

# Oracle instance assessment

ITOUG **Tech Day**  
Milano - 11 Novembre 2016



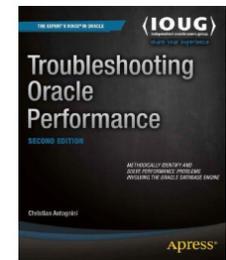
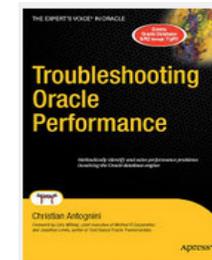


- CEO di ICTeam (dal 2006)
- Cofondatore di ICTeam (1999)
- Banca Popolare di Bergamo (1994-1999)
- Olivetti (1986-1994)
- Laurea in Scienze dell'Informazione a Milano (1986)

## Performance geek

- la passione nasce fin dalla tesi sull'ottimizzazione delle performance di uno UNIX BSD (Olivetti 1985 - 1986)
- l'analisi delle performance Oracle inizia nel 1988 nella relazione Oracle-Unix
- sviluppo le funzionalità asynchronous I/O, post/wait, list I/O, priority ed affinity cpu, pipe in memory nel kernel per supportare Sybase, Informix ed Oracle nel benchmark TPC-A (1990 Olivetti è terza al mondo con Oracle nei sistemi x86, prima è Compaq)
- inizio a lavorare sulle performance più dal punto di vista Oracle (modellazione, SQL, parametri, ...) dal 1994
- ancora oggi faccio il Performance Consultant (da 9i a 12c)

Ho avuto l'onore di essere uno dei technical reviewer del libro dell'amico Christian Antognini

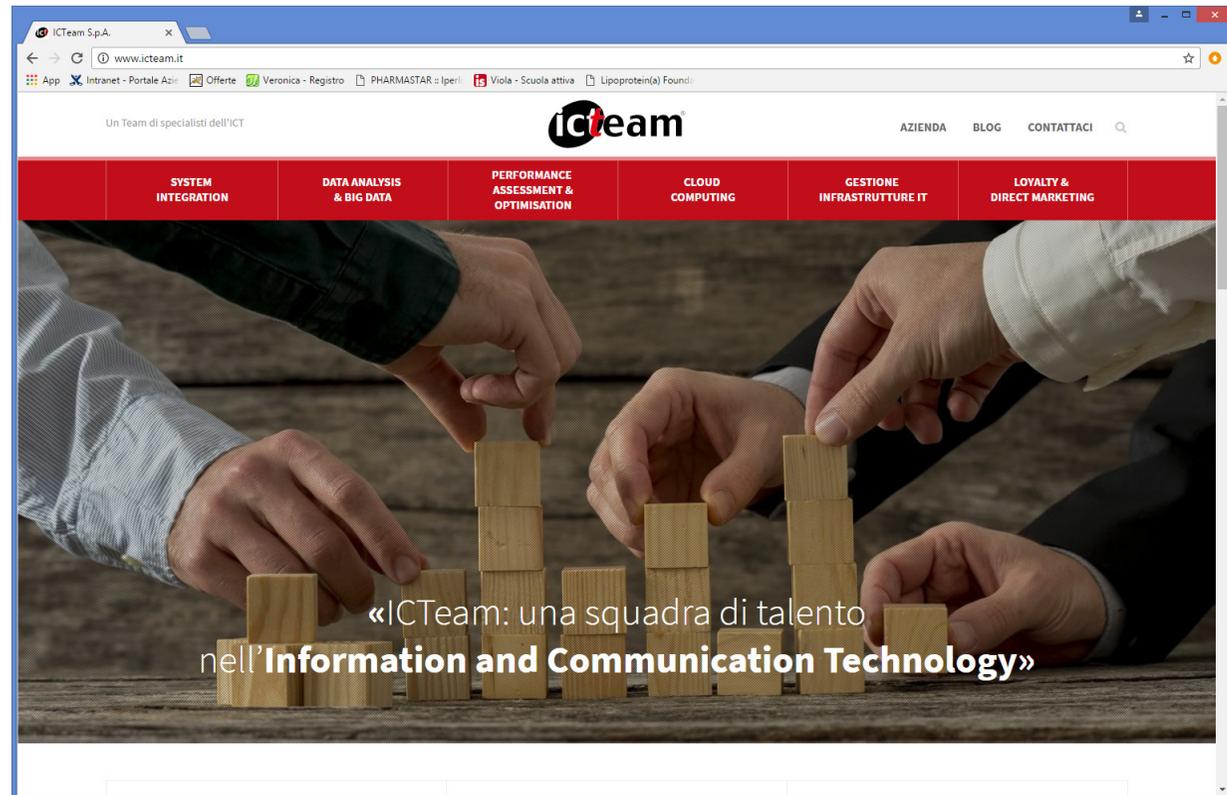


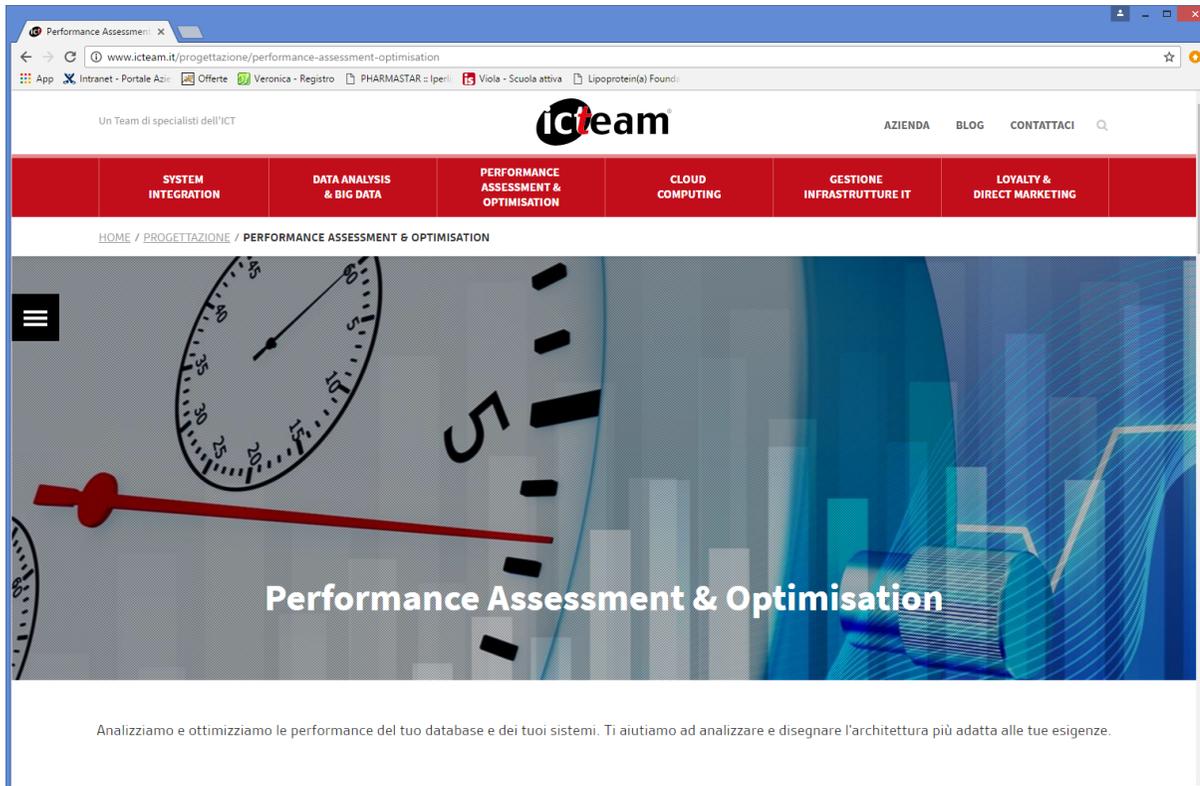


- ICTeam nasce nel 1999
- 6 soci fondatori tutti provenienti dalla Banca Popolare Bergamo che implementò un progetto più unico che raro: la completa rifondazione del sistema informativo senza mainframe, tutto su Unix ed Oracle

Sul mercato abbiamo messo la nostra competenza tecnica ed il nostro entusiasmo nel fare progetti e nel risolvere problemi.

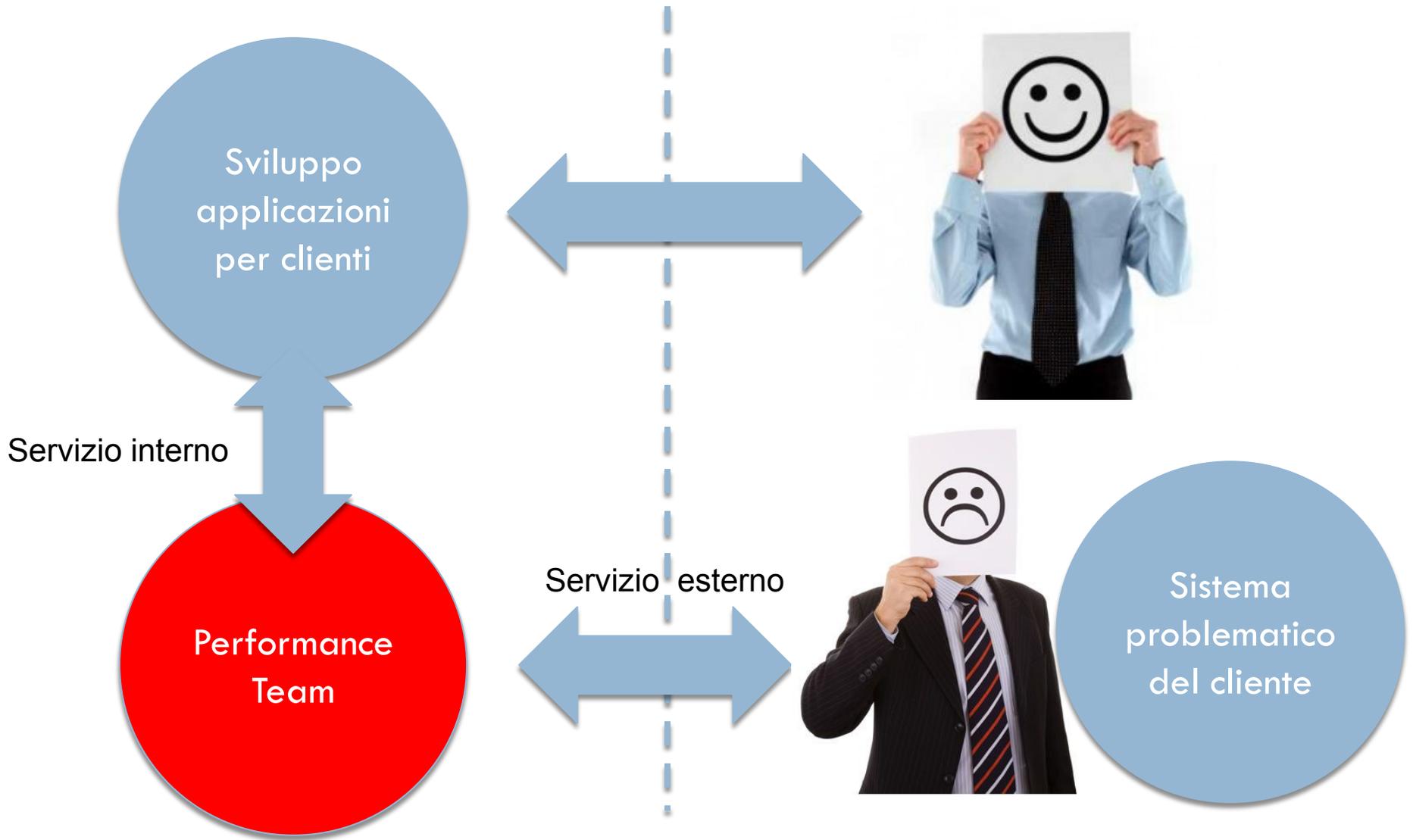
Siamo 110 professionisti, di cui 104 tecnici e tecnici vogliamo restare!





