

# **CPU Usage, Isolation, and Reduction in a Oracle Database Virtual Environment**

## **Session 322**

Michael Zechman  
Database Consultant  
Convertible Technology, Inc.



# Introduction

- **Goal - To share what we have learned in our pursuit to reduce CPU cycle usage**
- **Beneficial for Virtual Environment**
  - **Shares CPU and Memory**
- **Can be used in any environment**
- **My Customers - Service Provider(s)**
- **History of why and how we got to this point**



# Session Agenda

- **Why reduce CPU cycle usage?**
- **Targeting CPU Usage and Statistics**
- **Linux Resource Consumption Tips**
- **Oracle Resource Consumption Tips**
- **Items under review**
- **Comments/Questions**



# Session Agenda

- **Why reduce CPU?**
- **Targeting CPU Usage**
- **Linux Resource Consumption Tips**
- **Oracle Resource Consumption Tips**
- **Items under review**
- **Comments/Questions**



# Why Reduce CPU Usage ?

- **It's all about \$'s - SAVE MONEY!**
- **Helps Performance**
- **For Service Provider (data center)**
- **For Customers**
- **z/VM manages SUSE Linux Virtual Servers**



# Service Provider

- Reduce the number of required CPU's
- CPU Physical Cost factor
- Hardware costs
- Software cost/licensing
- Smaller Footprint in Data Center
- **COST – Potential hundred's of thousand's \$ per CPU**



# Customer

- Customer billed by CPU cycle usage
- Customers share Software cost
- Reduce CPU Usage for databases and applications
- Keeps you honest and system tuned
- Some apps are not honest!
- Wide variety of simple and complex solutions



# Session Agenda

- **Why reduce CPU?**
- **Targeting CPU Usage**
- **Linux Resource Consumption Tips**
- **Oracle Resource Consumption Tips**
- **Items under review**
- **Comments/Questions**





# Targeting CPU Usage

- **Make sure Oracle database is tuned/goals!**
  - **Requires regular database monitoring.**
- **Monitor Processes at the Server level**
- **Target Processes at Server level to reduce CPU usage on the server**
- **Target Processes at the Database level to reduce CPU**



# Process CPU Usage

- **Gather INDIVIDUAL PROCESS CPU usage statistics (all processes)**
- **Variety of tools to gather statistics**
- **Some Free, some not**
- **We use Velocity software at the z/VM level**
- **Keep CPU usage for each process at 5 minute intervals**



# Process CPU Usage

- **Summarize Top 5 or 10 CPU usage processes by each Server for an hour, every hour, for each day.**
- **Remainder of process - OTHER**
- **Programmed reports – BEING REPLACED**
- **Loaded Statistics into Oracle Tables**
- **Created standard SQL's to view statistics**



# Process Reports

- 11g SELECT PIVOT works great!
- SUM, GROUP BY, ROLLUP
- Target high CPU usage processes
- Surprised at what we FOUND!
- Compare production systems
- Compare test systems
- Compare OS and Database Upgrades



# Some Processes

- **ZIP, GZIP, BZIP, etc.**
- **SNMPD**
- **OEM Agent**
- **KSWAPD0**
- **BPBKAR**
- **GREP**
- **TAR**



# Some Processes

- **Security Monitoring Scripts**
- **Perl**
- **RMAN**
- **PYTHON**
- **JAVA**



# Some Processes

- **Oracle was biggest consumer**
- **Other processes combined were anywhere between 20 to 90% of the CPU usage on a server**
- **Test Servers – Oracle was less**
- **Production Servers – Oracle was more**
- **Administrative Tasks Use CPU Cycles**



# Sample Daily Load

- CREATE TABLE ext\_stat\_top5\_hourly org external ...;
- CREATE TABLE stat\_top5\_hourly (  
hostname char(8),  
stat\_date date,  
hour\_hh char(2),  
process\_name char(8),  
cpu\_seconds number(8,2) ) tablespace stats\_data;
- INSERT INTO stat\_top5\_hourly from ext\_  
stat\_top5\_hourly;
- COMMIT;





