

Prática

Levantamento e diagnóstico de ambiente

1) Analisar o sistema operacional

Faça um levantamento dos recursos disponíveis

Memória

Free -m

CPU

Top

IO

yum install hdparm

hdparm -tT /dev/sda

Quais comandos e aplicativos utilizar?

Vmstat

```
-----  
---  
DESCRICAO DAS PRINCIPAIS COLUNAS DE SAIDA DO VMSTAT  
-----  
---  
*   r           Numero de processos aguardando execucao  
*   b           Numero de processos nao interrompiveis em  
espera  
    swpd        Qtde. de memoria virtual em kb
```

```

    free      Qtde. de memoria livre em kb
    buff      Qtde. de buffer de memoria em kb
    cache     Qtde. de cache de memoria em kb
*   si        Qtde de memoria carregada (in) em swap
(disk/second)
*   so        Qtde de memoria gravada (out) em swap
(disk/second)
    bi        Blocos lidos por segundo
    bo        Blocks escritos por segundo
    us        Percentual de tempo de cpu usada por
processos de usuario
    sy        Percentual de tempo de cpu usada pelo
sistema
*   id        Percentual de tempo de cpu ociosa
*   wa        Tempo de cpu esperando por I/O

```

```

-----
---
```

```

-----
---
```

IDENTIFICANDO PROBLEMAS

```

-----
---
```

- 1- o valor de "r" nao pode ser maior que qtde de cpus ou nucleos do servidor
- 2- se "b" (processes sleeping) eh consistentemente maior que 0 entao nao ha CPU suficiente para executar todos os processos existentes.
- 3- se "so" (swap out) and "si" (swap in) sao consistentemente maior que 0, ha pouca memoria
- 4- se "id" eh consistentemente igual a 0 entao nao ha CPU suficiente.
- 5- "wa" alto indica contencao de I/O no disco/storage

lstat

Top ou htop

-- teclas de atalho do top:

 - d para mudar tempo de
refresh:

- barra de espacos atualizar refresh instantaneamente
- B negrito para principais valores
- h ver help
- l para ver consumo de cpu "por cpu" (e nao total geral)
- c para ver detalhes dos nomes/caminhos dos processos

LISTAR TOP 10 CPU:

```
> ps -e -o pcpu,pid,user,tty,args | sort -n -k 1 -r | head
```

LISTAR TOP 10 MEMORY:

```
> ps -e -o pmem,pid,user,tty,args | sort -n -k 1 -r | head
```

2) Analisando o BD

-- Verificar maiores wait events das sessoes atuais

```
SELECT      EVENT,
            (sum(time_waited) / 100)
total_time_waited_sec,
            SUM(TOTAL_WAITS) total_waits
FROM        V$SESSION_EVENT
WHERE       EVENT NOT IN
            ('SQL*Net message from
client', 'SQL*Net message to client', 'pmon timer',
'smon timer',
            'rdbms ipc message', 'jobq
slave wait', 'rdbms ipc reply', 'i/o slave wait', 'PX
Deq: Execution Msg')
GROUP BY    event
order by    total_time_waited_sec desc;
```

Mostra top queries com pior desempenho

```
select b.username username, a.disk_reads reads,
```

```

a.executions exec, a.disk_reads /decode
(a.executions, 0, 1,a.executions) rds_exec_ratio,
a.sql_text Statement
from V$sqlarea a, dba_users b
where a.parsing_user_id = b.user_id
and a.disk_reads > 100000
order by a.disk_reads desc;

```

```

-- ver top events nos ultimos 30 minutos (linha com
valor NULL eh o tempo que a sessao nao teve wait time
-- p/ consultar a V$ACTIVE_SESSION_HISTORY precisa
ter a licenca da option "Diagnostics Pack"

```

```

SELECT      EVENT,
            sum(WAIT_TIME) / 1000000 total_cpu_sec,
            sum(TIME_WAITED) / 1000000 total_wait_sec,
            SUM(WAIT_TIME + TIME_WAITED) / 1000000
TOTAL_sec
FROM        V$ACTIVE_SESSION_HISTORY
WHERE       SAMPLE_TIME > to_timestamp(sysdate -
30/60/24)
GROUP BY   EVENT
order by   TOTAL_sec desc;

```

```

-- VER top WAIT EVENTS POR CLASSE
select      wait_class, sum(time_waited),
sum(time_waited)/sum(total_waits)
            sum_waits
FROM        V$SYSTEM_WAIT_CLASS
GROUP BY   WAIT_CLASS
order by   3 desc;

```

```

-- ver TOP WAIT EVENTS POR Application E Concurrency
select      a.event, a.total_waits, a.time_waited,
a.average_wait
from        v$system_event a, v$event_name b,
v$system_wait_class c
where       a.event_id=b.event_id
and         b.wait_class#=c.wait_class#
AND         C.WAIT_CLASS IN
('Application','Concurrency')
order by   average_wait desc;

```

-- VER TEMPO TOTAL DE WAITs E DE CPU ULTIMOS EVENTOS.
Aqui da para ver se BD gastou mais tempo com wait
events ou cpu e da p/ ver detalhes de onde foi gasto
o tempo

```
SELECT  METRIC_NAME, VALUE, metric_unit
FROM    V$SYSMETRIC
WHERE   METRIC_NAME IN ('Database CPU Time
Ratio','Database Wait Time Ratio')
AND     INTSIZE_CSEC = (select max(INTSIZE_CSEC) from
V$SYSMETRIC);
```

– Principais consumidores de CPU

```
select
    ss.username,
    se.SID,
    VALUE/100 cpu_usage_seconds
from
    v$session ss,
    v$sesstat se,
    v$statname sn
where
    se.STATISTIC# = sn.STATISTIC#
and
    NAME like '%CPU used by this session%'
and
    se.SID = ss.SID
and
    ss.status='ACTIVE'
and
    ss.username is not null
order by VALUE desc;
```