- dal 2013 abbiamo creato uno specifico gruppo di persone che si occupa esclusivamente di performance e prevalentemente di performance su Oracle
- siamo in grado di analizzare architetture complesse soffermandoci su tutte le componenti in gioco (hardware e software, di sistema ed applicativo)
- abbiamo esaminato alcuni dei più grandi ed importanti database Oracle d'Italia

[performance\\_team@icteam.it](mailto:performance_team@icteam.it)



- L'assessment di una istanza Oracle tipicamente ci viene richiesto per uno (o per una combinazione) dei seguenti motivi:
- utenti tipicamente *adirati* per le performance del sistema;
  - degrado del trend di performance nel tempo;
  - dubbi sulla possibilità di far crescere il carico a fronte di una ipotetica carenza hardware;
  - basso consumo di risorse (magari dopo un investimento hardware ...);
  - presenza di colli di bottiglia in determinati momenti della giornata o per specifiche attività.

Troppo spesso siamo coinvolti quando l'incendio è già in atto e quindi *ci tocca fare i pompieri*.

Anche se oggettivamente siamo un po' meno belli di quelli qui rappresentati ...



La parola assessment può essere tradotta in italiano come:

- determinazione, stima, valutazione, giudizio, opinione;
- ma anche come accertamento tributario e direi che non è il nostro caso ...

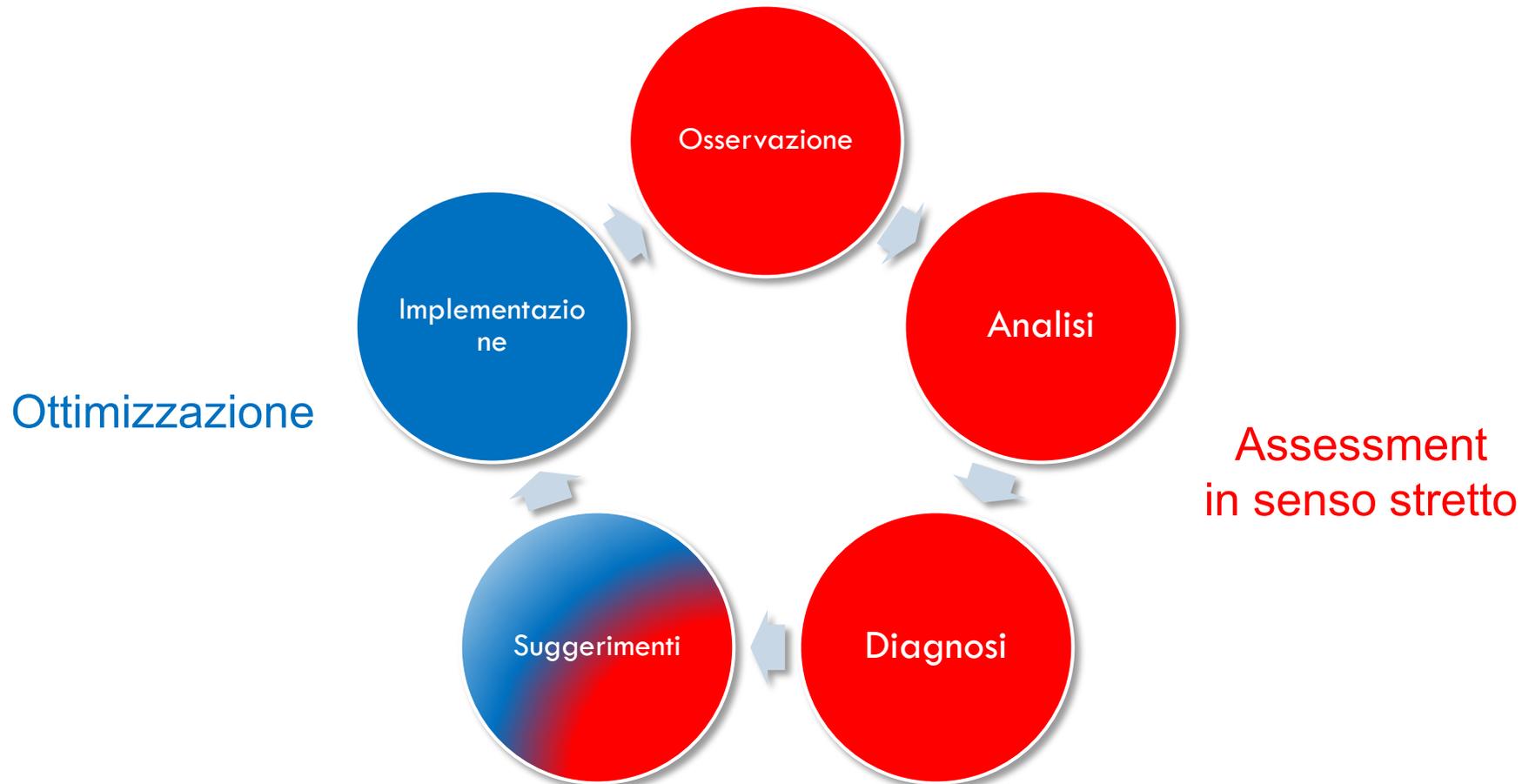
L'assessment quindi dovrebbe essere in primis una fase di analisi atta a determinare un giudizio su un database Oracle.

Attenzione questa fase deve considerare gli altri attori in gioco e soprattutto l'obiettivo del giudizio che si è chiamati a dare.

Ad esempio i seguenti scenari sono differenti:

- l'analisi viene richiesta perché gli utenti sono insoddisfatti;
- l'analisi viene richiesta perché, in previsione di un aumento sensibile del carico (il business segnala un sensibile incremento delle richieste), pur essendo positiva l'attuale situazione ci si preoccupa di comprendere se il maggior carico previsto potrà essere ben supportato.

In un assessment di performance l'obiettivo del solo giudizio viene ampliato nell'individuare i problemi presenti, definire una diagnosi ed i possibili interventi di ottimizzazione, fissando una scaletta di priorità della loro implementazione.



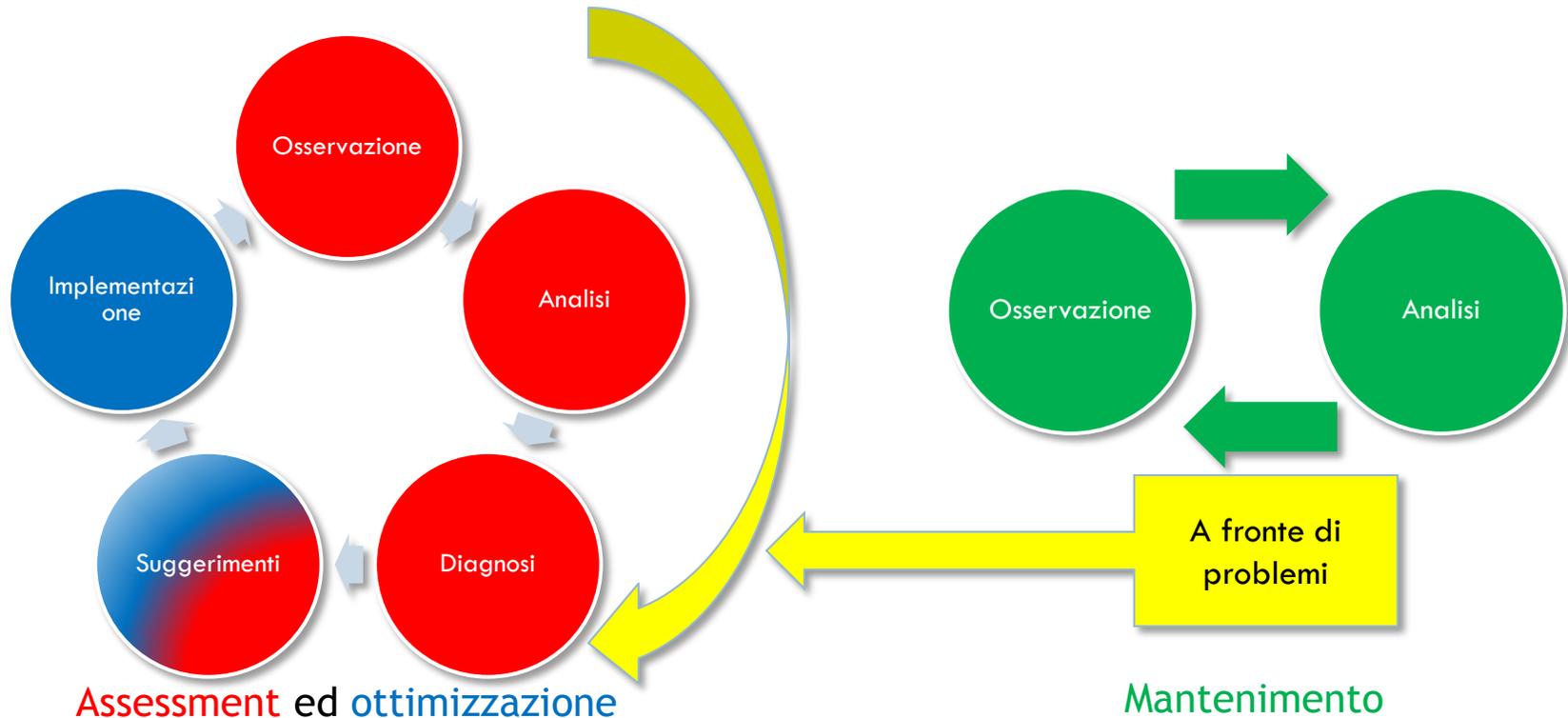
L'assessment relativo alle performance dovrebbe prevedere i seguenti passi:

- l'**osservazione** del database, possibilmente per un intervallo temporale consono alla modalità di utilizzo del database stesso o del problema specifico che si vuole indagare (dati storici, dati recenti, monitoring in real time, non solo del database). L'osservazione può in alcuni casi essere guidata dal cliente che ha conoscenza del problema da risolvere;
- l'**analisi** del comportamento osservato del database e delle altre componenti, quanto più possibile con una visione olistica (risorse hardware, configurazione database, statement, applicazione, tipologia di carico, ...);
- la **diagnosi** con l'individuazione degli elementi con comportamento non consono (siano essi di natura hardware, sistemistica o applicativa). Durante la diagnosi si deve fare attenzione ai falsi positivi ed ai falsi negativi (ad esempio consumi CPU, carico I/O, ...). L'output della diagnosi è il **giudizio** sul database ed il suo ecosistema con l'individuazione degli eventuali problemi presenti;

L'assessment relativo alle performance dovrebbe prevedere quindi i seguenti passi (continua):

- una serie di **suggerimenti** atti ad ottimizzare la situazione. L'elenco dei suggerimenti viene associato alla priorità, e ove possibile ai tempi e costi di realizzazione e relativi benefici. Per migliorare l'esito di questa fase sarebbe fondamentale avere un ambiente specchio dove poter provare la maggior parte dei suggerimenti, ma ciò è praticamente impossibile, vuoi per i costi indotti, vuoi per l'impossibilità fisica di farlo, vuoi per l'eventuale impossibilità di replicare lo stesso carico, ... I **suggerimenti** sono comunque il tipico output atteso dal cliente (sia esso interno o esterno). Si è fra la fase di assessment e quella di ottimizzazione;
- Il passo più importante che dovrebbe seguire è **l'implementazione dei suggerimenti**. L'implementazione si traduce in una serie di attività che possono spaziare in diverse direzioni. La sequenza dovrebbe seguire la scaletta di priorità ed il rapporto costi/benefici indicato prima. Anche in questa fase sarebbe fondamentale avere un ambiente specchio dove poter provare.

