

基线与容量

徐戟

南瑞集成/子衿技术团队

2016年1月

关于自己

徐戟：网名白鳢，老网虫白鳢 (QQ:62565)

南瑞集成/子衿技术团队 首席架构师

从1992年起从事系统集成行业,曾就职于DEC、赛格计算机、长天集团、联想集团等大型系统集成商

主持开发过全国首套电信级联机实时计费系统；国内首套三检合一的检验检疫综合管理系统；金融大前置平台IPP

出版过3本书：《Oracle DBA优化日记》，《Oracle RAC日记》，《DBA的思想天空》

“信息无障碍研究会” 专业顾问

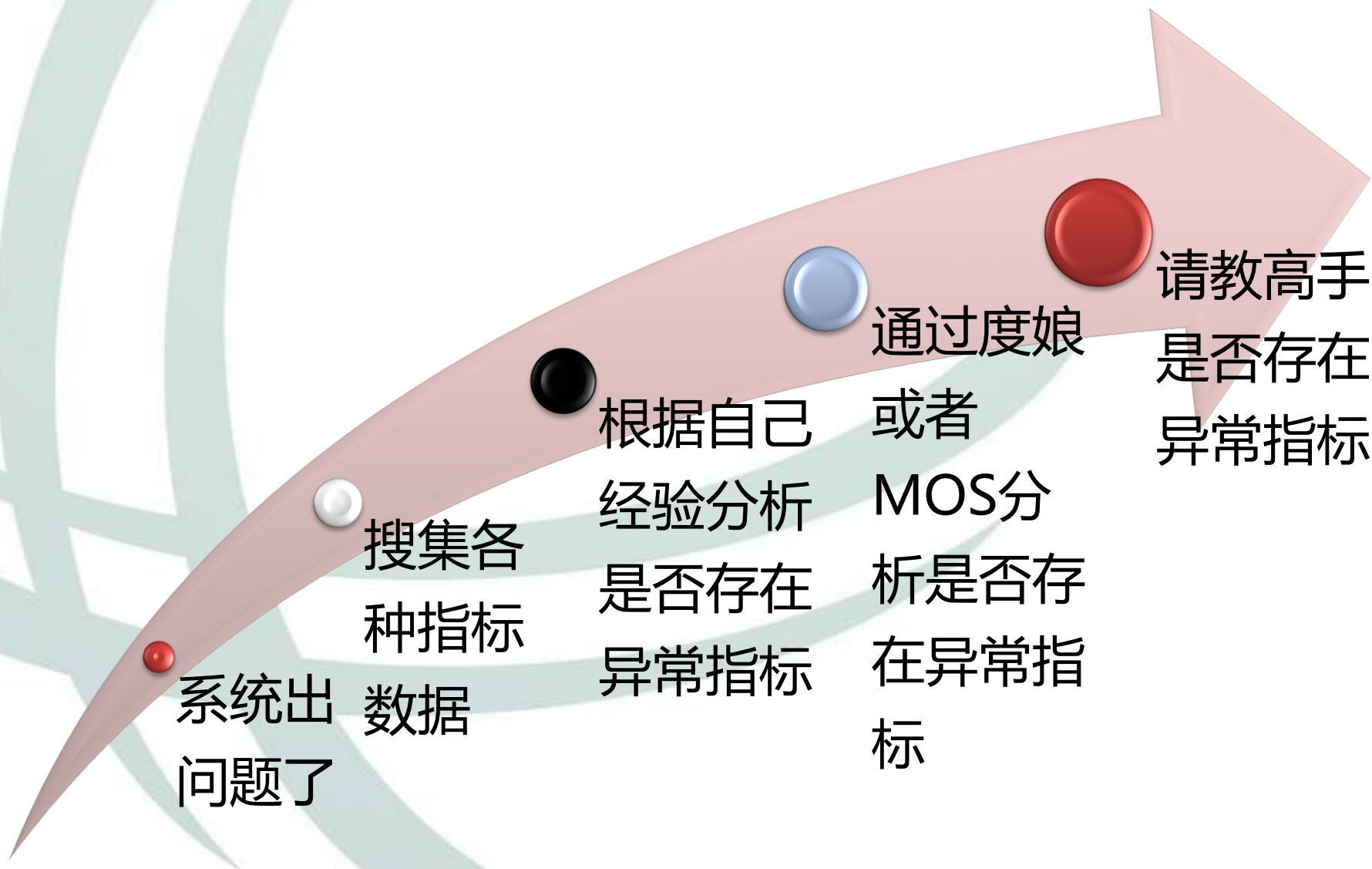
DBA晋级的特征



根据现象分析问题



根据指标分析问题



根据基线分析问题

系统出问题了

采集系统各种指标

通过基线数据分析问题

根据容量分析问题

不仅仅考虑基线

考虑系统的总体负载能力

考虑数据库标准操作产生的资源消耗情况

考虑不同硬件之间的容量差异

考虑信息系统长期发展的可持续问题

系统为什么变慢了？

~~~~~	Per Second	Per Transaction
	-----	-----
Redo size:	943,485.99	5,398.83
Logical reads:	1,885,469.44	10,789.06
Block changes:	5,408.82	30.95
Physical reads:	3,018.65	17.27
Physical writes:	807.43	4.62
User calls:	25,691.83	147.01
Parses:	5,337.02	30.54
Hard parses:	67.79	0.39
Sorts:	8,733.19	49.97
Logons:	0.99	0.01
Executes:	12,025.36	68.81
Transactions:	174.76	

- 8月2日CPU从早上7点30开始到下午5点30都基本上处于100%
- 平时这个系统的CPU使用率不超过40%
- 8月3日起未见异常

## Top 5 Timed Events ~~~~~

Event	Waits	Time (s)	Avg %Total wait (ms)	%Total Call Time	Wait Class
CPU time		194,992		70.0	
db file sequential read	5,140,193	30,020	6	10.8	User I/O
library cache lock	243,658	17,364	71	6.2	Concurrenc
library cache pin	2,437	4,346	1783	1.6	Concurrenc
log file sync	629,952	3,091	5	1.1	Commit



# 可能的原因

---

- 应用出了问题？
- 共享池争用？
- 某个SQL出现问题？
- 存储系统出现问题？
- 服务器出现问题？
- .....

# 关注点-根据指标分析

~~~~~	Per Second	Per Transaction
	-----	-----
Redo size:	943,485.99	5,398.83
