14일	46.37	117.82	13.08	16.95	273.62
7월 15일	100.37	261.90	18.98	27.14	575.56
평균	89.09	227.29	16.10	23.38	531.27

표 4 일자별 scan rate와 page in/out의 평균값

위의 표에서 가장 주목할 점은 높은 Scan rate이며 이는 시스템이 현재 메모리 부족에 처해 있거나 과도한 I/O로 인해 file buffer cache가 활발하게 사용 되고 있다는 것을 의미 한다. 현재 Scan rate는 평균값은 위의 표와 같지만 업무가 집중되는 시간에는 2000이상 최대 3900을 넘는다.(7월 10일) 또 하나 눈 여겨 보아야 할 것은 page in/out비율인데 측정 기간 동안 page in/out의 비율이 평균 9.72로 page out에 비해 in이 무척 높은 값을 가진다. page in이 일어 나는 경우는 1)어플리케이션을 메모리에 적재 할 경우 2)파일 시스템 상의 파일들을 file buffer cache에 적재 할 경우 3)swap 장치에서 swap out되었던 메모리 page를 다시 메모리에 불러 들이는 경우 등으로 나뉠 수 있다. 위의 경우처럼 실제로 page in된 page가 out보다 현저히 높을 경우, 대부분의 page in은 1)과 2)의 상황에서의 page in이다. 만약 메모리 부족 때문에 scan rate와 page in이 높게 나타난다면 이에 상응하는 page out과 swap메모리의 크기, swap장치의 디스크 활동률 또한 높아야 한다.

- Swap 메모리 사용 현황

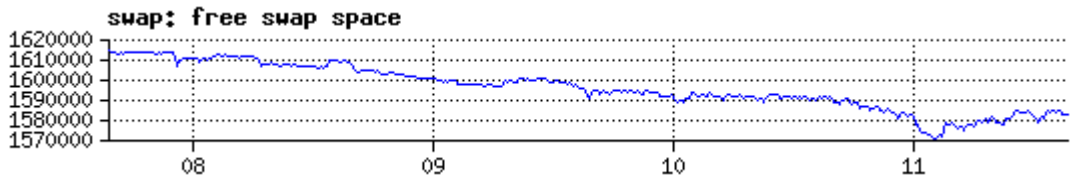


그림 3. 7월 9일의 swap 이용 변화(부분)

7월 9일 업무 시작 시간 경의 swap공간 이용의 변화 그래프를 보면 swap공간이 급격히 감소하는 듯 보이지만 위 그래프에서 가장 높은 값과 낮은 값의 차이는 10MB도 안되는 적은 차이를 보이므로 실제 하루 전체의 변화량은 거의 없다.

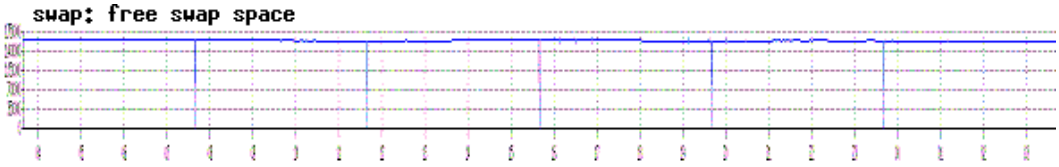


그림 4. 7월 9일 swap 이용 변화 (전체)

하루 전체의 swap사용률은 위 그래프에서 보는 것과 같이 변화가 거의 없다. 이 사실은 sar명령어의 w옵션으로도 확인 가능 하다.

2.2.2 메모리 진단 결과

현재 물리적 메모리는 현재의 어플리케이션을 운영하는데 부족함이 없어 보이나, 과도한 I/O로 인해 file buffer cache의 활동이 증가하고 이로 인한 page scan이 증가하여 더욱 I/O병목을 부추기고 있다.