# Sample SQL's

```
define START_DATE = 20-MAR-2013
define END_DATE   = 30-MAR-2013
select * from
(select stat_date, hostname, cpu_seconds
 from loz_stat_top5_hourly
 where stat_date >= to_date('&start_date','DD-MON-YYYY')
 and
      stat_date <= to_date('&end_date','DD-MON-YYYY') )
PIVOT (sum(cpu_seconds)
 for hostname in ('SERVRZ03','SERVRZ10','SERVRZ11'))
order by 1;
```



# Sample SQL's

STAT_DATE	'SERVRZ03'	'SERVRZ10'	'SERVRZ11'
20-MAR-13	16873.36	16776.53	35594.55
21-MAR-13	18279.45	13447.69	28611.39
22-MAR-13	16596.94	17115.24	23374.67
23-MAR-13	14609.62	13667.23	31877.07
24-MAR-13	12495.09	11216.99	61232.52
25-MAR-13	27112.06	14722.32	13298.33
26-MAR-13	23234.74	26387.51	90735.40
27-MAR-13	25606.14	17155.42	22731.16
28-MAR-13	20304.94	18697.91	22089.23
29-MAR-13	18598.43	14744.03	31206.50
30-MAR-13	9662.79	7082.97	24973.24



# Sample SQL's

```
define START_DATE = 25-MAR-2013
define END_DATE   = 25-MAR-2013
select * from
(select process_name, hostname, cpu_seconds
 from loz_stat_top5_hourly
 where stat_date >= to_date('&start_date','DD-MON-YYYY') and
        stat_date <= to_date('&end_date','DD-MON-YYYY')
 )
 PIVOT (sum(cpu_seconds)
 for hostname in ('SERVRZ03'))
 order by 1;
```



# Sample SQL's

```
Process 'SERVRZ03'
```

```
-----  
BPBKAR          373.08  
CONVERSI        17.11  
CRON            376.37  
EXTPROC         151.9  
FIND            45.25  
GREP            410.73  
GZIP            3042.57  
INIT            1845.39  
KSWAPD0         22.96  
MILLES          28.05  
NotAccnt        103.34  
ORACLE          19522.37  
OTHER           807.53  
SH              19.23  
SNMPD           269.57  
SQLPLUS         46.62  
XARGS           30.53
```



# Session Agenda

- Why reduce CPU?
- Targeting CPU Usage
- **Linux Resource Consumption Tips**
- Oracle Resource Consumption Tips
- Items under review
- Comments/Questions



---

# Reduce Linux RPM's that are Installed

- Helps reduces the amount of Disk space used
- Reduces the Number of Linux services that get created
- Reduce the software updates/bug/security updates that are required

