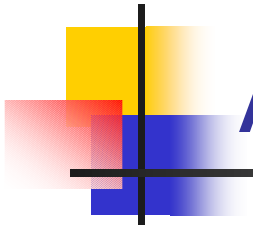
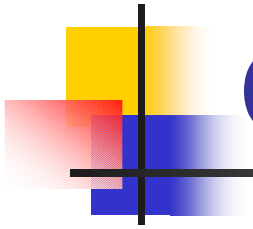
Quantitative Modeling and Capacity Planning with Oracle Load-Factors

Brahmaiah Jarugumilli & Shankar Jayaganapathy



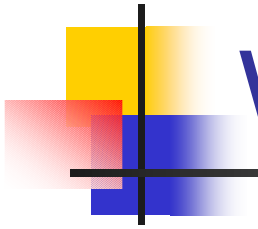
Agenda & Objectives

- Understand System workload
- Collecting workload statistics
- Distinguishing load from effect
- Load patterns and dimensions
- Correlation between load-factors
- Putting workload characteristics to use
- Stress Tests- A New Approach



Overview Of Workload

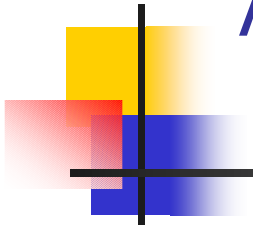
- What is system workload?
- How is a system’s workload influenced?
- Why should we characterize a system’s workload?



What is System Workload?

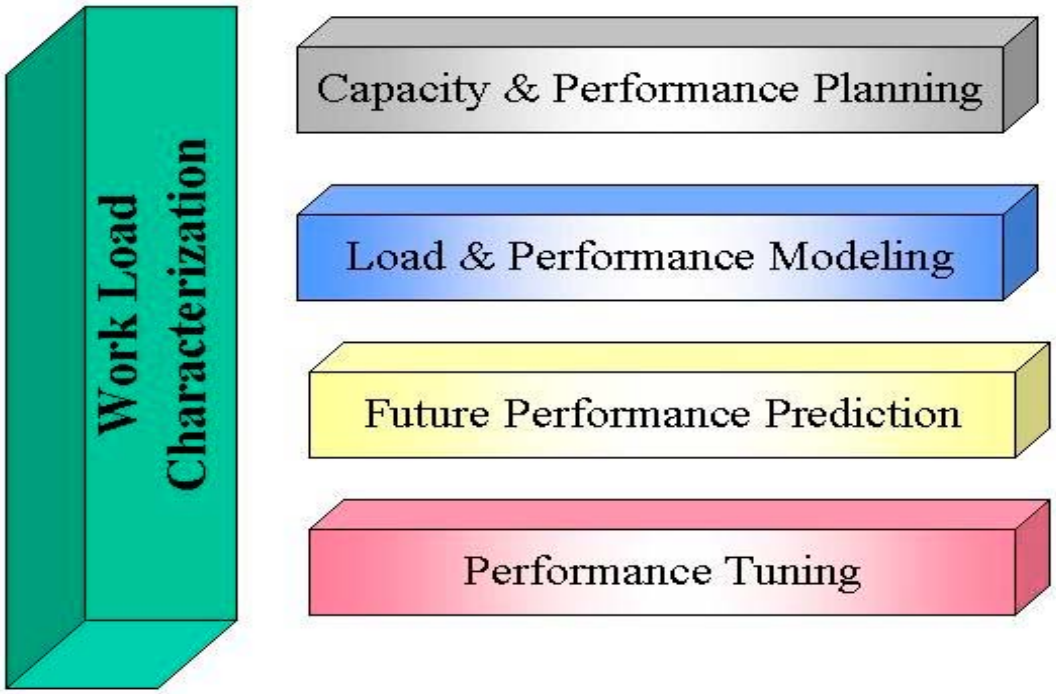
- The demand placed on a system’s resources in order to do the assigned work can be defined as workload. In a database system, the demand arises from queries and transactional requests (commits and rollbacks), whose conglomerate could be observed in terms of CPU, Memory and I/O related resource consumption or usage.