2.3 I/O 부분

2.3.1 System 및 I/O Bus Configuration

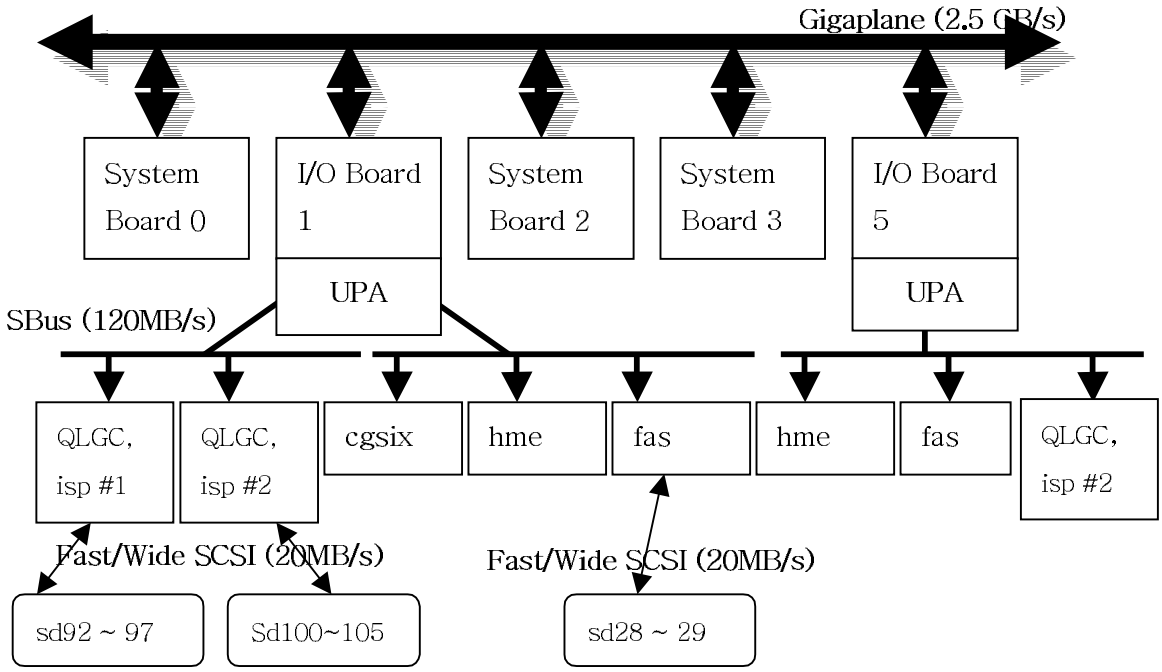


그림 5 시스템의 I/O 및 시스템 버스 구성

항목	최대 대역폭(throughput)	설명
Gigaplane	2.6GB/sec (2.5GB/sec)	UPA버스 사이를 연결 하는 시스템 버스
UPA	1.3GB/sec (1.2GB/sec)	UltraSPARC 프로세스를 위한 시스템 버스
SBus	90/120MB/sec	I/O 장치들을 위한 I/O버스
Fast/Wide SCSI-2	20MB/sec (8~10MB/sec)	Fast/Wide SCSI
Cgsix	8~20MB/sec	Turbo GX Graphic card
Hme	4MB/src	Fast Ethernet

표 5. 각 버스 및 I/O 장치 별 대역폭

그림 5는 현재 시스템의 시스템 버스 및 I/O버스의 구성을 나타내며, 표 5는 버스 및 장치 별 대역폭을 나타낸다. 위의 자료들을 근거로 계산 해 보면 현재 각 시스템 보드들을 연결 하는 Gigaplane의 대역폭(2.5GB/s)과 시스템 보드의 UPA 버스의 대역폭(1.2GB/s)은 부하 균형이 이루어 지고 있다. 또한 가장 I/O 활동이 왕성한 SCSI 디스크의 대부분이 I/O 보드 1번의 첫 번째 SBus에 연결 되어 있는 상태이다. 만약 SCSI 디스크가 최대 대역폭을 사용한다고 하면

Fast/Wide SCSI-2의 최대 대역폭 * scsi-2버스의 수
= 20MB/s * 2 = 40MB/s < 90/120MB (SBus의 최대 대역폭)

이므로 I/O 보드 1번의 첫번째 Sbus는 현재 병목이 없다. 그러나 각 디스크들의 활동량을 고려 하면 더 많은 SCSI 인터페이스를 추가하여 디스크들의 부하를 분배 해 주면 더욱 I/O병목이 감소 된다.