L'implementazione dovrebbe essere seguita da una nuova osservazione ed analisi.  
Nel mondo perfetto questo diventerebbe un ciclo infinito di continuo miglioramento; in realtà le risorse non sono infinite e ci si ferma al raggiungimento dell'obiettivo di business con il miglior compromesso tra il beneficio possibile ed il costo relativo.



Attenzione: altra cosa è la comprensione e l'intervento a cuore aperto a fronte di un problema istantaneo:

- degrado improvviso di uno o più statement (ad esempio a causa di un cambio piano inaspettato);
- manifestazione di eventi di wait che vanno a bloccare il database in modo del tutto inatteso;
- degrado delle performance di qualche componente hardware che va a modificare il comportamento del database;
- degrado dovuto ad un rilascio di software che mini in qualche modo il comportamento del database.



VS



Nel mondo però non si vive di sola teoria e l'esperienza ci insegna che:

- le performance dovrebbero essere un tema basilare da affrontare immediatamente in fase di progettazione di un ambiente/applicazione. Troppo spesso questo non accade, perché il relativo impegno è visto come un costo in termini economici e un ritardo sui tempi. Il risultato negativo è talvolta quello di inseguire frettolosamente i problemi a fine progetto, con uno stress non indifferente;
- le applicazioni nascono per fare qualcosa e poi si trasformano o subiscono un aumento di carico non preventivato, quindi la scalabilità futura è una tema che bisognerebbe sempre porsi;
- nella maggior parte dei casi l'attenzione alle performance si pone quando il problema è già scoppiato e quando gli utenti hanno già in mano la forza ...;
- quindi l'assessment ed il pronto soccorso sono talvolta diventati interscambiabili nella nomenclatura, l'importante è avere ben chiaro come agire nei vari casi e come portare cultura (sì intendo proprio cultura) delle performance nel mondo.

Per eseguire un assessment *serio*, così come per risolvere problemi urgenti, serve una cassetta degli attrezzi che dovrebbe contenere:

- la possibilità di ragionare su dati statistici (a diversi livelli: sistema operativo, database, storage, applicativo) mantenuti per un intervallo temporale significativo secondo la natura stessa dell'ambiente che si vuole analizzare;
- la possibilità di osservare direttamente il sistema utilizzando, secondo necessità ed opportunità, tutte le leve ammesse:
  - AWR e ASH, quindi di fatto è necessario avere la licenza del Diagnostic Pack. Ciò è utile sia per le attività di osservazione/analisi in real time che di trend;
  - SQL trace per le attività di osservazione in real time;
  - Real Time SQL Monitoring, quindi avere la licenza del Tuning Pack;
  - script ad hoc su tavole di sistema (le V\$, talvolta ma di rado le X\$, e sicuramente le DBA\_HIST\_%);
- conoscenza del comportamento di Oracle, ma anche degli altri elementi dell'ecosistema (sistema operativo, storage, ...);
- esperienza e successi sul campo di battaglia (autostima e credibilità);
- Internet (accesso alle informazioni presenti in rete: metalink, blog, guru);
- libri.

Fra gli strumenti sono da inserire anche delle caratteristiche personali:

- pazienza e controllo, sotto diversi punti di vista:
  - pazienza nell'osservare (anche ore ed ore ed in qualsiasi momento della giornata/nottata);
  - pazienza e controllo quando sotto pressione, soprattutto se si è in pronto intervento;
- curiosità e passione: la competenza si costruisce essendo curiosi e volendo capire quanto più possibile, sempre e comunque;
- capacità di stupirsi: a volte ci si trova di fronte a situazioni non previste e strane, lo stupirsi aiuta ad indagare a fondo;
- saper ripartire: ogni volta non è mai la stessa musica ...;
- costanza: di fronte ad un problema non si molla mai ...;
- il piacere della sfida: capire e risolvere fa rilasciare endorfine.



And last but not least ... due componenti da mixare:

## disciplina (Discipline) e indisciplina (Indiscipline)

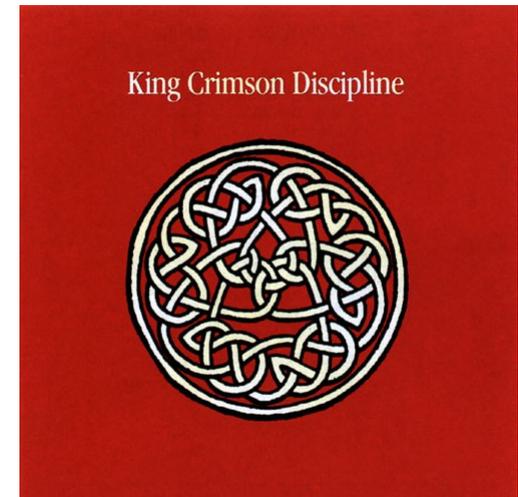
*Lo dico così posso vincere la scommessa di nominare anche questa volta Robert Fripp in una presentazione.* Discipline e Indiscipline sono due canzoni dei King Crimson che esprimono proprio l'armonia che possa/debba esservi fra la creatività e l'essere efficaci.

[Per chi fosse curioso una delle tante recensioni sull'album Discipline:  
https://emamartin.wordpress.com/2013/08/02/king-crimson-discipline/](https://emamartin.wordpress.com/2013/08/02/king-crimson-discipline/)

*[RF] Discipline is not an end in itself, just a means to an end*

*The word in our culture can sometimes have a pejorative feel to it. [But] to me, discipline is liberating; it's not constricting or restricting at all.*

*Discipline is the capacity to be effectual in time. That is: we can make a commitment, we can say "I \_will\_ do this," and know it will be done. And this is a remarkable freedom. Because if you make a commitment, it will be honored; and if you're working with other people who say, "I will do this," and you can bank on it, [a lot] becomes possible and your life takes a quantum leap. So it's liberating, not constraining or restricting.*



## Esistono ricette preconfenzionate? Penso proprio di NO.

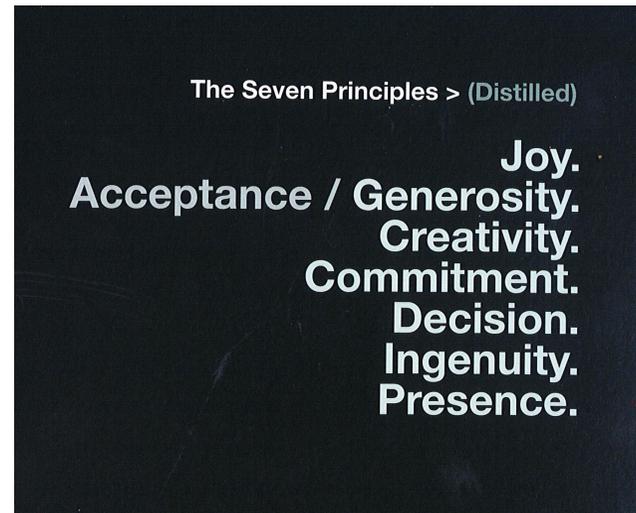
Penso che vi siano dei principi, anche se non sempre gli stessi vengono seguiti con costanza nel tempo.

Ad esempio questi sono dei principi in campi diversi:



Non fumare, non diventare obesi, eseguire 30m di esercizio tutti i giorni e seguire una dieta più sana (- o zero carne, + frutta e verdura e cibi integrali) diminuirebbe del 78% il rischio di malattie croniche.

Fonte: *Sei quello che mangi di Michael Greger (2016)* e <http://adph.org/gal/assets/LifeSimpleSeven.pdf>



I 7 principi dei King Crimson per creare musica e performance ottimali.

Fonte: *King Crimson European Tour 2016*

Proviamo ad enunciare 7 possibili principi seguendo la falsariga dei 7 simple step dell'*American Heart Association*:

1. mantieni a norma la tua pressione per evitare di affaticare il cuore (< 120/80)
2. mantieni a norma il tuo colesterolo per evitare l'aterosclerosi (LDL < 100, HDL > 45)
3. mantieni a norma il glucosio nel sangue per evitare il diabete di tipo 2 (emoglobina glicata < 5.7)
4. dieta corretta (quanto più vegetariana possibile) per evitare infiammazioni, radicali liberi, ossidazione, colesterolo, etc
5. mantieni un peso a norma per evitare grassi in eccesso e sforzo cuore (indice BMI intorno ai 21, no pancia)
6. sempre attivi per contrastare ossidazione, grasso, etc (> 30m di esercizi al giorno)
7. non fumare (basta guardare i polmoni dei fumatori)!

1. mantieni il numero di LIO per statement più basso possibile per non affaticare il sistema
2. mantieni le contese (latch, lock, ...) le più basse possibili per non ostruire l'attività
3. limita il numero di PIO quanto più possibile, e comunque ottimizza la latenza e la banda dell'I/O
4. non inserire SQL spazzatura, vale a dire statement che utilizzino «inutilmente» troppe risorse (troppe LIO, troppe PIO, ..)
5. il workload complessivo non deve avvicinarsi alla saturazione delle risorse, viceversa sarà necessario ottimizzare o crescere
6. mantenere in forma tutto l'ecosistema (ad esempio non far peggiorare lo storage)
7. limita l'uso della TEMP. Non aggiungere ulteriore I/O a quel che si deve già fare!

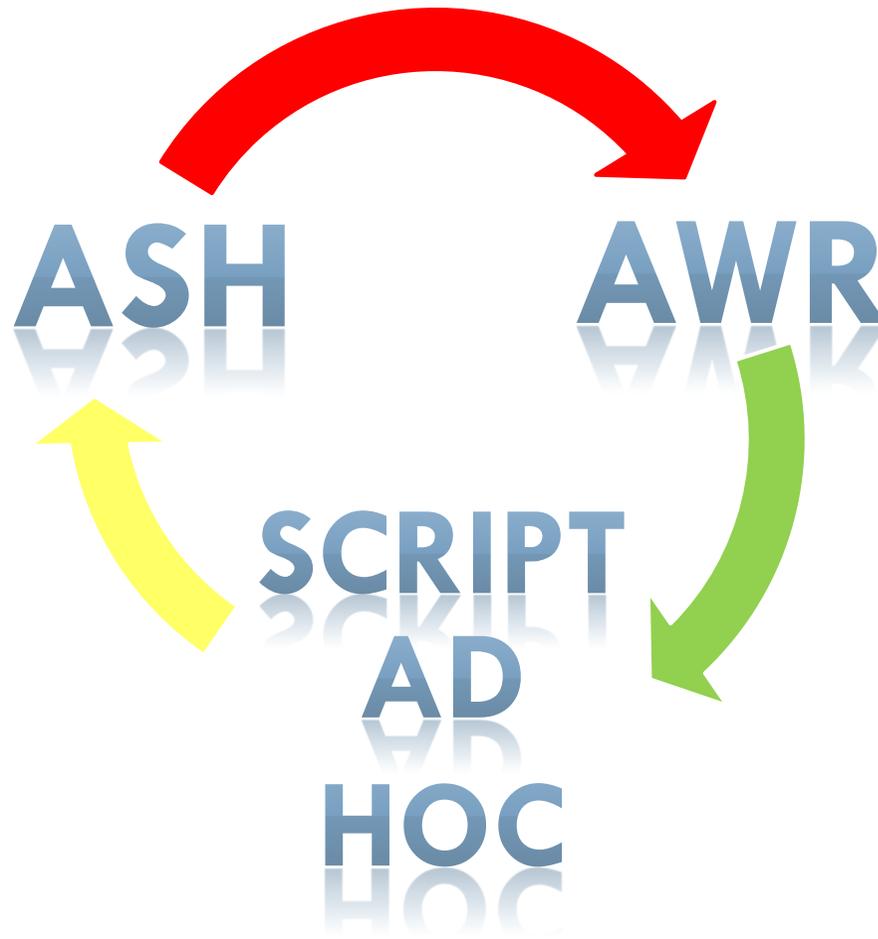
Non ci si può dimenticare un principio fondamentale:

Fai che il database utilizzi il piano di esecuzione migliore possibile!