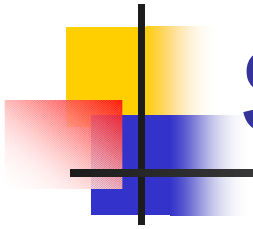
A few **“influential factors”** -



- End users and their behavior- intensity with which work is done
- Concurrency of online users and the mix of the programs they are executing
- Nature of the End-user layer of the application - like various Forms, canned reports vs. Parameterized Reports, etc.
- Application execution architecture- multiple tiers, proxy serviceability, threads of execution, serial vs. parallel execution, etc.
- Security and audit features
- Layers, levels and sizes of caching- at Oracle, OS, Disk system, Network(JDBC caching), App server(JAVA caching), web server(HTTP Caching), client cache, etc.
- Data models and nature of the data - like volatility, reuse, currency, READ vs. READ-WRITE nature of business transactions, etc.

Why Characterize system Workload?

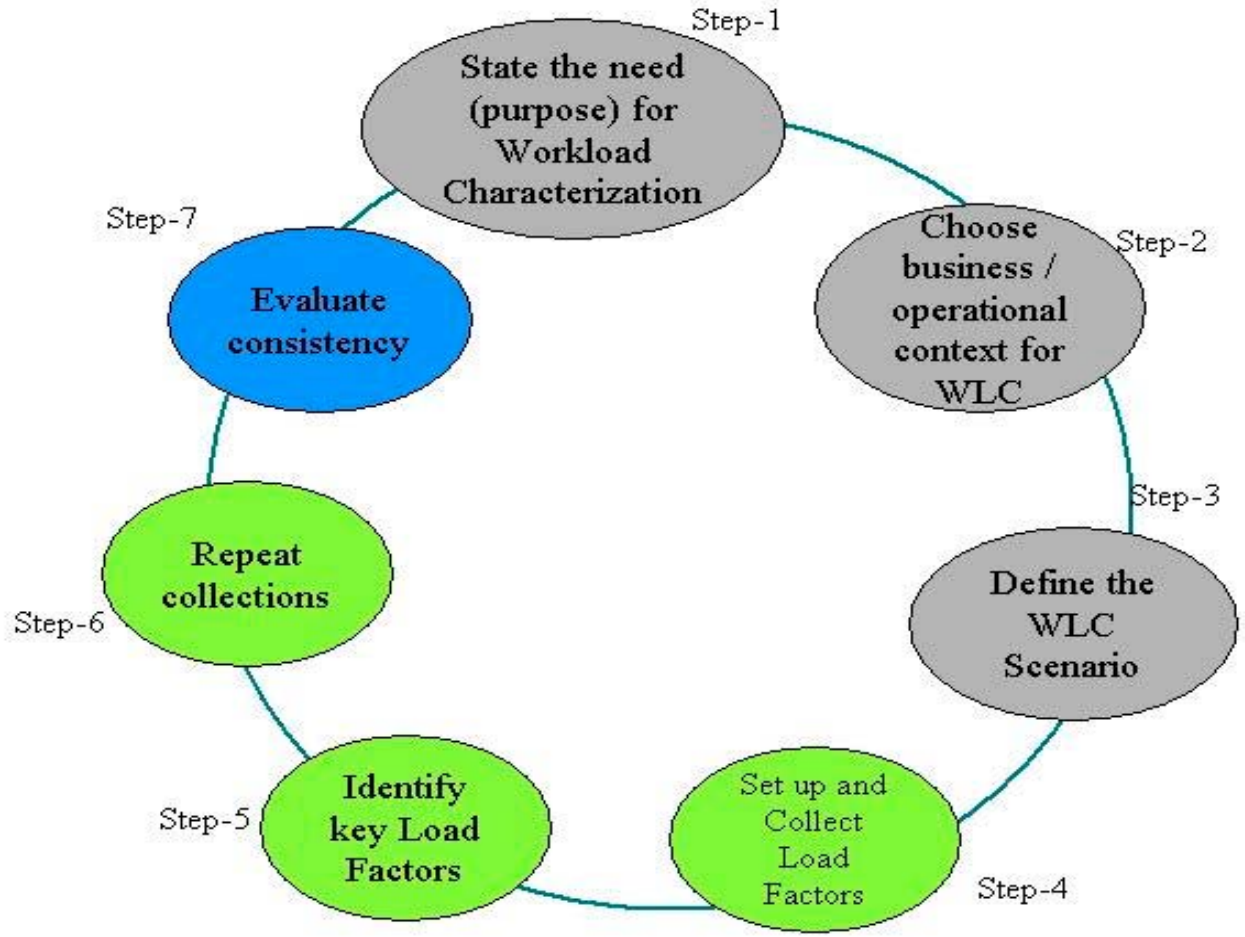


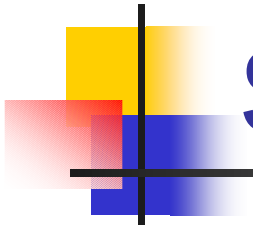


Statistics Collection

- Tactical issues with statistics collection

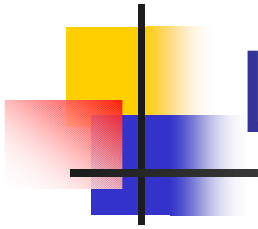
Tactical Issues in Collecting Statistics