2.3.2 디스크 사용 현황

- Key Factor

Factor	출처	설명	일반적 관리 기준
Wait	Iostat	I/O 요청이 wait-queue[11]에서 대기 하고 있는 요청수	
%w	Iostat	Wait-queue 가 비어 있지 않은 비율	< 20

[11] OS의 장치 드라이버가 SBus와 SCSI버스를 통해 I/O요청을 전달 하기 전 대기 하는 Queue

Avwait	Sar	I/O 요청이 wait-queue 에서 대기하고 있는 평균 시간(ms)	
%b	Iostat	디스크가 현재 I/O 요청을 처리하고 있는 비율	< 65
Svc_t	Iostat	디스크가 실제로 I/O 요청을 처리하는 시간	< 100
Kw + Kr/s	Sar,iostat	초당 I/O 의 크기(Kbytes)	< 10MB

o 디스크 최대 사용량(괄호안은 최대 사용량은 일으킨 디스크)

일자	Rw/s	Krw/s	Wait	Svc_t	%w	%b
7월 9일	319.9	12856.0	0.5	300.7	1	100
7월 10일	322.2	10467.8	1.8	410.2	3	100
7월 11일	312.2	10688.7	11.1	325.2	8	100
7월 12일	328.0	10162.5	0.5	346.1	1	100
7월 13일	316.5	10180.2	0.5	361.7	1	100
7월 14일	258.2	8182.4	0.5	383.3	1	100

표 6. 일자별 디스크 최대 사용량

위의 표는 일자 별 디스크의 최대 사용량을 나타낸다. 전반적으로 문제가 되는 것은 11일의 wait값과 전체에 걸친 높은 svc_t, %b이다. Krw/s는 최대갑이 SCSI-2의 허용 범위인 20MB/s는 넘지 않고 있으나 typical throuput인 10MB엔 초과 하고 있다. 그러나 이 값은 DiskSuite으로 믂여진 메타 디스크이므로 개개의 디스크들은 SCSI-2버스의 허용치에 충분 하다.

Maximum values for each I/O device							
I/O-dev	rw/s	Krw/s	wait	actv	svc_t	%w	%b
total	1639.2	29987.7	1.0	28.0	971.2	1	1113
max	319.9	12856.0	0.5	6.5	300.7	1	100
md0	319.9	12856.0	0.0	6.5	131.9	0	100
md1	247.9	2034.5	0.0	4.4	76.7	0	100
md2	87.5	1745.8	0.0	0.7	193.1	0	71
md3	197.4	3389.0	0.0	2.0	152.7	0	97
md4	10.4	74.2	0.0	0.2	162.1	0	15
md5	177.1	3902.6	0.0	2.4	300.7	0	100
sd21	0.0	0.0	0.0	0.0	0.0	0	0

sd28	53.9	1846.0	0.5	1.3	219.5	1	57
sd29	48.3	641.6	0.5	0.8	104.3	0	47
sd92	62.1	721.1	0.0	0.7	32.8	0	52
sd93	61.1	711.9	0.0	0.7	41.3	0	52
sd94	183.4	2593.2	0.0	3.9	24.0	0	85
sd95	183.1	2565.7	0.0	3.7	27.8	0	83
sd96	182.6	2583.2	0.0	3.5	22.8	0	84
sd97	153.0	1710.8	0.0	3.4	33.1	0	91
sd100	184.1	2586.0	0.0	3.3	21.5	0	80
sd101	182.9	2572.3	0.0	3.0	21.8	0	81
sd102	154.4	1703.5	0.0	3.4	32.7	0	91
sd103	154.0	1701.9	0.0	3.5	25.1	0	90
sd104	61.3	713.3	0.0	0.7	18.8	0	50
sd105	0.0	0.0	0.0	0.0	0.2	0	0
st11	0.0	0.0	0.0	0.0	0.0	0	0
st12	3.2	3.2	0.0	0.0	2.4	0	1
st24	5.5	5.5	0.0	0.0	1.4	0	1
nfs1	0.0	0.0	0.0	0.0	10.5	0	0

표 7. 7월 9일의 각 디스크별 최대 사용량

위의 표는 Krw/s가 가장 높은 7월 9일의 각 디스크 별 통계이다. Sd92부터 sd105까지는 DiskSuite를 이용하여 md0부터 md5로 디스크들이 통합 되어 하나의 볼륨을 이루고 있다. 따라서 md0등의 통계는 그 볼륨이 포함하고 있는 개개의 SCSI 디스크들의 통계를 포함 하고 있다. 이 표에서 각 메타 디스크(md0~5)들의 사용률이 정상이 아님을 알 수 있다. %b는 몇 개를 제외하고 모두 100%에 근접하거나 100%에 다달았고 sct_t도 md1을 제외하고는 모두 100ms를 초과 하고 있다. 특히 md0는 초당 읽기/쓰기가 12MB를 넘어 서고 있어서 심한 I/O의 불균형을 보여 준다. Sd92부터 sd105까지의 개개의 SCSI 디스크들도 이에 따라 높은 수치를 보이고 있다.