Non è un compito così banale, si tratta di:

- modellare in maniera ottimale la base dati;
- configurare al meglio il database ed il query optimizer;
- definire la migliore strategia di calcolo statistiche possibile;
- scrivere bene l'SQL, sì anche questo ha la sua importanza
- utilizzare eventualmente gli hint o altri strumenti (SQL profile/plan baseline) atti a consigliare/condizionare il query optimizer;
- capire profondamente i piani esecuzione per saperli interpretare;
- conoscere le funzionalità Oracle per individuare delle *scorciatoie intelligenti*;
- **insomma: saper configurare il database e saper cosa sia fare SQL Tuning!**

Solitamente partiamo con un'analisi complessiva (a meno che vi sia un problema preciso già sul tavolo) e da un primo colpo d'occhio. Si utilizzano gli strumenti ed i dati sotto licensing Diagnostic Pack.



Le slide che seguono non hanno alcuna pretesa di essere esaustive.

Sono solo qualche pillola, o se si preferisce qualche goccia del mare di quel che si può analizzare in un database Oracle.

Per approfondire la materia pratica servirebbero giorni e giorni ...

In due parole: il mondo AWR permette di campionare una serie di dati statistici e di generare report su questi dati.

Il report AWR presenta queste sezioni:

- **Main Report:** riguarda l'istanza relativa del report. C'è tanta roba! Dalla 12c vi sono ancora più informazioni ed una sezione ADDM;
- **RAC Statistics:** informazioni presenti in caso di configurazione RAC. Tendono a rappresentare l'interazione, o le contese fra i report;
- **Exadata ...:** informazioni presenti solo se l'istanza risiede su Exadata.

## Main Report

- [Report Summary](#)
- [Wait Events Statistics](#)
- [SQL Statistics](#)
- [Instance Activity Statistics](#)
- [IO Stats](#)
- [Buffer Pool Statistics](#)
- [Advisory Statistics](#)
- [Wait Statistics](#)
- [Undo Statistics](#)
- [Latch Statistics](#)
- [Segment Statistics](#)
- [Dictionary Cache Statistics](#)
- [Library Cache Statistics](#)
- [Memory Statistics](#)
- [Replication Statistics \(GoldenGate, XStream\)](#)
- [Streams Statistics](#)
- [Resource Limit Statistics](#)
- [Shared Server Statistics](#)
- [init.ora Parameters](#)
- [Active Session History \(ASH\) Report](#)
- [ADDM Reports](#)

## RAC Statistics

- [RAC Report Summary](#)
- [Global Messaging Statistics](#)
- [Global CR Served Stats](#)
- [Global CURRENT Served Stats](#)
- [Global Cache Transfer Stats](#)
- [Interconnect Stats](#)
- [Dynamic Remastering Statistics](#)

## Exadata Configuration and Statistics

- [Exadata Report Summary](#)
- [Exadata Server Configuration](#)
- [Exadata Server Health Report](#)
- [Exadata Statistics](#)

Il report AWR è fondamentale per capire cosa stia facendo una istanza Oracle (oltre che per confrontare due nodi, o due intervalli temporali diversi, o il comportamento di un singolo statement). Tante sono le cose che è possibile osservare/analizzare:

- la configurazione host e database;
- il comportamento dell'host che ospita il database;
- il tempo speso nel database;
- gli indicatori di *movimentazione/throughput* (LIO, PIO, parse, user call, ...);
- dove viene speso il tempo nel database (eventi);
- le statistiche degli statement più pesanti;
- I/O indotto e tempi di risposta;
- contese;
- PGA/TEMP;
- accesso ai segmenti;
- ...

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	37668	27-Ott-13 22:15:13	272	1.4
End Snap:	37675	28-Ott-13 00:00:03	226	1.6
Elapsed:		104.83 (mins)		
DB Time:		5,683.49 (mins)		

### Operating System Statistics - Detail

Snap Time	Load	%busy	%user	%sys	%idle	%iowait
27-Ott 22:15:13	41.89					
27-Ott 22:30:06	69.29	10.83	8.84	1.56	89.17	10.43
27-Ott 22:45:15	56.36	6.75	4.95	1.38	93.25	10.38
27-Ott 23:00:23	74.57	7.74	5.68	1.58	92.26	12.12
27-Ott 23:15:32	62.34	11.30	9.12	1.66	88.70	12.32
27-Ott 23:30:02	28.20	15.58	13.68	1.60	84.42	5.43
27-Ott 23:45:03	52.93	14.53	12.83	1.39	85.47	3.93
28-Ott 00:00:03	52.89	6.36	4.66	1.30	93.64	8.14

Per le indagini è utile avere snapshot frequenti (es. 15 min.) per poter essere più precisi, il default è di 60 min.

Il report AWR mostra dati aggregati per l'intervallo prescelto che potrebbe contenere più snapshot.

Si consiglia di mantenere in linea almeno 3 mesi di dati e/o di provvedere a salvare i dati dei campionamenti AWR in altro repository.

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	37668	27-Ott-13 22:15:13	272	1.4
End Snap:	37675	28-Ott-13 00:00:03	226	1.6
Elapsed:		104.83 (mins)		
DB Time:		5,683.49 (mins)		

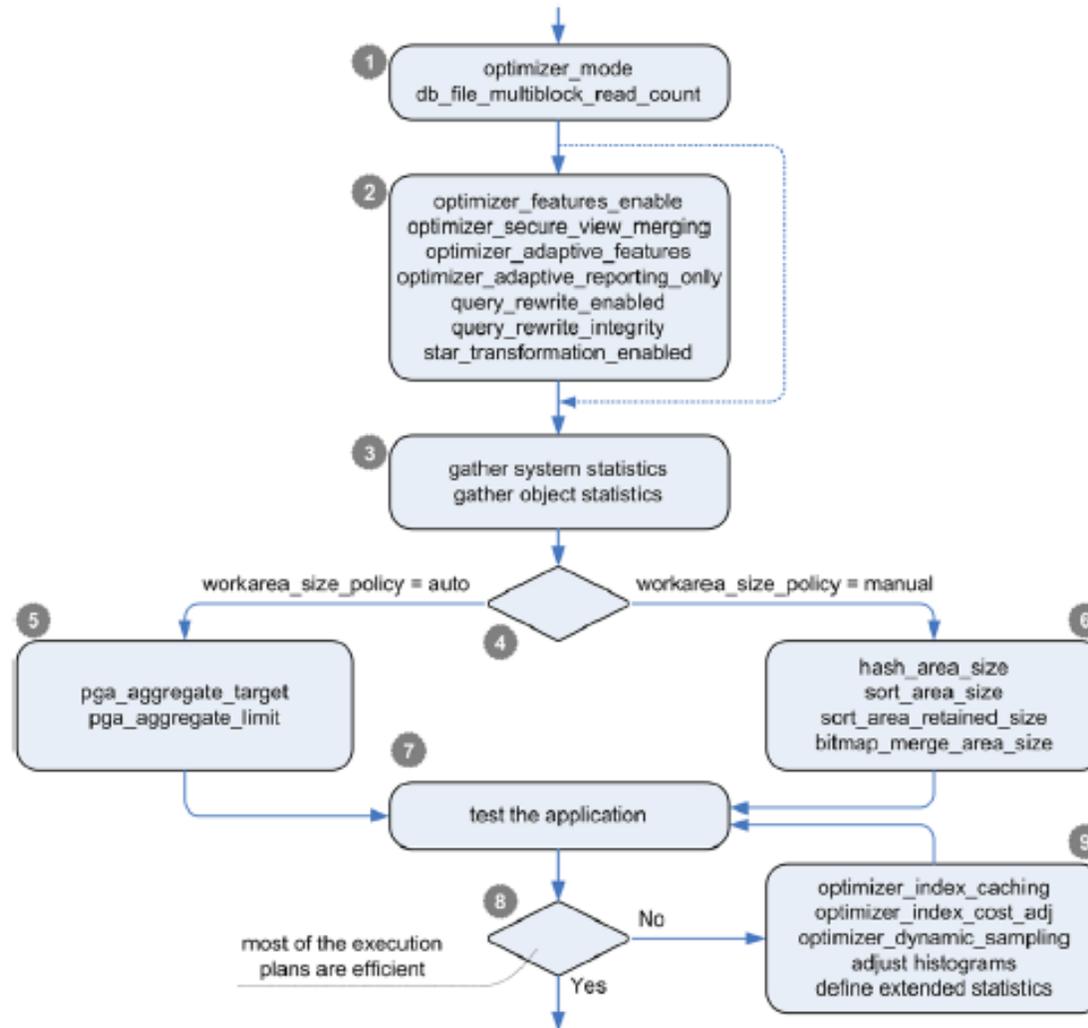
Analizzare i dati di un report AWR con dati aggregati di più snapshot potrebbe risultare talvolta fuorviante nel non rappresentare dei picchi, per cui può essere utile:

- o generare report per ogni snapshot;
- o utilizzare query ad hoc sulle tavole DBA\_HIST\_% che sono alla base degli stessi report AWR.