Source of Statistics

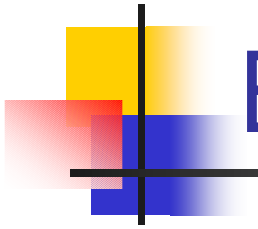
- Some common views that provide database statistics are:
  - v$sysstat,
  - v$sesstat,
  - v$sgastat,
  - v$pgastat,
  - v$rollstat,
  - v$undostat,
  - v$filestat,
  - v$latch and
  - v$session_wait.



Load and Effect

- **Load and its Dimensions:**
  - Oracle system statistics reflect and represent system load.
  - Workload in a system essentially is multi-dimensional and is a complex non-linear addition of them.

- **Load Effects:**
  - Oracle wait events are significant in observing the resultant effects on the system under a given load.
  - The types of queues and amount of waiting will fall into different categories (enqueues, child latches, etc.) and will also vary in terms of waited time (nanoseconds to microseconds to seconds.



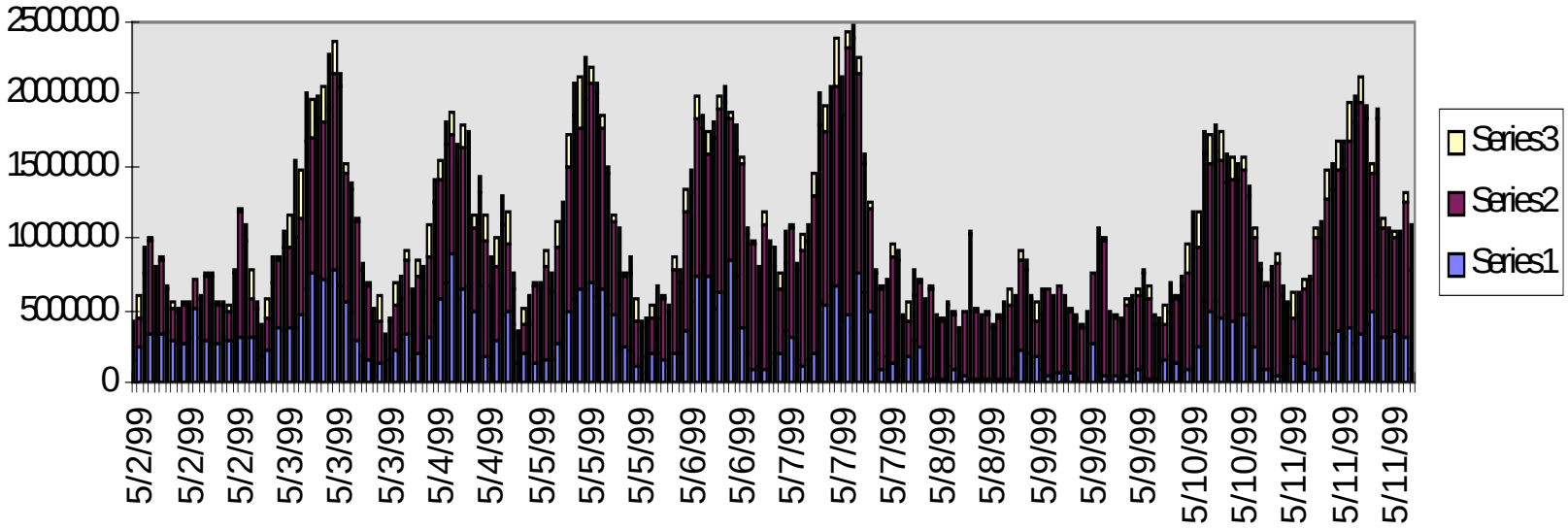
Environment Specific Waits

- An OLTP database will predominantly spend most of its time waiting on certain specific resources (latch free, log file sync, db file scattered read etc.)

- These may be different for a data warehouse (DWH) where parallel query waits are more common (PX Deq: Execute Reply, PX Deq: Execution Msg etc.).