Logical reads:	1,885,469.44	10,789.06
Block changes:	5,408.82	30.95
Physical reads:	3,018.65	17.27
Physical writes:	807.43	4.62
User calls:	25,691.83	147.01
Parses:	5,337.02	30.54
Hard parses:	67.79	0.39
Sorts:	8,733.19	49.97
Logons:	0.99	0.01
Executes:	12,025.36	68.81
Transactions:	174.76	

Top 5 Timed Events ~~~~~			Avg %Total wait Call		
Event	Waits	Time (s)	(ms)	Time	Wait Class
	-----	-----	-----	-----	-----
CPU time		194,992		70.0	
db file sequential read	5,140,193	30,020	6	10.8	User I/O
library cache lock	243,658	17,364	71	6.2	Concurrenc
library cache pin	2,437	4,346	1783	1.6	Concurrenc
logfile sync	629,952	3,091	5	1.1	Commit

关注点-根据基线进行分析

~~~~~	Per Second	Per Transaction
	-----	-----
Redo size:	943,485.99	5,398.83
Logical reads:	1,885,469.44	10,789.06
Block changes:	5,408.82	30.95
Physical reads:	3,018.65	17.27
Physical writes:	807.43	4.62
User calls:	25,691.83	147.01
Parses:	5,337.02	30.54
Hard parses:	67.79	0.39
Sorts:	8,733.19	49.97
Logons:	0.99	0.01
Executes:	12,025.36	68.81
Transactions:	174.76	

Top 5 Timed Events			Avg %Total		
~~~~~			wait	Call	
Event	Waits	Time (s)	(ms)	Time	Wait Class
	-----	-----	-----	-----	-----
CPU time		194,992		70.0	
db file sequential read	5,140,193	30,020	6	10.8	User I/O
library cache lock	243,658	17,364	71	6.2	Concurrenc
library cache pin	2,437	4,346	1783	1.6	Concurrenc
logfile sync	629,952	3,091	5	1.1	Commit