---

# Reduce Linux RPM's that are Installed

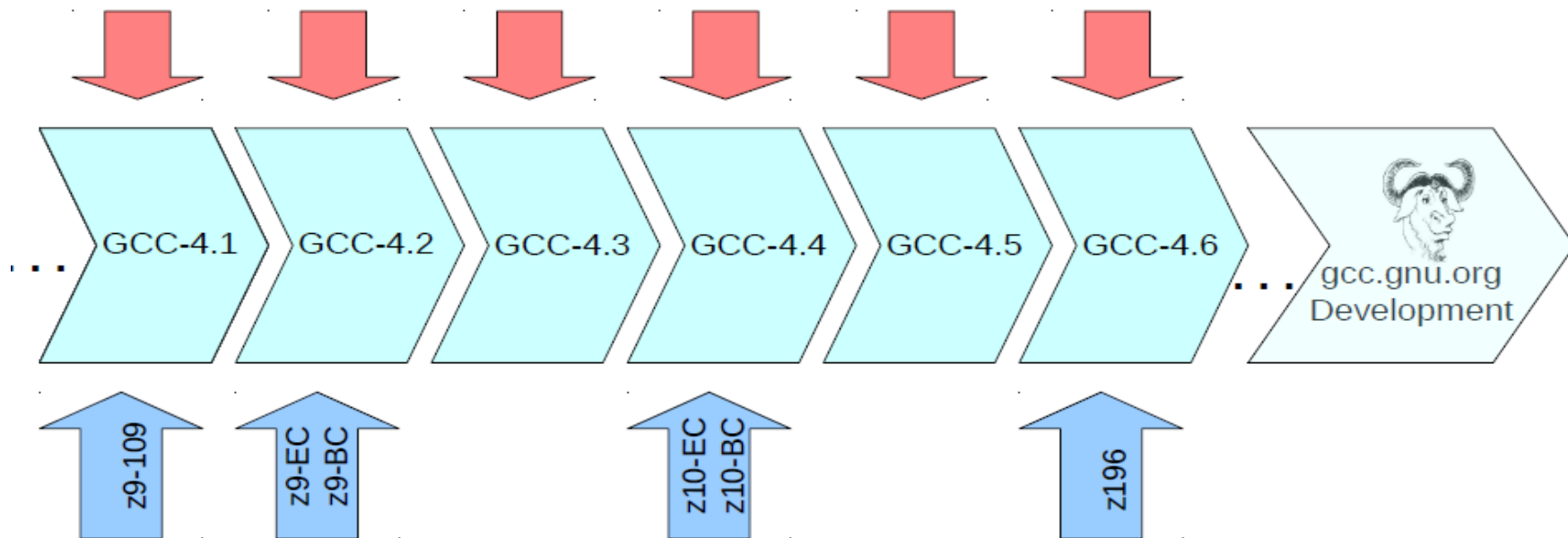
- Helps Reduce problems seen by Oracle Support
  - Too many rpms can cause problems
  - Multiple gcc / g++ Versions in Linux MOS Note: 444084.1
  - May be needed if supporting multiple Oracle Versions



# GCC Compiler Evolution

## GCC compiler evolution

IBM development provides patches to exploit new GCC features also in Linux on System z (i. e. software DFP, hardware optimized instruction scheduling)



IBM development provides patches to exploit IBM System z and IBM zEnterprise hardware features in new GCC versions (i. e. new instructions, hardware DFP)





---

# Turning off any Unneeded Services List

- Keep the golden image as lean as possible in terms of processor usage, some of these services can be turned off with chkconfig command:

```
chkconfig fbset off  
chkconfig work-remotefs off  
chkconfig postfix off  
chkconfig splash off  
chkconfig splash_early off  
chkconfig smartd off
```



---

# Linux RPM considerations

- **Staying current is extremely important**

- glibc performance improvements with each release
- gettimeofday() – several vendor improvements

**V**irtual **D**ynamically-linked **S**hared **O**bject (**VDSO**) is a shared library provided by the kernel. This allows normal programs to do certain system calls without the usual overhead of system calls like switching address spaces.

On a z196 system for example by using the VDSO implementation **six times** reduction in the function calls are possible.

Newer Linux distributions (RHEL 6, SLES 11) have this feature and it's enabled by default.



---

# Additional CPU Savers \$\$\$\$

- Consider Using Linux Huge Pages for Oracle Database Memory
- → In general 10-15% can be gained by the reduction in CPU usage as well as having a lot more memory for applications that would be consumed in Linux Page Tables...

# Networking Performance Considerations (1)

- For Oracle RAC Environments, **choose your MTU size carefully**. Set it to the maximum size supported by all hops on the path to the final destination to avoid fragmentation.
  - Use the **tracpath** command to verify the path MTU size.
  - If the application sends data in chunks of  $\leq 1400$  bytes, use an MTU size of 1492:
    - 1400 bytes user data plus protocol overhead.
  - If the application is able to send bigger chunks, use an MTU size of 8992.
  - Sending packets  $> 1400$  with an MTU size of 8992 will increase throughput and save CPU cycles.
- For HiperSockets, select an MTU size to suit the workload:
  - If the application is capable of sending large packets, a larger MTU size will increase throughput and decrease cpu cycles.
  - An MTU size of 56K is recommended only for data streaming workloads with packets  $> 32KB$ .