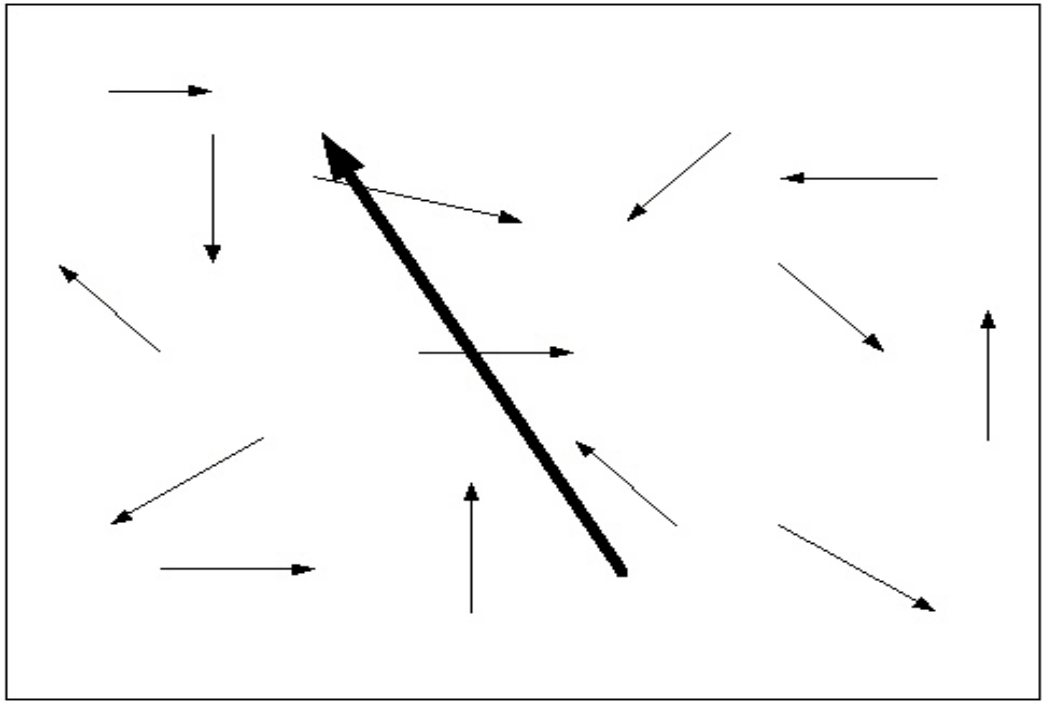
Load Patterns

Intensity of Physical Reads during the course of a day



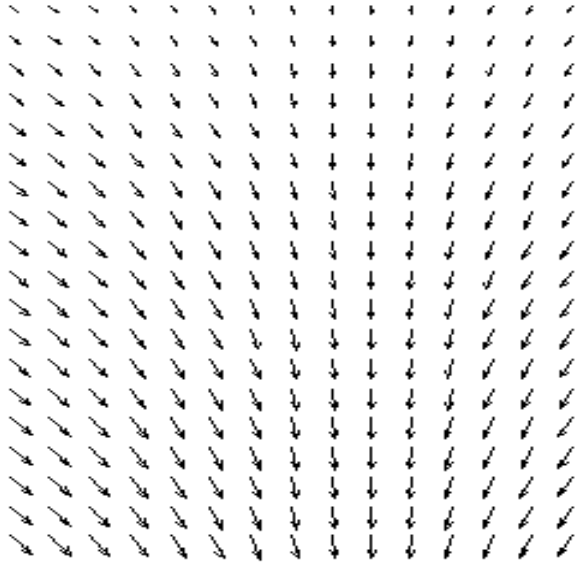
Meaningful Correlation Between Load-factors

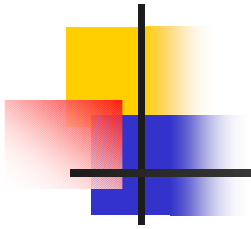
Vectors in a vector space—Example 1



Meaningful Correlation Between Load-factors

Vectors in a vector space—Example 2.





Identifying Relationships

Based on work :

DBWR related activity
PHYSICAL Writes
DB Block changes
REDO activity
UNDO segment activity
Transaction table writes.