- 시간 대 별 사용률

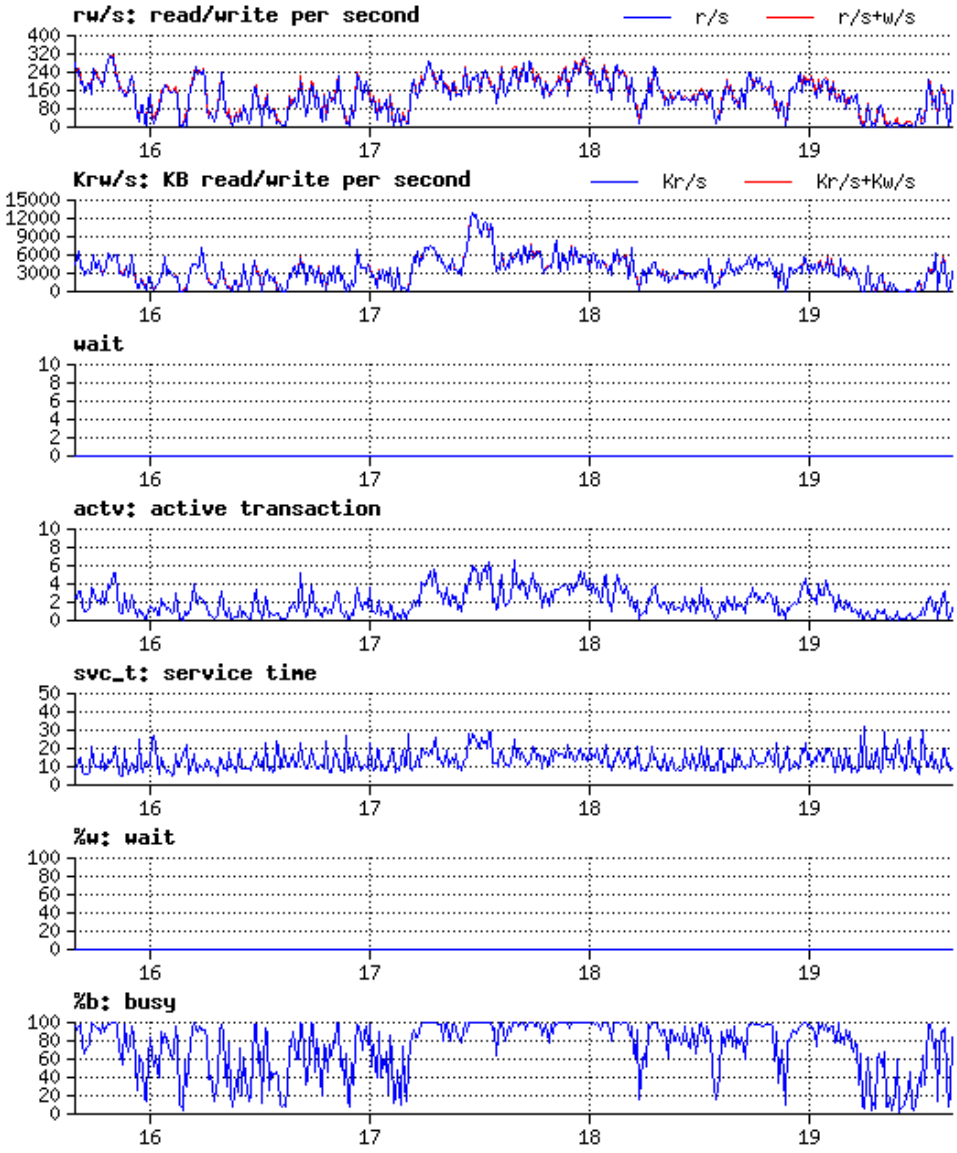


표 8. 7월 9일 15시 30분부터 19시 50분까지의 md0 사용률

위의 그래프는 7월 9일 md0의 krw/s가 가장 높은 시간대의 그래프이다. 역시 %b와 많은 I/O요청을 보인다. 또 읽기와 쓰기 횟수를 비교해 보면 읽기 횟수에 비해 쓰기 횟수는 거의 없음을 보여 준다. 즉 md0는 대다수의 SCSI요청이 읽기에 집중 되고 있다는 것을 보여주며 이 것은 OS상의 file buffer cache의 활동에 영향을 준다.