# Osservazione: le 121 DBA\_HIST\_%

```
DBA_HIST_ACTIVE_SESS_HISTORY
DBA_HIST_APPLY_SUMMARY
DBA_HIST_ASH_SNAPSHOT
DBA_HIST_BASELINE
DBA_HIST_BASELINE_DETAILS
DBA_HIST_BASELINE_METADATA
DBA_HIST_BASELINE_TEMPLATE
DBA_HIST_BG_EVENT_SUMMARY
DBA_HIST_BUFFERED_QUEUES
DBA_HIST_BUFFERED_SUBSCRIBERS
DBA_HIST_BUFFER_POOL_STAT
DBA_HIST_CAPTURE
DBA_HIST_CLUSTER_INTERCON
DBA_HIST_COLORED_SQL
DBA_HIST_COMP_IOSTAT
DBA_HIST_CR_BLOCK_SERVER
DBA_HIST_CURRENT_BLOCK_SERVER
DBA_HIST_DATABASE_INSTANCE
DBA_HIST_DATAFILE
DBA_HIST_DB_CACHE_ADVICE
DBA_HIST_DISPATCHER
DBA_HIST_DLM_MISC
DBA_HIST_DYN_REMASTER_STATS
DBA_HIST_ENQUEUE_STAT
DBA_HIST_EVENT_HISTOGRAM
DBA_HIST_EVENT_NAME
DBA_HIST_FILEMETRIC_HISTORY
DBA_HIST_FILESTATXS
DBA_HIST_IC_CLIENT_STATS
DBA_HIST_IC_DEVICE_STATS
DBA_HIST_INSTANCE_RECOVERY
DBA_HIST_INST_CACHE_TRANSFER
DBA_HIST_INTERCONNECT_PINGS
DBA_HIST_IOSTAT_DETAIL
DBA_HIST_IOSTAT_FILETYPE
DBA_HIST_IOSTAT_FILETYPE_NAME
DBA_HIST_IOSTAT_FUNCTION
DBA_HIST_IOSTAT_FUNCTION_NAME
DBA_HIST_JAVA_POOL_ADVICE
DBA_HIST_LATCH
DBA_HIST_LATCH_CHILDREN
DBA_HIST_LATCH_MISSES_SUMMARY
DBA_HIST_LATCH_NAME
DBA_HIST_LATCH_PARENT
DBA_HIST_LIBRARYCACHE
DBA_HIST_LOG
DBA_HIST_MEM_DYNAMIC_COMP
DBA_HIST_MEMORY_RESIZE_OPS
DBA_HIST_MEMORY_TARGET_ADVICE
DBA_HIST_METRIC_NAME
DBA_HIST_MTTR_TARGET_ADVICE
DBA_HIST_MUTEX_SLEEP
DBA_HIST_MVPARAMETER
DBA_HIST_OPTIMIZER_ENV
DBA_HIST_OSSTAT
DBA_HIST_OSSTAT_NAME
DBA_HIST_PARAMETER
DBA_HIST_PARAMETER_NAME
DBA_HIST_PDB_INSTANCE
DBA_HIST_PERSISTENT_QMN_CACHE
DBA_HIST_PERSISTENT_QUEUES
DBA_HIST_PERSISTENT_SUBS
DBA_HIST_PGASTAT
DBA_HIST_PGA_TARGET_ADVICE
DBA_HIST_PLAN_OPERATION_NAME
DBA_HIST_PLAN_OPTION_NAME
DBA_HIST_PROCESS_MEM_SUMMARY
DBA_HIST_REPLICATION_TBL_STATS
DBA_HIST_REPLICATION_TXN_STATS
DBA_HIST_REPORTS
DBA_HIST_REPORTS_CONTROL
DBA_HIST_REPORTS_DETAILS
DBA_HIST_REPORTS_TIMEBANDS
DBA_HIST_RESOURCE_LIMIT
DBA_HIST_ROWCACHE_SUMMARY
DBA_HIST_RSRC_CONSUMER_GROUP
DBA_HIST_RSRC_PLAN
DBA_HIST_RULE_SET
DBA_HIST_SEG_STAT
DBA_HIST_SEG_STAT_OBJ
DBA_HIST_SERVICE_NAME
DBA_HIST_SERVICE_STAT
DBA_HIST_SERVICE_WAIT_CLASS
DBA_HIST_SESSMETRIC_HISTORY
DBA_HIST_SESS_SGA_STATS
DBA_HIST_SESS_TIME_STATS
DBA_HIST_SGA
DBA_HIST_SGASTAT
DBA_HIST_SGA_TARGET_ADVICE
DBA_HIST_SHARED_POOL_ADVICE
DBA_HIST_SHARED_SERVER_SUMMARY
DBA_HIST_SNAP_ERROR
DBA_HIST_SNAPSHOT
DBA_HIST_SQLBIND
DBA_HIST_SQL_BIND_METADATA
DBA_HIST_SQLCOMMAND_NAME
DBA_HIST_SQL_PLAN
DBA_HIST_SQLSTAT
DBA_HIST_SQL_SUMMARY
DBA_HIST_SQLTEXT
DBA_HIST_SQL_WORKAREA_HSTGRM
DBA_HIST_STAT_NAME
DBA_HIST_STREAMS_APPLY_SUM
DBA_HIST_STREAMS_CAPTURE
DBA_HIST_STREAMS_POOL_ADVICE
DBA_HIST_SYSMETRIC_HISTORY
DBA_HIST_SYSMETRIC_SUMMARY
DBA_HIST_SYSSSTAT
DBA_HIST_SYSTEM_EVENT
DBA_HIST_SYS_TIME_MODEL
DBA_HIST_TABLESPACE
DBA_HIST_TABLESPACE_STAT
DBA_HIST_TBSPC_SPACE_USAGE
DBA_HIST_TEMPFILE
DBA_HIST_TEMPSTATXS
DBA_HIST_THREAD
DBA_HIST_TOPLEVELCALL_NAME
DBA_HIST_UNDOSTAT
DBA_HIST_WAITCLASSMET_HISTORY
DBA_HIST_WAITSTAT
DBA_HIST_WR_CONTROL
```

Query optimizer roadmap plan (da Troubleshooting Oracle Performance di Christian Antognini):



Cosa sta facendo l'istanza? Lavora molto ? O *attende* molto?

	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	37668	27-Ott-13 22:15:13	272	1.4
End Snap:	37675	28-Ott-13 00:00:03	226	1.6
Elapsed:		104.83 (mins)		
DB Time:		5,683.49 (mins)		

O fa poco (o nulla) ?

	Snap Id	Snap Time	Sessions	Cursors/Session	Instances
Begin Snap:	2152	20-Ott-15 16:00:02	45	1.1	2
End Snap:	2153	20-Ott-15 17:00:07	317	1.9	2
Elapsed:		60.09 (mins)			
DB Time:		37.61 (mins)			

Che tipo di carico sta gestendo?

Si veda la prossima slide ...

## Load Profile

	Per Second	Per Transaction	Per Exec	Per Call
DB Time(s):	4.4	11.4	0.03	0.48
DB CPU(s):	1.6	4.2	0.01	0.18
Background CPU(s):	0.1	0.2	0.00	0.00
Redo size (bytes):	4,964,670.3	12,882,480.7		
Logical read (blocks):	28,545.0	74,069.3		
Block changes:	7,121.8	18,479.8		
Physical read (blocks):	8,627.7	22,387.4		
Physical write (blocks):	1,993.0	5,171.4		
Read IO requests:	322.9	837.8		
Write IO requests:	387.2	1,004.6		
Read IO (MB):	134.8	349.8		
Write IO (MB):	31.1	80.8		
IM scan rows:	0.0	0.0		
Session Logical Read IM:				
Global Cache blocks received:	169.1	438.8		
Global Cache blocks served:	194.6	504.9		
User calls:	9.2	23.9		
Parses (SQL):	10.6	27.5		
Hard parses (SQL):	0.8	2.0		
SQL Work Area (MB):	39.6	102.7		
Logons:	1.0	2.7		
Executes (SQL):	127.0	329.4		
Rollbacks:	0.0	0.0		
Transactions:	0.4			

## Instance Efficiency Percentages (Target 100%)

Buffer Nowait %:	99.81	Redo NoWait %:	99.91
Buffer Hit %:	96.94	In-memory Sort %:	100.00
Library Hit %:	99.27	Soft Parse %:	92.86
Execute to Parse %:	91.64	Latch Hit %:	99.84
Parse CPU to Parse Elapsed %:	9.49	% Non-Parse CPU:	96.74
Flash Cache Hit %:	84.43		

La sezione TOP Event evidenzia il comportamento macroscopico di un'istanza Oracle fornendo una classifica dei primi 10 eventi in ordine decrescente rispetto ai secondi spesi.

Le informazioni riguardano il numero di volte che si è verificato quell'evento di attesa, il tempo complessivo di attesa e quello medio, la % sul DB Time e la classe di appartenenza.

Top 10 Foreground Events by Total Wait Time

Event	Waits	Total Wait Time (sec)	Wait Avg(ms)	% DB time	Wait Class
DB CPU		9300,2		52.4	
direct path read temp	483,581	2309,7	5	13.0	User I/O
direct path write temp	406,665	766,6	2	4.3	User I/O
db file sequential read	185,181	533,5	3	3.0	User I/O
direct path read	64,242	315,6	5	1.8	User I/O
db file scattered read	40,835	281,8	7	1.6	User I/O
log file sync	91,950	250,1	3	1.4	Commit
log buffer space	1,659	89,7	54	.5	Configuration
enq: TX - row lock contention	550	24,2	44	.1	Application
control file sequential read	37,895	23,6	1	.1	System I/O

Il 52% del Db Time è speso in utilizzo di CPU

Il 17% circa del Db Time è speso in utilizzo di TEMP

L'interpretazione di questi dati è fondamentale per capire le eventuali problematiche presenti.

Le slide che seguono mostrano la rappresentazione grafica dell'esito di query ad hoc sulle tavole (sarebbe meglio dire viste) DBA\_HIST\_%.

Le grandezze che sono rappresentate nelle slide successive sono:

- il consumo di CPU (host e database);
- i tempi di latenza dell'I/O per diverse tipologie di I/O;
- la banda di I/O raggiunta.

Questi esempi dimostrano come partendo dalle DBA\_HIST\_% sia possibile costruire una serie di query ad hoc atte a studiare il comportamento ed il carico del database.

Prima però è da visionare l'elenco delle metriche che sono presenti nelle viste:

- per singolo minuto (DBA\_HIST\_SYSMETRIC\_HISTORY)
- per singolo snapshot (DBA\_HIST\_SYSMETRIC\_SUMMARY), ed in quest'ultimo caso i valori elaborati sono:
  - valori minimo, medio e massimo raggiunti nell'intervallo;
  - deviazione standard;
  - la somma dei quadrati delle deviazioni.

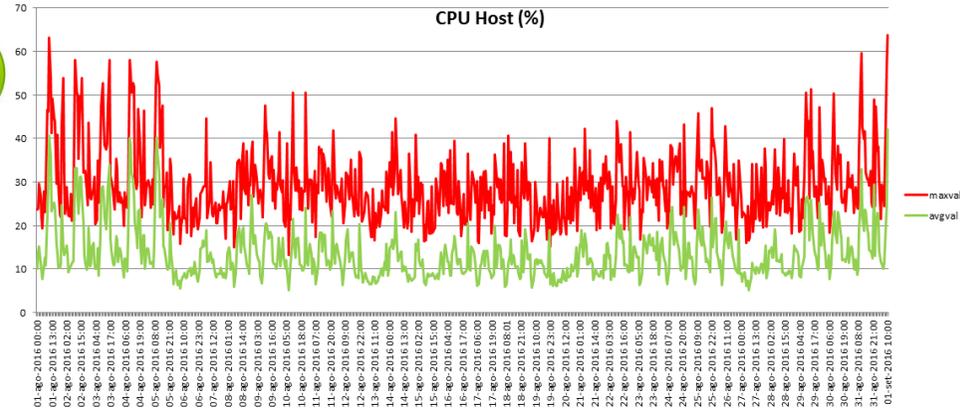
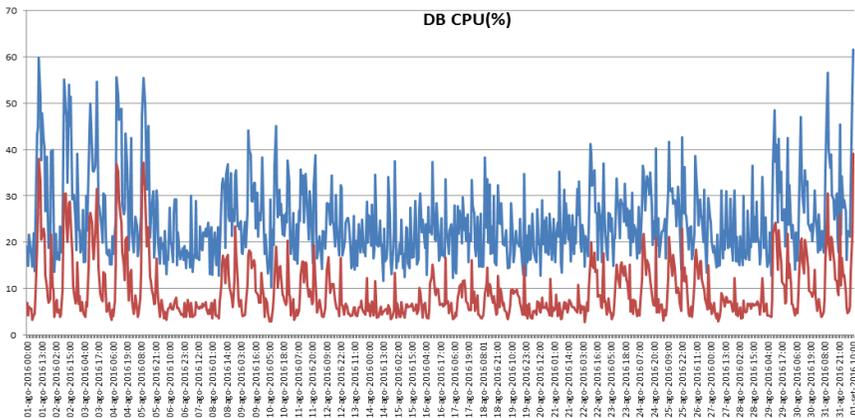
Average Active Sessions  
Average Synchronous Single-Block Read Latency  
DB Block Changes Per Txn  
Enqueue Requests Per Txn  
Executions Per Sec  
I/O Megabytes per Second  
I/O Requests per Second  
Logical Reads Per Txn  
Logons Per Sec  
Network Traffic Volume Per Sec  
Physical Reads Per Sec  
Physical Reads Per Txn  
Physical Writes Per Sec  
Redo Generated Per Sec  
Redo Generated Per Txn  
Response Time Per Txn  
SQL Service Response Time  
Total Parse Count Per Txn  
User Calls Per Sec  
User Transaction Per Sec