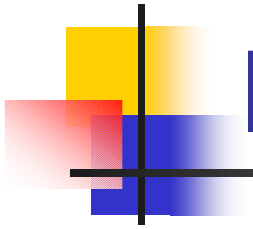
Based on a class of work (Ex. Class #2):

redo blocks written
redo entries
redo log space wait time
redo size
redo synch writes
redo write time
redo writes
redo buffer allocation retries
redo log space requests
redo ordering marks
redo synch time
redo wastage
redo writer latching time

Regression Analysis to Determine Relationships-1

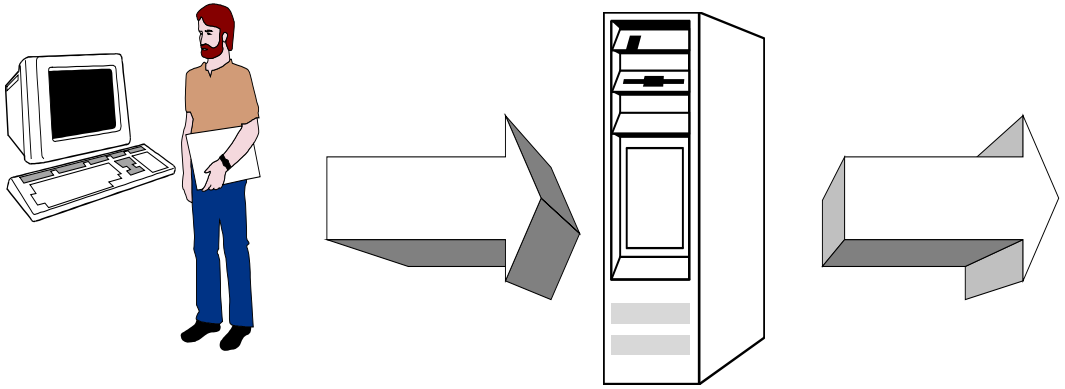
- Statistical regression modeling
  - helps to build a mathematical model based on cause and effect.
- Traditional Variance analyses
  - are inappropriate to analyze a complex system such as an Oracle database, which has layers of functionality. These layers represent different technical operational structures are independent but at times follow a daisy chain like activity-precedence approach.

Regression Analysis to Determine Relationships-2



- Independence and Interindependence
  - Independence is prevalent (consistency management, bulk or highly selective index fetching) at the same time as Interdependence (physical disk reads as a result of failure to cache a data block)
- Oracle load-factors exhibit both:
  - determinism
  - spontaneity
- Model Strength
  - quantified through the R-Squared value of the regression model

The Scaling Model- Regression



Users and Work-stations growth
PLUS
Application behavior
PLUS
User behavior on the work-station
=
Active user proxy variable

10- 12 DRIVERS that drive the consumption and queueing of resources ultimately determining the various counters in t he DB reflecting as System-Statistics

100 other statistics in Oracle DB that are muti-dimensional representations of the data server load

Regressions Equations- Set-1

Regressions Equations- Set-2

Regressions Equations- Set-3

An Illustration

Multiple R	0.964626							
R Square	**0.930504**							
Adjusted R Square	0.92151							
Standard Error	0.092282							
Observations	97							
ANOVA								
	Df	SS	MS	F	Significance F			
Regression	11	9.691956	0.881087	103.4625	**2.38E-44**			
Residual	85	0.72386	0.008516					
Total	96	10.41582						
	Coefficients	Standard Error	t Stat	P-value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
Intercept	-3.10908	0.971842	-3.19916	0.001938	-5.04136	-1.17679	-5.04136	-1.17679
X Variable 1	-1.63339	0.225192	-7.25333	1.76E-10	-2.08113	-1.18565	-2.08113	-1.18565
X Variable 2	-0.01199	0.041317	-0.29028	0.772311	-0.09414	0.070156	-0.09414	0.070156
X Variable 3	-0.15072	0.065202	-2.31161	0.023219	-0.28036	-0.02108	-0.28036	-0.02108
X Variable 4	-5.02366	1.662474	-3.0218	0.00332	-8.32911	-1.71821	-8.32911	-1.71821
X Variable 5	-0.64851	0.237838	-2.7267	0.007769	-1.1214	-0.17563	-1.1214	-0.17563
X Variable 6	2.722918	0.158554	17.1734	2.25E-29	2.407669	3.038167	2.407669	3.038167
X Variable 7	0.383586	0.123069	3.116834	0.002494	0.138891	0.62828	0.138891	0.62828
X Variable 8	-0.00618	0.186118	-0.03318	0.973609	-0.37623	0.363877	-0.37623	0.363877
X Variable 9	5.524968	1.780838	3.102455	0.002605	1.984184	9.065753	1.984184	9.065753
X Variable 10	0.111868	0.04582	2.441486	0.016701	0.020766	0.20297	0.020766	0.20297
X Variable 11	-0.09114	0.063939	-1.42541	0.157701	-0.21827	0.035989	-0.21827	0.035989