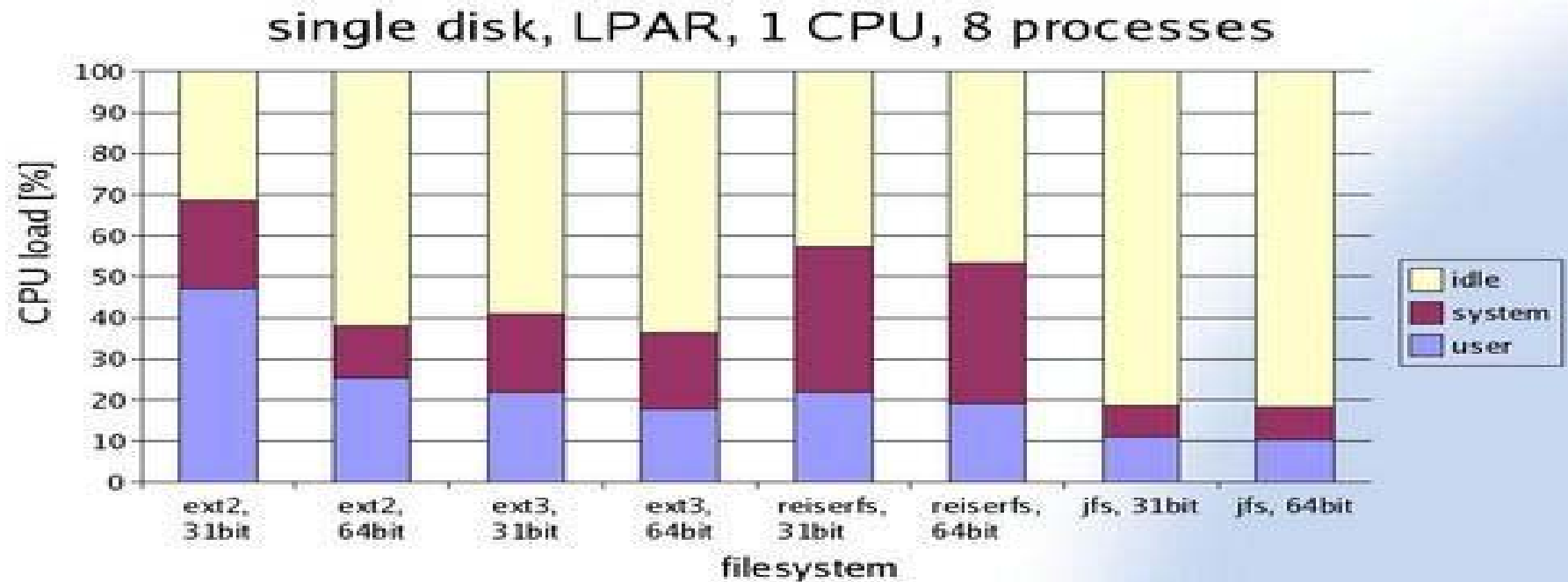


# File System Options

- EXT2 - most widespread Linux file system.
- EXT3 - evolved from ext2, adds journaling features.
- EXT4 –**only supported with RH 5.6/OL 5.6/SLES 11 or greater**  
**(recommend to test first)**
- JFS - a port of OS/2 Warp Server jfs to Linux.
- Reiserfs – journaling behavior is comparable to ext3 in order mode.
- **Recommend using ext3 or ext4 due** its journaling capabilities and reduced cpu load
- See latest performance report at:  
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/linux390/perf/ZSW03027-USEN-00.pdf>