Active Parallel Sessions	Enqueue Deadlocks Per Sec
Active Serial Sessions	Enqueue Deadlocks Per Txn
Average Active Sessions	Enqueue Requests Per Sec
Average Synchronous Single-Block Read Latency	Enqueue Requests Per Txn
Background Checkpoints Per Sec	Enqueue Timeouts Per Sec
Background CPU Usage Per Sec	Enqueue Timeouts Per Txn
Background Time Per Sec	Enqueue Waits Per Sec
Branch Node Splits Per Sec	Enqueue Waits Per Txn
Branch Node Splits Per Txn	Execute Without Parse Ratio
Buffer Cache Hit Ratio	Executions Per Sec
Captured user calls	Executions Per Txn
Cell Physical IO Interconnect Bytes	Executions Per User Call
Consistent Read Changes Per Sec	Full Index Scans Per Sec
Consistent Read Changes Per Txn	Full Index Scans Per Txn
Consistent Read Gets Per Sec	GC CR Block Received Per Second
Consistent Read Gets Per Txn	GC CR Block Received Per Txn
CPU Usage Per Sec	GC Current Block Received Per Second
CPU Usage Per Txn	GC Current Block Received Per Txn
CR Blocks Created Per Sec	Global Cache Average CR Get Time
CR Blocks Created Per Txn	Global Cache Average Current Get Time
CR Undo Records Applied Per Sec	Global Cache Blocks Corrupted
CR Undo Records Applied Per Txn	Global Cache Blocks Lost
Current Logons Count	Hard Parse Count Per Sec
Current Open Cursors Count	Hard Parse Count Per Txn
Current OS Load	Host CPU Usage Per Sec
Cursor Cache Hit Ratio	Host CPU Utilization (%)
Database CPU Time Ratio	I/O Megabytes per Second
Database Time Per Sec	I/O Requests per Second
Database Wait Time Ratio	Leaf Node Splits Per Sec
DB Block Changes Per Sec	Leaf Node Splits Per Txn
DB Block Changes Per Txn	Library Cache Hit Ratio
DB Block Changes Per User Call	Library Cache Miss Ratio
DB Block Gets Per Sec	Logical Reads Per Sec
DB Block Gets Per Txn	Logical Reads Per Txn
DB Block Gets Per User Call	Logical Reads Per User Call
DBWR Checkpoints Per Sec	Logons Per Sec
DDL statements parallelized Per Sec	Logons Per Txn
Disk Sort Per Sec	Long Table Scans Per Sec
Disk Sort Per Txn	Long Table Scans Per Txn
DML statements parallelized Per Sec	Memory Sorts Ratio

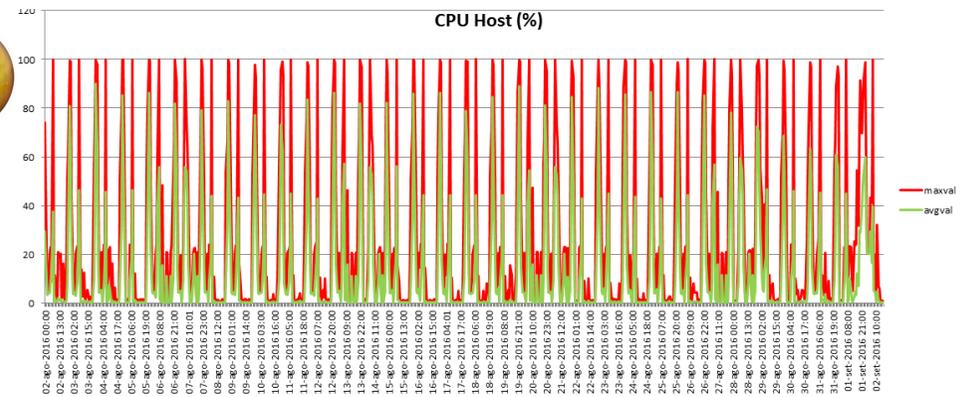
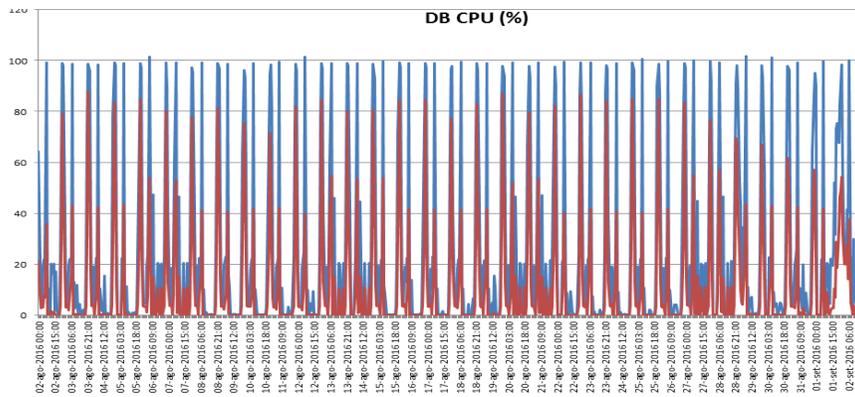
Network Traffic Volume Per Sec	Redo Generated Per Txn	Workload Capture and Replay status
Open Cursors Per Sec	Redo Writes Per Sec	
Open Cursors Per Txn	Redo Writes Per Txn	
Parse Failure Count Per Sec	Replayed user calls	
Parse Failure Count Per Txn	Response Time Per Txn	
PGA Cache Hit %	Row Cache Hit Ratio	
Physical Read Bytes Per Sec	Row Cache Miss Ratio	
Physical Read IO Requests Per Sec	Rows Per Sort	
Physical Read Total Bytes Per Sec	Run Queue Per Sec	
Physical Read Total IO Requests Per Sec	Session Count	
Physical Reads Direct Lobs Per Sec	Session Limit %	
Physical Reads Direct Lobs Per Txn	Shared Pool Free %	
Physical Reads Direct Per Sec	Soft Parse Ratio	
Physical Reads Direct Per Txn	SQL Service Response Time	
Physical Reads Per Sec	Streams Pool Usage Percentage	
Physical Reads Per Txn	Temp Space Used	
Physical Write Bytes Per Sec	Total Index Scans Per Sec	
Physical Write IO Requests Per Sec	Total Index Scans Per Txn	
Physical Write Total Bytes Per Sec	Total Parse Count Per Sec	
Physical Write Total IO Requests Per Sec	Total Parse Count Per Txn	
Physical Writes Direct Lobs Per Txn	Total PGA Allocated	
Physical Writes Direct Lobs Per Sec	Total PGA Used by SQL Workareas	
Physical Writes Direct Per Sec	Total Sorts Per User Call	
Physical Writes Direct Per Txn	Total Table Scans Per Sec	
Physical Writes Per Sec	Total Table Scans Per Txn	
Physical Writes Per Txn	Total Table Scans Per User Call	
PQ QC Session Count	Txns Per Logon	
PQ Slave Session Count	User Calls Per Sec	
Process Limit %	User Calls Per Txn	
PX downgraded to serial Per Sec	User Calls Ratio	
PX downgraded 1 to 25% Per Sec	User Commits Per Sec	
PX downgraded 25 to 50% Per Sec	User Commits Percentage	
PX downgraded 50 to 75% Per Sec	User Limit %	
PX downgraded 75 to 99% Per Sec	User Rollback Undo Records Applied Per Txn	
PX operations not downgraded Per Sec	User Rollback UndoRec Applied Per Sec	
Queries parallelized Per Sec	User Rollbacks Per Sec	
Recursive Calls Per Sec	User Rollbacks Percentage	
Recursive Calls Per Txn	User Transaction Per Sec	
Redo Allocation Hit Ratio	VM in bytes Per Sec	
Redo Generated Per Sec	VM out bytes Per Sec	

# Analisi: monitorare l'utilizzo di CPU

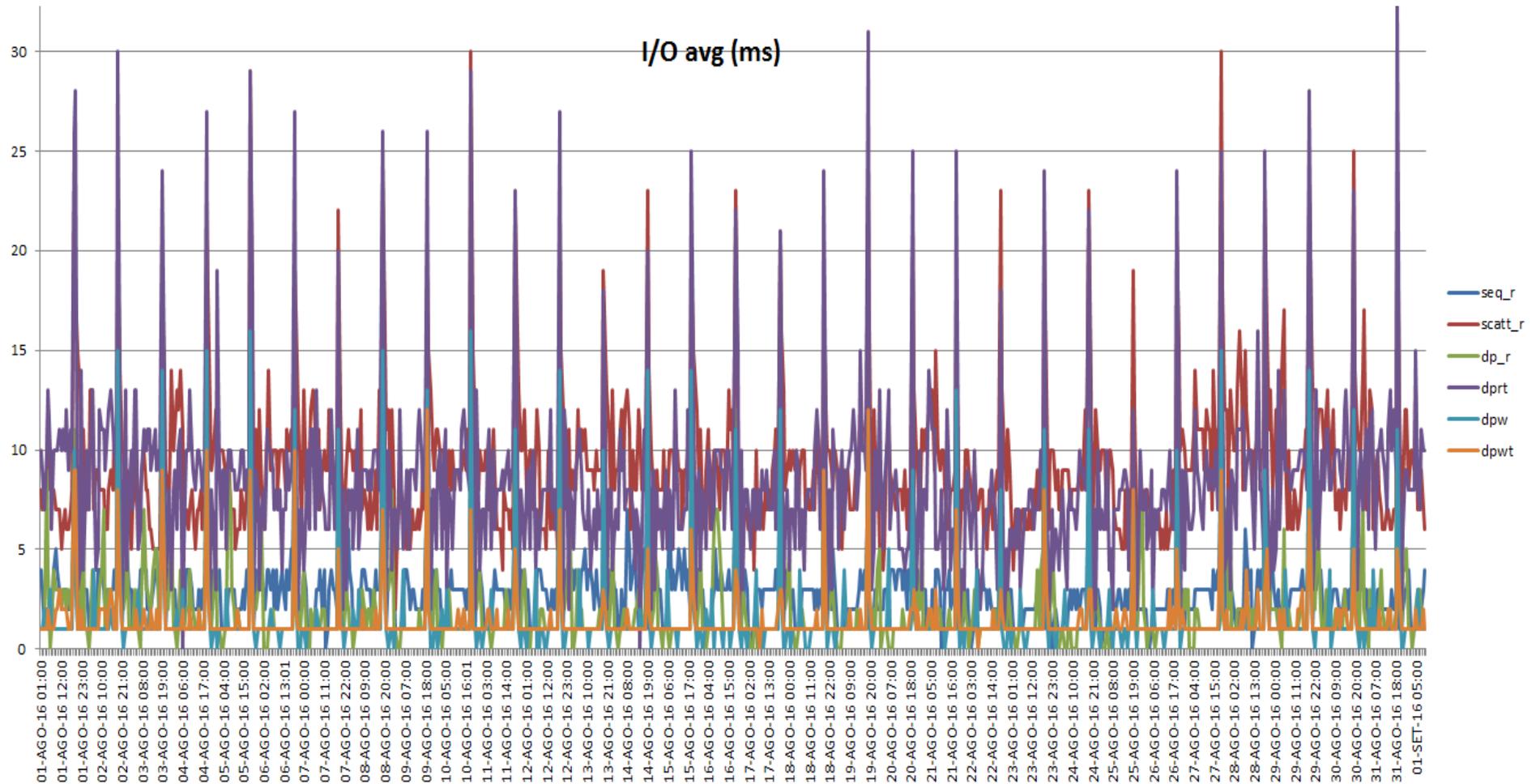
Per questo scopo solitamente si incrociano i dati statistici (Oracle) su consumi di CPU lato Host e lato Database:



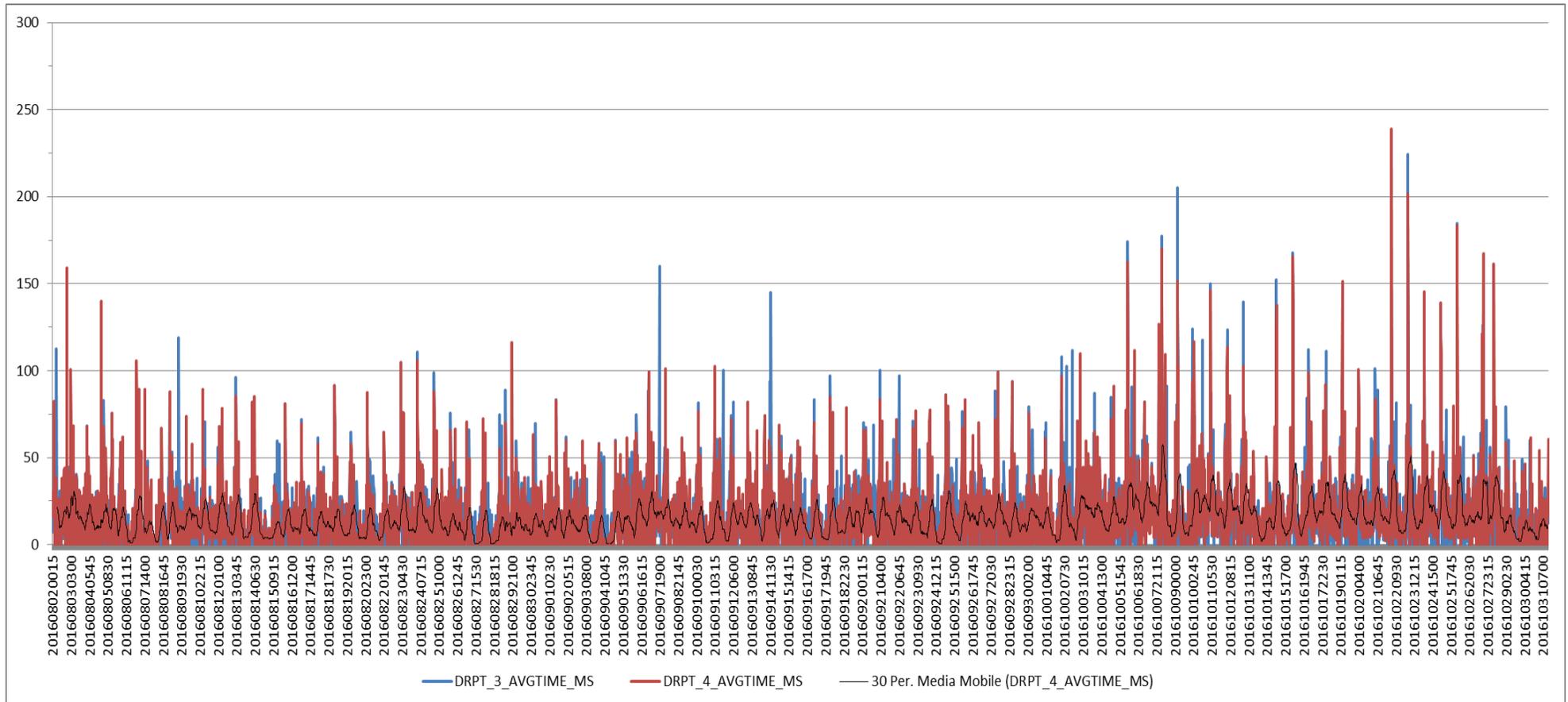
Per ognuna delle due misurazioni (con dati nel tempo per tutti il periodo di osservazione) vengono calcolati il valore di picco massimo (MAX) e il valore medio (AVG) per ogni singolo snapshot Oracle:



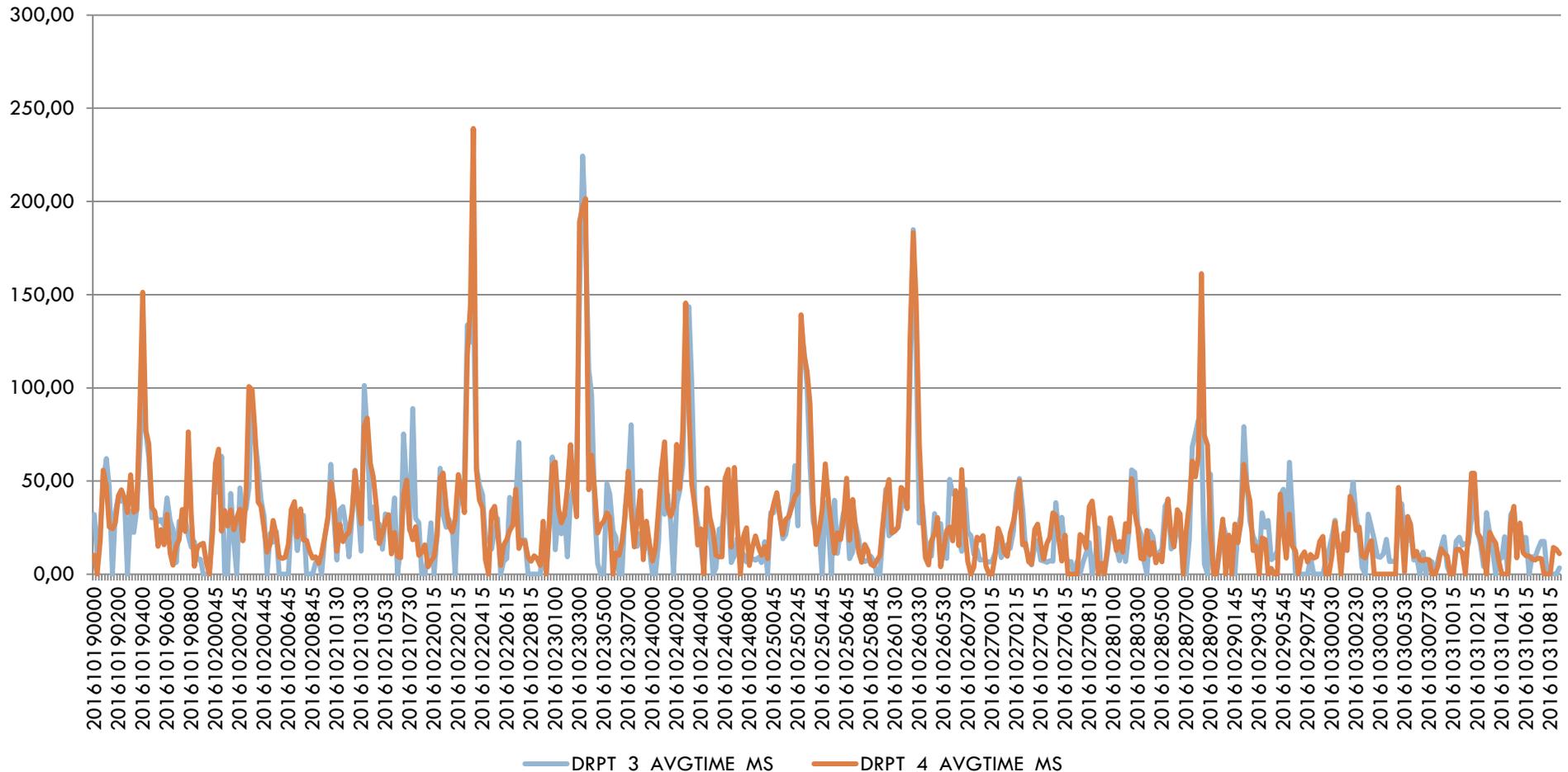
E' possibile monitorare i tempi medi delle varie tipologie di operazioni di I/O:



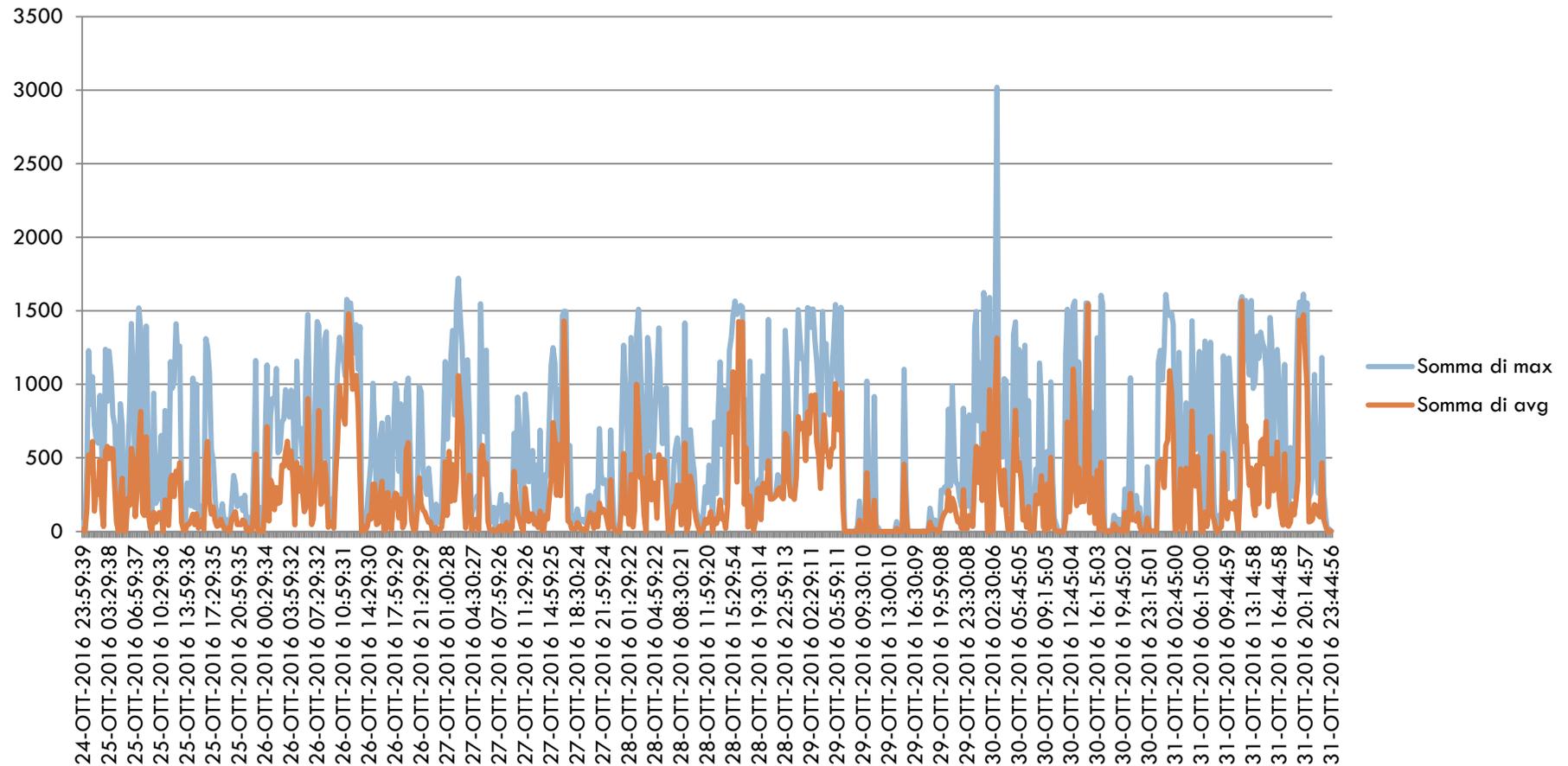
Ecco un altro caso in cui si può notare il degrado della latenza delle letture in physical read (quindi full table scan in parallel, ambito DWH in RAC su due nodi) ed una parziale risoluzione negli ultimi giorni:



Stessi dati della slide precedente ma con una focalizzazione negli ultimi giorni contenenti un primo intervento a partire dal 28 ottobre:



Da tabella Pivot (ecco perché vi è la nomenclatura somma di ...) il throughput in MB/s con un picco dovuto al cambio dell'ora legale ... Sembrerebbe un sistema *cappato*



Quando l'incidenza in percentuale sul DB Time dei wait event riconducibili a letture e scritture sulla TEMP (eventi 'direct path read temp' e 'direct path write temp') è rilevante, risulta utile/indispensabile condurre un'indagine sull'utilizzo di PGA e TEMP.