“CrVAL = -3.109 + [X1 *(-1.633)] +...+ [X6 * (2.72)] +….[X9 * (0.056)]…

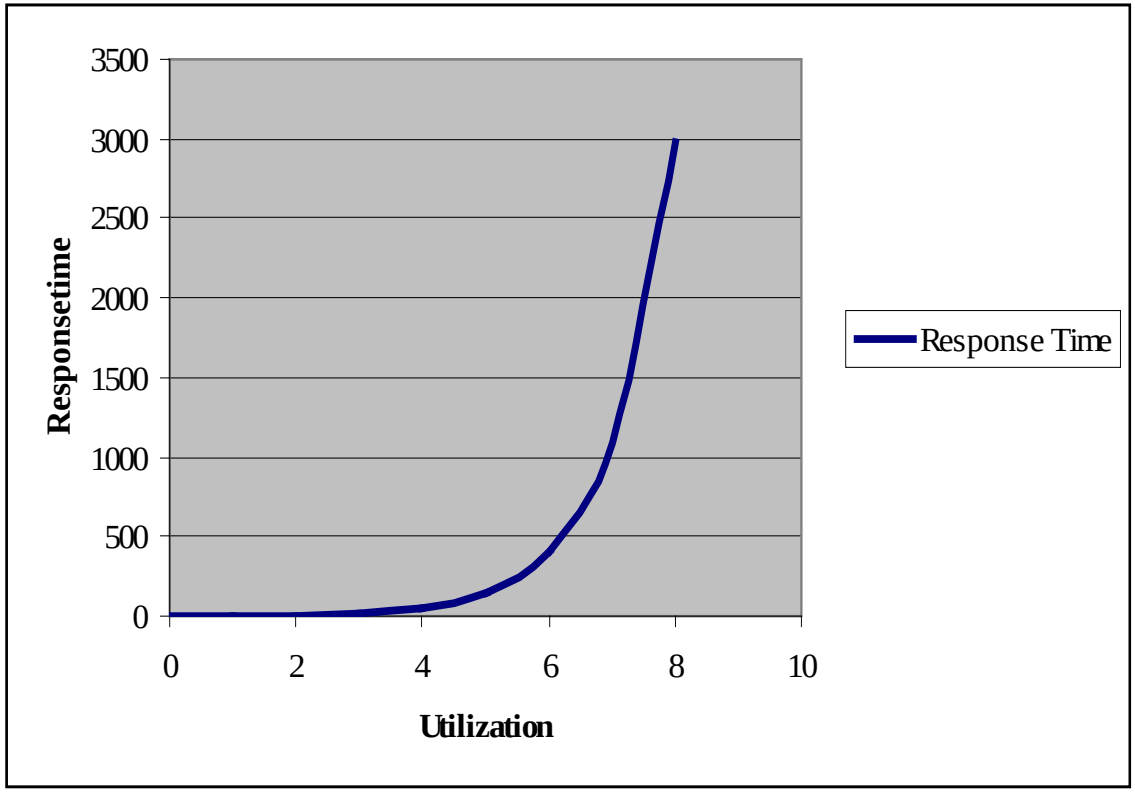
Use of Log Values-1

- Linear and non-linear relationships
  - relationships are not always linear due to their complex interactions
- Non-linear relationships
  - Require non-linear regression techniques
  - Use of log values

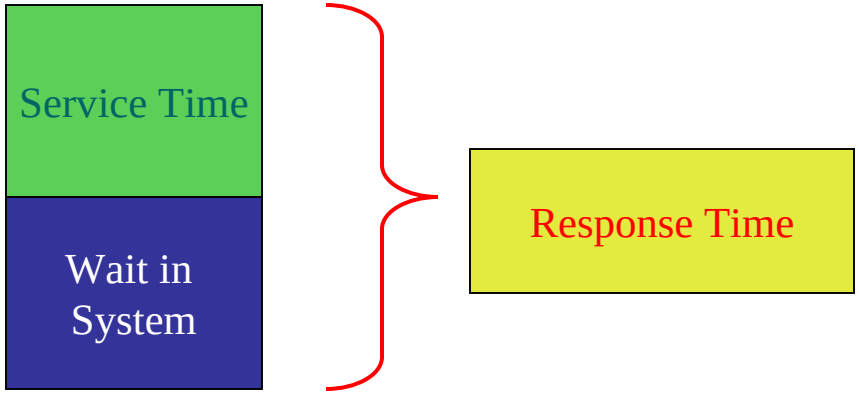
Use of Log Values-2

- Logarithmic values of the variable reduce the non-linearity to linearity between log predicates
- Non-linear models are more reliable because they address both:
  - linear
  - non-linear behavior