关注点-根据容量进行分析

~~~~~	Per Second	Per Transaction
	-----	-----
Redo size:	943,485.99	5,398.83
Logical reads:	1,885,469.44	10,789.06
Block changes:	5,408.82	30.95
Physical reads:	3,018.65	17.27
Physical writes:	807.43	4.62
User calls:	25,691.83	147.01
Parses:	5,337.02	30.54
Hard parses:	67.79	0.39
Sorts:	8,733.19	49.97
Logons:	0.99	0.01
Executes:	12,025.36	68.81
Transactions:	174.76	

Top 5 Timed Events ~~~~~			Avg %Total wait Call		
Event	Waits	Time (s)	(ms)	Time	Wait Class
	-----	-----	-----	-----	-----
CPU time		194,992		70.0	
db file sequential read	5,140,193	30,020	6	10.8	User I/O
library cache lock	243,658	17,364	71	6.2	Concurrenc
library cache pin	2,437	4,346	1783	1.6	Concurrenc
logfile sync	629,952	3,091	5	1.1	Commit

**业务部门的人突然说某个业务很慢，而你发现系统资源都没有瓶颈，数据库也没有什么指标明显异常，你该如何进一步分析？**

---

## 从业务运行基线入手

---

## 从业务模块下钻到SQL

---

## 分析核心SQL的基线

---

检查核心SQL的的执行次数，每次执行的CPU，IO等开销以及执行计划等与基线相比是否有变化

# IT苦力的做法

先做个AWR报告，看看有没有有问题的SQL

通过各种手段问清楚相关业务是否和这条SQL有关

做个SQL AWR报告

分析执行计划，看看有没有问题



---

对于系统而言，从运维的角度来说，基线-BASELINE是某个运行状态在某个时间上的快照

---

对于DBA来说,基线是用来判断某个指标是否正常的参考条件

---

某个指标、某个运维经验、某个现象都可以成为运维基线

---

对DBA来说，有价值的基线往往来自于长期运维的经验，而不是简单的某个指标

# 有价值的基线-举例

---

某个系统，活跃会话数超过100，系统资源就会紧张，超过200，系统就会有问题

---

某条查询语句在某一个时刻并发执行数量超过100，系统就有可能出现性能问题

---

DBTIME/采样时间大于100说明系统中某些常用SQL的执行计划出现了错误

---

LOG FILE SYNC超过10毫秒，系统就容易HANG

# 基线其实很简单

---

基线是DBA对系统的看法和经验

---

DBA往往会把正常的系统的某些指标归纳为基线

---

基线可以通过积累来获得和完善

---

基线的采集可以通过多种方式进行，自动，手动均可

# 基线管理不简单

理论上你可以把系统中的所有指标都作为基线管理起来

实际上充分“理解”的基线才对运维有意义