2.3.3 I/O 부분 진단 결과

I/O부분에서의 가장 큰 문제는 I/O의 부하 불균형에 있다. 특히 버스 부분에서는 I/O 보드 1의 첫번째 SBus에 모든 디스크들이 물려 있기 때문에 SBus의 병목을 일으킬 수 있고 또한 md0등 메타 디스크들은 거의 100%에 육박하는 busy time을 기록하고 있으며 특히 md0는 그 중에서도 가장 I/O활동이 활발하고 sct_t 또한 높은 상태이다.

2.4 Network 부분

2.4.1 Network Interface 활용 현황

o 일자별 최대 초당 Packet 수

일자	Input Packets	Output Packets	Total Packets	Throuput(KB/s)
7 월 9 일	1118	1119	2237	3355
7 월 10 일	785	1663	2448	3672
7 월 11 일	627	625	1252	1878
7 월 12 일	644	680	1324	1986
7 월 13 일	787	662	1449	2173
7 월 14 일	507	529	1036	1554

표 9. 일자 별 packet 수

위의 표는 일자 별 초당 최대 패킷의 수를 보여 준다. 이 시스템은 100M Fast 이더넷 인터페이스인 hme0를 갖고 있고 MTU[12]가 1500이므로 최대 트래픽은 보통 8.75MB/s(70Mbps)으로 계산 할 수 있으며 각 일자 별 트래픽은 7월 9,10일에 3MB/s를 넘었을 뿐 이다. 또한 input/output 양방향 모두 error는 발견 되지 않았고 collision도 0에 가까웠다

.

2.4.2 Network 부분 분석

Network부분에서는 아무런 문제점도 발견 하지 못 하였다.

2.5 기타 시스템 설정 부분

2.5.1 UFS buffer cache(:bufhwm)

buffer cache는 파일 시스템 상의 파일들에 대한 inode와 그에 따른 메타 데이터들을 캐쉬 한다. Sar -b로 모니터링 할 수 있으며 %rcache 항목이 95%이상을 유지 해야 한다. 현재 진단 대상 시스템은 %rcache가 100%를 유지 하고 있으므

[12] MTU(Maximum Transfer Unit): 이더넷 프레임의 최대 크기

로 현재 크기인 20471808Byte는 적당하다.

2.5.2 DNLC (Directory Name Lookup Cache:ncsize)

DNLC는 UFS와 NFS상에서 디렉토리나 파일의 이름과 해당하는 vnode를 캐쉬 하고 있는 부분이다. 이 것은 sar -a 명령의 namei/s과 iget/s의 비율로 측정 할 수 있다. 또한 vmstat -s로 캐쉬 hit rates를 알 수 있다. Namei/s는 초당 DNLC 요청 횟수 이고 iget/s는 DNLC miss로 인한 실제 디스크 읽기 횟수이다. 대상 시스템은 현재 iget/s가 측정 기간 동안 0을 기록 하고 있으므로 DNLC는 조정 할 필요가 없다.

```
...
2446292407 system calls
1269029651 total name lookups (cache hits 97%)
141999 toolong
...
```

2.5.3 Inode cache(ufs_ninode)

Inode cache는 현재 열려 있거나 최근에 열렸던 파일의 inode를 캐쉬 한다. 이 값은 netstat -k명령어로 알 수 있고 이때 inode_cache항목의 maxsize가 그 크기를 보여 준다. 또한 maxsize reached부분은 inode cache를 초과한 요청 횟수를 보여 준다. Sar -g의 %ufs_ipf의 값이 항상 0이어야 한다. 이 값은 DNLC의 값과 같아야 한다.

3 결론 및 성능 향상을 위한 제언

2장에서 기술 했듯이 진단 대상 시스템은 크게 다음과 같은 문제를 포함 한다..