Developing Capacity Models

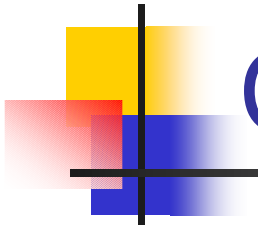


Distinguish Between Work Load and Wait Events



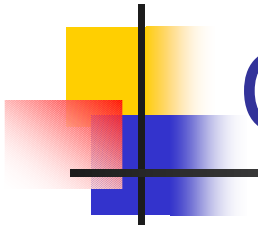
- Service Time is a conglomerate of ‘inescapable’, networked response times
- This is the minimum time from the most optimal ‘way’ the work [eg: FTS vs. Index Scans]

Time Window ÷ Response Time = Work Done (Throughput)



QT to the Rescue

- Queuing Theory
  - Universal Applicability
    - Grocery store to Computer Science
  - Considerable research and publications
  - Queues form because of finite resources

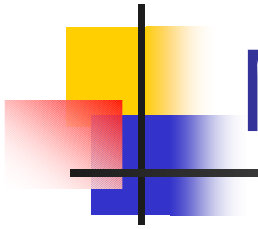


QT- Foundations

- Queueing Theory
  - Essential Components of Queues
    - λ - The Arrival rate of jobs into the system
    - μ - The service rate at a resource counter
    - U – The utilization of the resource
  - Fundamentally,
    - $U = \lambda / \mu$
    - $\lambda / \mu < 1$, Queue builds up
    - Response time $(R) = 1 / [\mu (1 - U)]$

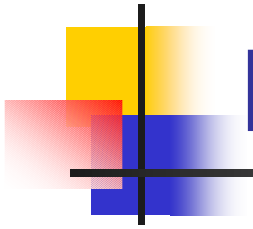
QT- Types of Queues

- Types Of Queues
  - M/M/m – Random Arr, Svc, m- counters
  - M/M/1
  - M/M/∞
  - M/M/m/B – Buffrd jobs, no Arr beyond ‘B’
  - G/G/m - Known rates, like Mfg systems



MVA—Mean Value Analysis

- What is MVA?
  - Simplified determination of queue behavior and ignores variances in response times.
  - Adopts and iterative computational algorithm
  - Closed queues assumed



MVA—Mean Value Analysis

- Formulas
  - N - number of jobs or users in the system.
  - m - number of devices in the queue system
  - Z - think time
  - S_i - is the service time at i^{th} device
  - V_i - is the number of visits to the i^{th} device
  - X - system throughput = N / (Z + R)
  - R - Response time = $\sum_1^m (R_i * V_i)$
  - R_i - R at i^{th} device = $S_i * (1+Q_i)$
  - Q_i - jobs in the queue at i^{th} device = $X * V_i * R_i$

MVA—Mean Value Analysis

- Algorithm

Initialize the run for i = 1 to m with $Q_i = 0$;

FOR n = 1 to N LOOP

BEGIN

For i = 1 to m LOOP

$R_i = \sum S_i * (1 + Q_i)$

$R = {}_1\sum^m (R_i * V_i)$

$X = N / (Z + R)$

For i = 1 to m

$Q_i = X * V_i * R_i$

END

Throughput at ith device $X_i = X * V_i$ and

Utilization $U_i = X * S_i * V_i$

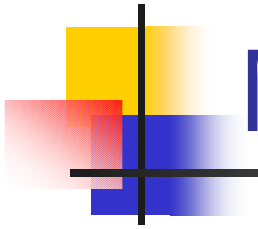
Developing Capacity Models

All values gathered and regressed over 10 Minute intervals

stat name	**CPU RELATED**		
USER CALLS	1684	17777	129339
CURSOR AUTHENTICATIO	68	598	4383
EXECUTE COUNT	454	4588	32905
OPENED CURSORS CUMUL	70	620	4539
PARSE COUNT	84	703	5092
RECURSIVE CALLS	97	532	3536
SORTS (ROWS)	14470	15563	26063
CPU Used (seconds)	75.3	284.1	2218.3
Number of CPUs	1	1	6
Projected Utilization	12.55	47.35	61.62
CPU used per sec	**0.1255**	**0.4735**	**3.6971667**
Simulated # of Jobs/Sec	**1**	**4**	**29.00**