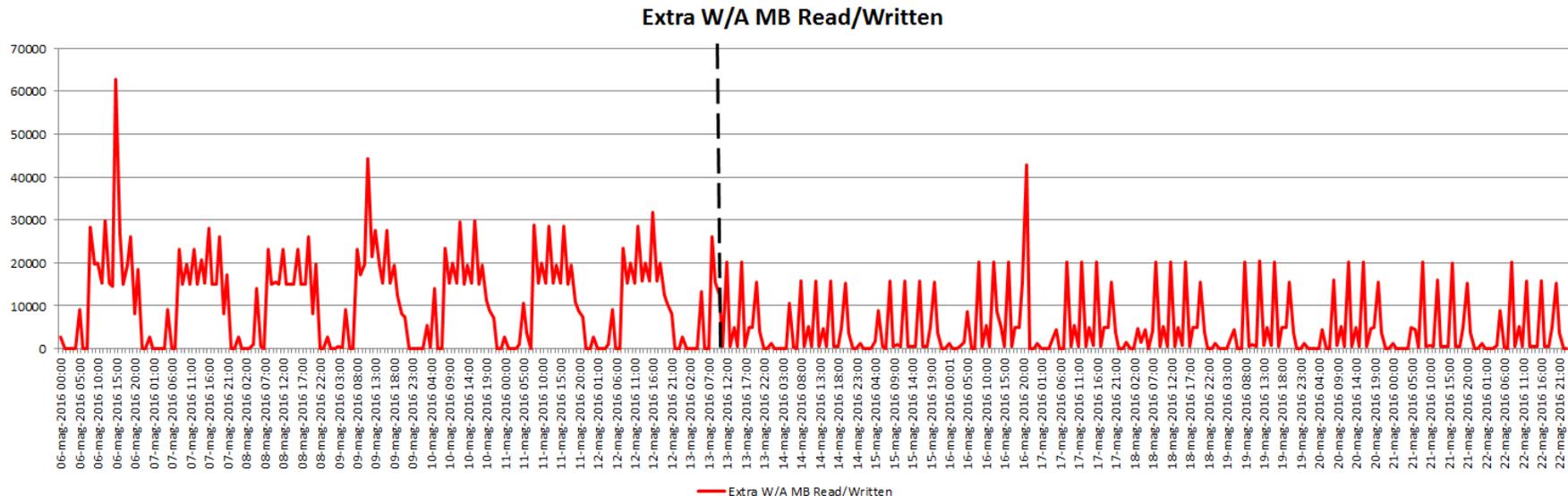
## PGA Aggr Summary

- PGA cache hit % - percentage of W/A (WorkArea) data processed only in-memory

PGA Cache Hit %	W/A MB Processed	Extra W/A MB Read/Written
80.22	977,283	240,949

Il dato esposto in AWR rappresenta un aggregato a livello dell'intervallo temporale del report. E' sempre possibile ottenere con script ad hoc un maggior dettaglio.

## Miglioramento performance SQL statement (use case)



### Interventi eseguiti:

1. Individuazione dei TOP Consumer di TEMP (ASH);
2. Incremento di PGA;
3. Modifica di alcuni hidden parameter su PGA;
4. Workarea\_size\_policy per specifici stmt → MANUAL;
5. Setting parametri hash\_area\_size/sort\_area\_size;

parametro	size	granularity
PGA_AGGREGATE_TARGET	40 GB	
pga_max_size	2 GB	byte
smm_max_size	2 GB	Kb
smm_px_max_size	2 GB	Kb

Il report AWR contiene anche diverse tabelle che riportano i top sql statement ordinati per una misura di riferimento (elapsed time, buffer gets, cpu time, etc) che vengono estrapolati dalla vista V\$SQLSTATS e poi storicizzati nella vista DBA\_HIST\_SQLSTAT.

Elapsed Time (s)	Executions	Elapsed Time per Exec (s)	%Total	%CPU	%IO	SQL Id	SQL Module	SQL Text
15,015.18	34	441.62	27.94	60.89	31.93	<a href="#">7hkttd8cgsdr</a>	JDBC Thin Client	
3,128.21	1	3,128.21	5.82	95.20	2.33	<a href="#">5iqhvxc3qy8d2</a>	DBMS_SCHEDULER	
3,125.88	1	3,125.88	5.82	95.25	2.29	<a href="#">0c34nrz0qkx7</a>	DBMS_SCHEDULER	
2,564.37	6	427.40	4.77	43.83	52.19	<a href="#">6v87n733utz8b</a>	DBMS_SCHEDULER	
2,230.80	5	446.16	4.15	49.72	39.69	<a href="#">abhs44dd6b0d8</a>	JDBC Thin Client	
1,854.36	87	21.31	3.45	22.31	77.37	<a href="#">94hqi5ucc5fhy</a>	JDBC Thin Client	
762.40	28	27.23	1.42	26.52	71.49	<a href="#">gr0411wfrxdif</a>	JDBC Thin Client	
692.38	1	692.38	1.29	10.51	90.69	<a href="#">5w0qffyjum9aw</a>	JDBC Thin Client	
528.64	13	40.66	0.98	32.72	66.13	<a href="#">4drua2sfah6k3</a>	JDBC Thin Client	
396.15	2	198.08	0.74	66.71	29.13	<a href="#">qjjik2j9x5mqa</a>	JDBC Thin Client	

Per avere informazioni di dettaglio su singoli SQL statement è sempre possibile utilizzare i dati AWR in combinazione con i dati ASH.

I soli dati AWR potrebbero non essere esaustivi, in particolare alcuni SQL statement potrebbero non essere campionati o presenti nel report AWR al verificarsi di alcune condizioni, come ad esempio:

- se non vengono superati alcuni valori di soglia;
- se lo statement nel frattempo è sottoposto a flush dallo shared pool;
- l'esecuzione di una DDL (esempio aggiunta di una colonna) con conseguente invalidazione dello statement SQL;
- l'esecuzione di object statistic (che potrebbe invalidare lo statement SQL - a seconda del valore del parametro no\_invalidate).

## SQL ordered by Elapsed Time

In qualche caso i dati potrebbero essere incompleti e quindi non del tutto significativi

- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100
- %Total - Elapsed Time as a percentage of Total DB time
- %CPU - CPU Time as a percentage of Elapsed Time
- %IO - User I/O Time as a percentage of Elapsed Time
- Captured SQL account for 16.2% of Total DB Time (s): 18,388
- Captured PL/SQL account for 10.1% of Total DB Time (s): 18,388

In questi casi l'analisi via ASH potrebbe aiutare a recuperare le informazioni *perse*.

Alcune caratteristiche dei dati ASH:

- ogni secondo vengono catturate in V\$ACTIVE\_SESSION\_HISTORY informazioni sulle sessioni attive;
- le informazioni contenute in V\$ACTIVE\_SESSION\_HISTORY vengono campionate e salvate in DBA\_HIST\_ACTIVE\_SESS\_HISTORY con frequenza di 10 secondi;
- il campionamento è una buona approssimazione della realtà di ciò che accade nel database (1/10 delle informazioni contenute nel buffer circolare V\$ACTIVE\_SESSION\_HISTORY);
- i dati ASH permettono di concentrare l'analisi su una attività specifica (a livello di sessione, o di statement SQL);
- possibilità di condurre analisi su differenti aspetti come:
  - statement che sperimentano degli eventi di wait specifici;
  - ricostruzione delle catene di sessioni bloccate e da chi;
  - livelli di allocazione di PGA e TEMP per sessione e/o statement;
  - carico di lavoro e particolarità per service name;
  - ...

## Segments by Logical Reads

- Total Logical Reads: 617,738,249
- Captured Segments account for 94.5% of Total
- When \*\* MISSING \*\* occurs, some of the object attributes may not be available

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Obj#	Dataobj#	Logical Reads	%Total
DDWH02_SM	TSDWHSMD01	TFCT_FCUD_RC_ANALYSIS_DET	FCUD_ACT	TABLE PARTITION	2557946	3050241	118,018,800	19.10
DDWH02_DIM	TSDWHDIMD01	TDIM_PVAN_VEHICLE_2		TABLE	2577313	3050119	101,542,368	16.44
DDWH01_DW	TSDWHI01	IAK_EXRR_1		INDEX	2580142	2580142	37,597,536	6.09
DDWH00_SA	TSDWHSAD01	TU_PARAMETERS		TABLE	2577477	3047099	26,346,560	4.27
DDWH01_DW	TSDWHD01	TM_PVAN_VEHICLE		TABLE	2576468	2576468	17,048,112	2.76

[Back to Segment Statistics](#)

[Back to Top](#)

## Segments by Physical Reads

- Total Physical Reads: 186,710,515
- Captured Segments account for 87.0% of Total
- When \*\* MISSING \*\* occurs, some of the object attributes may not be available

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Obj#	Dataobj#	Physical Reads	%Total
DDWH02_DIM	TSDWHDIMD01	TDIM_PVAN_VEHICLE_2		TABLE	2577313	3050119	39,275,069	21.04
DDWH02_SM	TSDWHSMD01	TFCT_FCUD_RC_ANALYSIS_DET	FCUD_ACT	TABLE PARTITION	2557946	3050241	16,848,090	9.02
DDWH01_DW	TSDWHD01	TT_PRCS_PRICE_CONDITIONS		TABLE	2576549	2576549	13,964,079	7.48
DDWH01_DW	TSDWHD01	TD_CLIM_CLAIM_POSITION		TABLE	2576338	2576338	8,030,014	4.30
DDWH01_DW	TSDWHD01	TM_PVAN_VEHICLE		TABLE	2576468	2576468	7,555,689	4.05

[Back to Segment Statistics](#)

[Back to Top](#)

## Segments by Physical Read Requests

- Total Physical Read Requests: 6,986,938
- Captured Segments account for 91.7% of Total
- When \*\* MISSING \*\* occurs, some of the object attributes may not be available

Owner	Tablespace Name	Object Name	Subobject Name	Obj. Type	Obj#	Dataobj#	Phys Read Requests	%Total
DDWH02_DIM	TSDWHDIMD01	TDIM_PVAN_VEHICLE_2		TABLE	2577313	3050119	2,553,627	36.55
DDWH02_DIM	TSDWHDIMD01	TDIM_CLHD_CLAIM_HEAD		TABLE	2577481	3050165	495,876	7.10
DDWH02_SM	TSDWHSMD01	TFCT_FCUD_RC_ANALYSIS_DET	FCUD_ACT	TABLE PARTITION	2557946	3050241	445,644	6.38
DDWH01_DW	TSDWHD01	TM_PVAN_VEHICLE		TABLE	2576468	2576468	271,202	3.88
DDWH01_DW	TSDWHD01	TT_PRCS_PRICE_CONDITIONS		TABLE	2576549	2576549	242,707	3.47

Si è cercato di definire cosa sia un assessment ed il suo ciclo di vita.

Un assessment indaga sul comportamento del database e sul suo carico, determina eventuali problemi e suggerisce come migliorare il tutto ed un piano di attuazione di questi suggerimenti.

L'implementazione positiva di questi suggerimenti è la fase finale dell'assessment e ne determina il successo.

Gli strumenti per osservare ed analizzare sono:

- tool Oracle (AWR, ASH, SQL trace, ...);
- query ad hoc su dati campionati, tipicamente via AWR ma è possibile utilizzare lo statspack o anche un campionamento sviluppato ad hoc;
- capacità ed esperienza umana.

Il tempo per mostrare quanto sia possibile analizzare è stato minimale, ma spero che si siano intuite le potenzialità e sia compreso quale possa essere il processo.

Grazie dell'attenzione!



[francesco.renne@icteam.it](mailto:francesco.renne@icteam.it)

Domande?