- 오라클 DB 의 활동에 의한 파일 시스템의 과도한 I/O 부하
- 이로 인한 OS 상의 file buffer cache 동작
- file buffer cache 메모리를 위한 page scanner 의 과도한 동작(2000/s)
- page scanner 로 인한 I/O 가중

위와 같은 문제를 해결 하는 데 있어서 가장 목표로 두어야 할 것은 I/O의 최대한 줄이는 것이며 이는 다음과 같은 방법을 생각 해 볼 수 있다. 평균 목표 수치는 Scran rate는 100이하 %b는 65이하 이다.

- 오라클 DBMS 의 튜닝
  - 오라클 DBMS 의 어플리케이션(Stored procedure, PL/SQL optimization)부분을 튜닝 함으로써 I/O 횟수를 최소화 한다. 이 부분의 개선이 가장 큰 성능 향상을 볼 수 있는 부분 이다.
- 오라클 SGA 를 늘림
  - 현재 오라클의 SGA 는 500MB 정도가 공유 메모리로써 사용 되고 있다. SGA 중 테이블 스페이스의 내용이 캐쉬되는 database buffer pool 을 더 늘리고 ISM 으로 선언하여 page out 되지 않게 하여 더 많은 데이터를 캐쉬 할 수 있도록 하여 디스크 I/O 를 줄인다. 현재 시스템의 메모리가 약 1GB 정도 되지만 1GB 의 메모리를 추가 장착 하여 SGA 를 1.5GB(전체 메모리의 70%)정도 할당하는 것이 바람직하다.
- 오라클 테이블 스페이스의 물리적 재배치
  - 오라클 테이블 스페이스가 위치한 파티션에 I/O 가 집중 되고 있기 때문에 그 쪽 파티션에 새로운 SCSI 컨트롤러와 디스크를 추가해 새로운 볼륨을 구성하여 디스크 부하를 분산 시킨다.
- 오라클 테이블 스페이스 파티션을 Direct I/O 로 이용
  - 현재 과도한 file system buffer cache 의 이용으로 인해 오히려 더욱 I/O 부하가 가중 되고 있는 상황이므로 오라클 테이블 스페이스가 위치한 파일 시스템을 OS 상의 버퍼를 거치지 않고 바로 장치 드라이버로 I/O 할 수 있도록 Direct I/O 를 이용 한다. 대부분의 RDBMS 가 그렇듯이 오라클 또한 DBMS 내부에서 이미 데이터를 캐쉬하므로 OS 상의 버퍼링은 그 만큼 overhead 로 존재 한다.
- OS Upgrade

- 솔라리스 8 부터 새로운 메모리 관리 기법이 default 로 적용 되므로 새로운 OS 로 업그래이드 하는 것이 바람직하다. 또한 솔라리스 7 부터 적용되는 파일 시스템 Logging 을 이용하여 파일 시스템의 I/O 를 줄일 수 있다.

o Priority paging 적용

- 기존 OS 에 패치를 적용하여 새로운 메모리 관리 기법인 Priority paging 을 적용한다. Priority paging 은 file buffer cache 와 어플리케이션 메모리의 page out 을 구분하여 함으로써 Scan rate 를 줄일 수 있다.

o UFS Write throttle 활성화

- UFS 파일 시스템이 쓰기를 시도 할 때 현재 디스크가 사용 중 이면 쓸 내용을 메모리에 적재하고 대기 한다. 이 것이 파일 하나 당 384KB 를 넘어서면 이 크기가 줄어 들 때까지 전체 프로세스가 멈추는 데 이 크기를 조절 함으로써 메모리와 I/O 성능을 높일 수 있다.

다음 표는 위에 기술한 방법들에 대한 구현 난이도와 영향도를 상,중,하로 나뉘어 보여준다. (효과는 추정임)

항목	구현 난이도	효과
오라클 튜닝(어플리케이션)	상	상
오라클 SGA추가	중	중
오라클 파티션 재배치	상	중
오라클 파티션 Direct I/O	하	상
OS Upgrade	중	상
Priority paging	하	상
UFS 튜닝	중	중

4 참고 문헌

- SA-400 Solaris System Performance Management, Sun Microsystems.
- Brian L. Wong, Configuration and Capacity Planning for Solaris Servers, Prentice Hall.
- Adrian Cockcroft, Sun Performance and Tuning 2nd Edition, Prentice Hall.
- Various articles from SunSite.