充分“理解”系统，然后才能总结出“有价值”的基线

基线的管理必须借助自动化手段

定期进行趋势分析，可能防患于未然

对违背基线规则的事件需要进行闭环管理

# 采集数据库基线的办法

---

- 手工脚本
- 自动化工具，比如EM
- 通过  
DBMS_WORKLOAD_REPOSITORY.CREATE_BASELINE

# 把整个AWR保存为基线

---

- 找台机器，建个AWR历史数据库
- 使用awrextr.sql定期导出AWR数据
- 使用awrload.sql导入AWR历史数据库（可以存放多个数据库的AWR资料）
- 使用awrrpti选择所需的数据库生成报告
- 通过自定义脚本直接分析AWR数据

# 基线的使用

## 故障应急分析

- 系统出现问题时的采集数据和基线数据进行对比
- 对于无法定位的问题，通过整体指标对比进行分析，比如 awrddrpt

## 趋势分析

- 当采集了足够数量的采样点后，趋势分析就成为可能
- 趋势分析可以以周、月、年为单位进行
- 对于持续恶化的指标要进行针对性分析

## 预警

- 积累了足够的采样后，可以框定某个指标的低阈值、高阈值和平均值
- 对于超过高低阈值的现象要进行闭环管理

# 基线预警与分析

---

基线预警可以根据某个单项指标的高低阈值简单设置

---

数据分析不能仅仅依靠某个指标简单的进行

---

关联性数据分析十分重要，比如共享池问题，需要分析共享池使用率、解析情况、执行情况、相关门锁丢失率、SGA RESIZE情况、库缓冲指标、字典缓冲、执行/解析较高的SQL等综合情况才能得到完整的分析结果



## 需要注意的一点

---

- 虽然使用工具可以让基线分析工作自动化，但是无目的的分析效果不佳
- 如果每天有一两个预警，那么我们可以很好的闭环分析，每天几十甚至上百的预警实际上对DBA来说意义不大
- 开始时候不要贪多，逐渐掌握更多的指标



已知系统的基线搞清楚了  
但是没有历史积累的系统怎么办

- 不同CPU之间的性能差异是什么样的？
- 我们新买的高端存储能提供什么样的IO能力？
- 升级了存储后一直存在问题，是存储本身的问题还是配置存在问题？
- 割接了几个省的系统，CPU使用率就100%了，问题出在哪了？

# 系统级容量基线

- 一块15000rpm的SAS盘的IOPS指标为150-200
- 一个磁盘块的物理IO延时大约在2-5毫秒
- 对OLTP系统而言，高端存储的CACHE命中率在60-70%
- 一个INTEL E7芯片CPU每秒可以处理300-500M的IO
- 一个千兆以太网在RAC INTERCONNECT上的流量大概80M/秒，而万兆以太网可以达到850M/秒
- .....

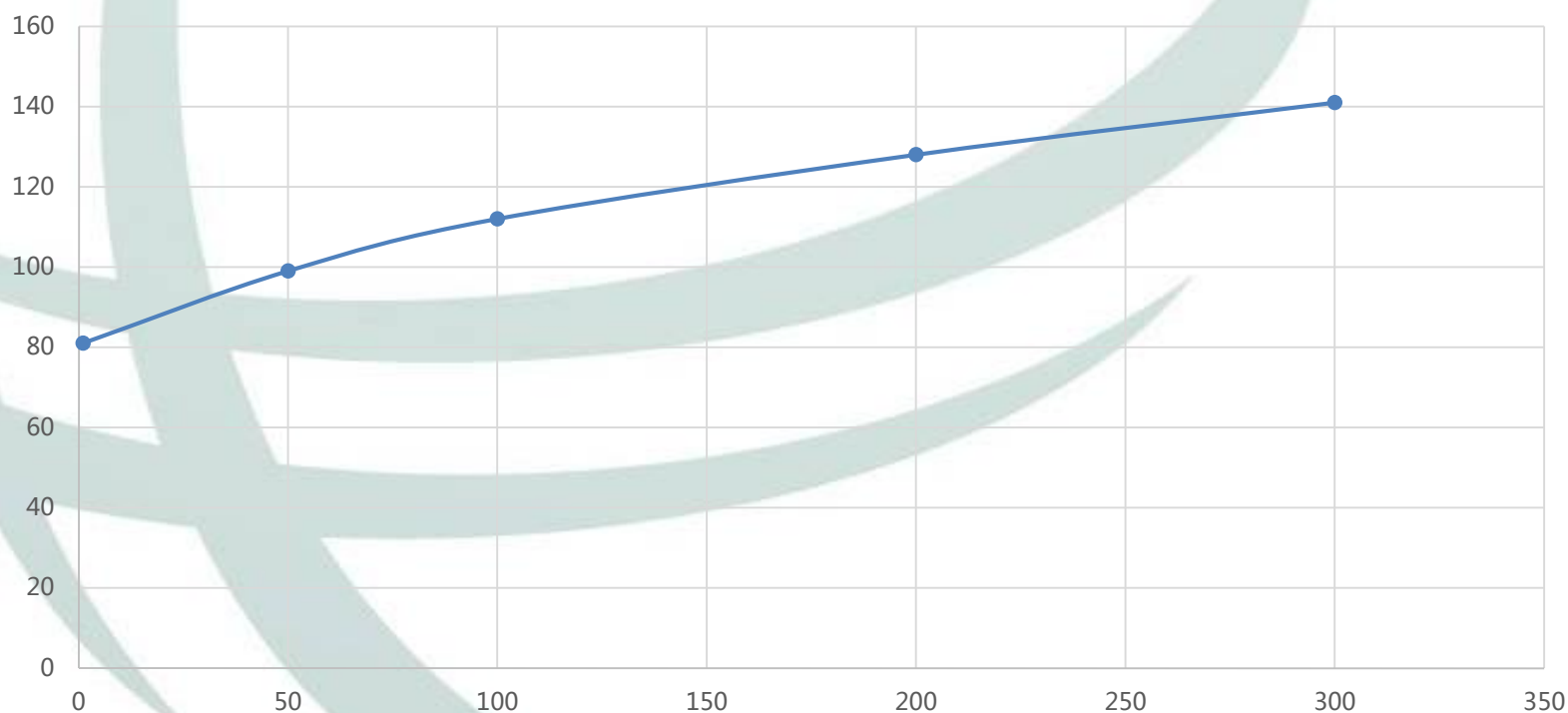
# Oracle数据库容量基线

- 一般OLTP系统的DB CACHE命中率在98%以上
- db file sequence read:在一个正常的OLTP系统中，占比超过50%，甚至更高，平均响应时间4毫秒左右，不超过10毫秒可以认为正常，超出20毫秒可能对系统产生较大影响
- Log file sync：在一个正常的系统中，小于4毫秒，如果使用中高 端存储，并且存储负载不是很高，该指标为1-2毫秒
- Ave global cache get time (ms)：一般在负载不是很高的系统中，为1-4毫秒，如果超过20毫秒，说明存在问题

# SQL容量基线