# File System Options



IBM



**COLLABORATE<sup>13</sup>**  
TECHNOLOGY AND APPLICATIONS FORUM  
FOR THE ORACLE COMMUNITY

# Kernel I/O Scheduler

- The I/O scheduler optimizes disk access, the strategy for optimization aims to minimize the number of I/O operations and disk head movements.
- The Linux 2.6 kernel offers a choice of four different I/O schedulers:
  - Noop Scheduler (noop)
  - Deadline Scheduler (deadline)
  - Anticipatory Scheduler (as)
  - Complete Fair Queuing Scheduler (cfq)
- Linux default is the “as” scheduler:
  - Designed to optimize access to physical disks.
  - Not suitable for typical storage servers used in the System z environment
  - Selected by setting the “elevator” boot parameter in /etc/zipl.conf
- Recommended I/O scheduler – **deadline or noop**



---

## Misc. Linux Tip

- Disk is cheaper than CPU – Most of the time
- Don't use repeated GZIP -c options
- Better to concatenate first if you must
- STOP Services, Including databases not in use
- Automated tools for customers to start and stop databases under Provide control
- Be aware of stopping DB's and keeping Listener up. Connects create process that does not connect to DB – Uses CPU!





# Session Agenda

- **Why reduce CPU?**
- **Targeting CPU Usage**
- **Linux Resource Consumption Tips**
- **Oracle Resource Consumption Tips**
- **Items under review**
- **Comments/Questions**



---

# Upgrade to the latest Release

## 11.2.0.3

- Oracle's **VKTM** process uses slightly less CPU minutes
  - (about **0.08** vs. 0.09 with 11.2.0.2)
- Great improvements with **ora\_dia0** process.
  - (about **0.07** sec cpu/minute vs. **0.28** with 11.2.0.2)
- Only Install the database modules that are needed
  - DB installed with **NO** options
    - The "gettimeofday" function is called **300 times every 15 seconds**.
  - DB installed with **all** options : (java, xml, Text, spatial, APEX, etc ..... )
    - The "gettimeofday" function is called **1500 times every 15 seconds**.



---

## Other Database CPU Reducing Parameters (at a cost)

- Review the v\$scheduled\_jobs in particular the sql tuning advisor and segment advisor to see the best usage for your environment – DAILY Jobs.
- RMAN Backups – Consider using Cumulative Merged Backups (RMAN Level 1 merged with Level 0) and or Oracle’s Advanced Compression.
- Perhaps not use RMAN at all



# RMAN Compression

Backup Compression	Backup Time	Compression Size Source DB - 1.29 GB	% Compression / Input MB/s
'Basic' 10gR2 (BZIP2) Compression	02:48 (168 s)	278.95 MB	78.9 % 7.89 MB/s
'High' 11gR2 (BZIP2) Compression	08:41 (521 s)	224.82 MB	83.0 % 2.54 MB/s
'Medium' (ZLIB) Compression	01:08 (68 s)	295.53 MB	77.6 % 19.46 MB/s
'Low' (LZO) Compression	00:28 (28 s)	357.03 MB	73.0 % 47.26 MB/s



# RMAN Compression

- RMAN Command -> **CONFIGURE COMPRESSION ALGORITHM 'Low'**
- **Oracle Advanced Compression Feature required for Low, Medium, High**
- **Very High CPU observed with BZIP2**
- About 1/7 CPU used on 'MEDIUM' versus Default Compression mode**



## Other Database CPU Reducing Parameters (at a cost)

- **filesystemio\_options=setall** for databases with file systems to reduce caching to Linux File system cache memory.
  - **disk\_asynch\_io=true**
- If seeing “**resmgr:cpu quantum**” wait events may want to disable the resource manager default plan
  - **resource\_manager\_plan = "**



---

# resmgr:cpu quantum Wait Event

**Additionally You need disable the Maintenance Window Resource Plan**

```
select window_name,RESOURCE_PLAN  
from DBA_SCHEDULER_WINDOWS;
```