```
select count(1) as count from (SELECT c.zpxmj1  
xmjl, a.htid, a.htmc, a.htbh, a.khmc, b.xmjc, a.xsddh, a.ssgs, TO_CHAR(a.htqdsj, 'yyyy-mm-dd')  
htqdsj, a.hte, a.erpxmbh, a.htzt, a.khfzr, TO_CHAR(a.htlyjfsj, 'yyyy-mm-dd') htlyjfsj, TO_CHAR(a.htlyjssj, 'yyyy-mm-  
dd') htlyjssj, c.xdbmid FROM bhtxx a LEFT JOIN bxmjbxx b ON a.xmid = b.xmid LEFT JOIN bhtxd  
c ON a.htid = c.htid WHERE NVL(a.xmid, '-1') = '-1' AND (NVL(phtid, '-1') <> '-1' OR NOT EXISTS  
(SELECT * FROM bhtxx c WHERE c.phtid = a.htid)) and c.htxdid is not null and a.ssgs = 'JY '  
ORDER BY a.htid desc) a
```

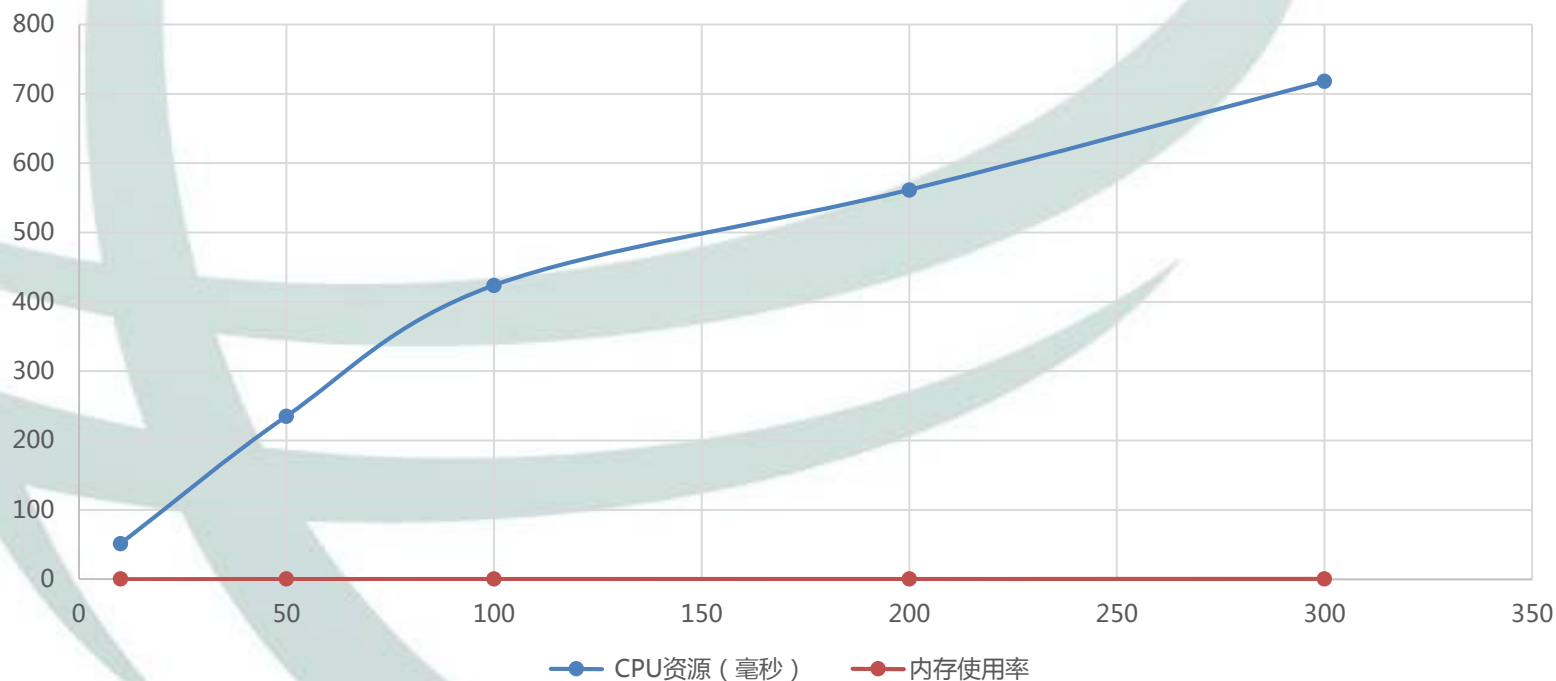
CPU资源 ( MS )



# 业务模块容量基线

序号	并发用户数		CPU资源（毫秒）	内存使用率
1	10		51.2	41.78%
2	50		235.2	41.83%
3	100		424	41.85%
4	200		561.6	41.86%
5	300		718.4	41.92%

模块容量曲线



# 系统基线的复杂性

---

- 不同类型的系统在系统级基线上表现差异较大
- 比如SQL单次执行时间是否超过1个CPU周期，在CPU相关基线上表现差异会较大
- 较为准确的系统级基线需要通过日常积累甚至专业分析才能获得



# 通过一个例子看IO

Load Profile Per Call ~~~~~	Per Second	Per Transaction	Per Exec
-----	-----	-----	-----
DB Time(s) : 0.01	19.5	0.6	0.00
DB CPU(s) : 0.00	7.8	0.3	0.00
Redo size:	1,088,428.3	35,735.8	
Logical reads:	710,902.8	23,340.7	
Block changes:	6,239.2	204.9	
Physical reads:	17,835.5	585.6	
Physical writes:	455.7	15.0	
User calls:	3,293.3	108.1	
Parses:	748.3	24.6	
Hard parses:	1.7	0.1	
W/A MB processed:	26.1	0.9	
Logons:	0.4	0.0	
Executes:	8,863.2	291.0	
Rollbacks:	0.5	0.0	
Transactions:	30.5		

# 通过一个例子看IO

Function Name	Reads: Data	Reqs per sec	Data per sec	Writes: Data	Reqs per sec	Data per sec	Waits: Count	Avg Tm(ms)
Direct Reads	457.1G	1061.4	129.632	3M	0.1	.000830	0	N/A
Buffer Cache Re	39.2G	1112.7	11.1082	0M	0.0	0M	3864.5K	3.7
DBWR	0M	0.0	0M	13G	288.4	3.68205	0	N/A
Others	5G	6.3	1.42823	5.2G	6.9	1.48113	22.8K	3.6
LGWR	1M	0.0	.000276	3.9G	97.0	1.09783	236.9K	3.5
Direct Writes	1M	0.0	.000276	68M	0.2	.018832	0	N/A
Streams AQ	0M	0.0	0M	0M	0.0	0M	1	0.0
TOTAL:	501.3G	2180.3	142.169	22.1G	392.6	6.28068	4124.3K	3.7

---

A stylized, light blue globe is positioned on the left side of the slide, partially cut off by the edge. It features several curved lines representing latitude and longitude. The text 'Q&A' is centered over the right side of the globe.

**Q&A**