**WINDOW\_NAME**

-----  
**MONDAY\_WINDOW**

**RESOURCE\_PLAN**

-----  
**DEFAULT\_MAINTENANCE\_PLAN**

**execute**

```
dbms_scheduler.set_attribute('MONDAY_WINDOW','RESOURCE_  
PLAN',");
```

**WINDOW\_NAME**

-----  
**MONDAY\_WINDOW**

**RESOURCE\_PLAN**

-----



# Investigate Locking Table Statistics for Some Tables

- **TRUNCATE** and **SQLLDLDR LOAD** is an Example:
- **DBMS\_STATS.UNLOCK\_TABLE\_STATS**(ownname => 'USERS', tabname => 'XXX');
- **DBMS\_STATS.GATHER\_TABLE\_STATS**(ownname => 'USERS ', tabname => ' XXX', estimate\_percent=>1, cascade =>TRUE, degree =>4);
- **DBMS\_STATS.LOCK\_TABLE\_STATS**(ownname => 'USERS', tabname => 'XXX');





# Session Agenda

- **Why reduce CPU?**
- **Targeting CPU Usage**
- **Linux Resource Consumption Tips**
- **Oracle Resource Consumption Tips**
- **Items under review**
- **Comments/Questions**





---

# ITEMS UNDER REVIEW

- **Oracle GRID Agent**
- **How To Effectively Investigate & Diagnose Grid Control Agent High CPU Utilization Issues? [ID 1089898.1]**
- **VKTM**



# ITEMS UNDER REVIEW

- SUSE10

```
ps -ef | grep vktm
```

```
oracle 18095 1 0 08:00 ? 00:00:09 ora_vktm_O1XX
```

```
oracle 18152 1 0 08:00 ? 00:00:09 ora_vktm_O2XX
```

- SUSE 11

```
ps -ef | grep vktm
```

```
oracle 8561 1 0 08:00 ? 00:01:18 ora_vktm_O1XX
```

```
oracle 8699 1 0 08:00 ? 00:01:18 ora_vktm_O2XX
```



# ITEMS UNDER REVIEW

## OLD SYSTEM (SUSE 10)

```
ps -ef | grep vktm
oracle 1534 1 0 08:00 ? 00:00:08 ora_vktm_OXXX
oracle 1599 1 0 08:00 ? 00:00:08 ora_vktm_OXXX
```

- home/oracle> strace -cp 1534  
Process 1534 attached - interrupt to quit Process 1534 detached

% time	seconds	usecs/call	calls	errors	syscall
99.21	0.174249	11	16455		nanosleep
0.79	0.001393	0	33214		gettimeofday
100.00	0.175642		49669		total

## NEW SYSTEM 1 (SUSE 11)

```
ps -ef | grep vktm
oracle 4030 1 0 10:29 ? 00:00:00 ora_vktm_OXXX
oracle 4212 3957 0 10:30 pts/1 00:00:00 grep vktm
oracle(0140):/home/oracle> strace -cp 4030 Process 4030 attached - i
```

% time	seconds	usecs/call	calls	errors	syscall
100.00	1.520628	7	218891		nanosleep
0.00	0.000004	4	1		restart_syscall
100.00	1.520632		218892		total



---

## ITEMS UNDER REVIEW

- SUSE 10

kernel timer interrupt frequency is approx. 100 Hz

- SUSE 11

kernel timer interrupt frequency is approx. 4000 Hz  
or higher



---

## ITEMS UNDER REVIEW

- Multi-treaded EXTPROC (Agent)
- Native Compiled PL/SQL Procedures
- Compression for historical data
- Advanced Security for Security and Encrypted Exports – EXPORT ENCRYPTION versus double the work



Questions?  
Comments?





# Summary

- **We have discussed various techniques we have used to reduce CPU usage on a database server environment**
- **Make sure you have good statistics**
- **Monitor before and after changes**



# THANK YOU

Michael Zechman