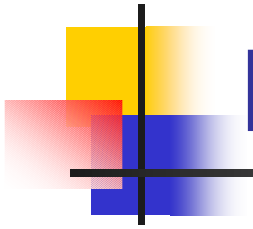
Developing Capacity Models

Simulation Number	μ	λ	ρ	m	Probability of zero jobs in Queue	Prob for 5 jobs	Prob for 20 jobs	Jobs in Queue	Utilization	Mean Response time	Mean waiting time
1	8	1	0.125	1	0.875	3.34E-06	1.02E-10	0.017857	12.5	0.142857	0.017857
2	8	4	0.5	1	0.5	0.007813	0.000244	0.5	50	0.25	0.125
3	8	6	0.75	1	0.25	0.044495	0.010559	2.25	75	0.5	0.375
4	8	8	0.5	2	0.5	0.007813	0.000244	0.5	50	0.25	0.125
5	8	12	0.75	2	0.25	0.044495	0.010559	2.25	75	0.5	0.375
6	8	13	0.8125	2	0.1875	0.053944	0.019101	3.520833	81.25	0.666667	0.541667
7	8	14	0.875	2	0.125	0.056099	0.028774	6.125	87.5	1	0.875
8	8	15	0.9375	2	0.0625	0.042433	0.03073	14.0625	93.75	2	1.875
9	8	16	0.66666 7	3	0.333333	0.029264	0.003854	1.333333	66.66667	0.375	0.25
10	8	17	0.70833 3	3	0.291667	0.036839	0.006569	1.720238	70.83333	0.428571	0.303571
11	8	18	0.75	3	0.25	0.044495	0.010559	2.25	75	0.5	0.375
12	8	19	0.79166 7	3	0.208333	0.051288	0.015949	3.008333	79.16667	0.6	0.475
13	8	20	0.83333 3	3	0.166667	0.055816	0.022431	4.166667	83.33333	0.75	0.625
14	8	21	0.875	3	0.125	0.056099	0.028774	6.125	87.5	1	0.875
15	8	22	0.91666 7	3	0.083333	0.049441	0.032	10.08333	91.66667	1.5	1.375
16	8	23	0.95833 3	3	0.041667	0.032277	0.02609	22.04167	95.83333	3	2.875
17	8	24	0.75	4	0.25	0.044495	0.010559	2.25	75	0.5	0.375
18	8	25	0.78125	4	0.21875	0.049738	0.014476	2.790179	78.125	0.571429	0.446429
19	8	26	0.8125	4	0.1875	0.053944	0.019101	3.520833	81.25	0.666667	0.541667
20	8	27	0.84375	4	0.15625	0.056377	0.024109	4.55625	84.375	0.8	0.675
21	8	28	0.875	4	0.125	0.056099	0.028774	6.125	87.5	1	0.875
22	8	29	0.725	5	0.275	0.039936	0.007999	1.911364	72.5	0.454545	0.329545
23	8	30	0.75	5	0.25	0.044495	0.010559	2.25	75	0.5	0.375
24	8	31	0.775	5	0.225	0.048752	0.01363	2.669444	77.5	0.555556	0.430556



MVA—Example

- Objective
  - System Response time and throughput comparison before and after tuning.
  - Identification of system bottlenecks and scalability.
- Assumptions
  - Transactions characterize load and in turn load causes resource consumption and queue formation

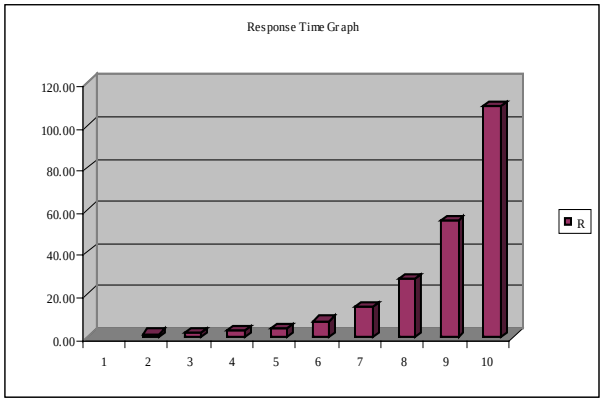


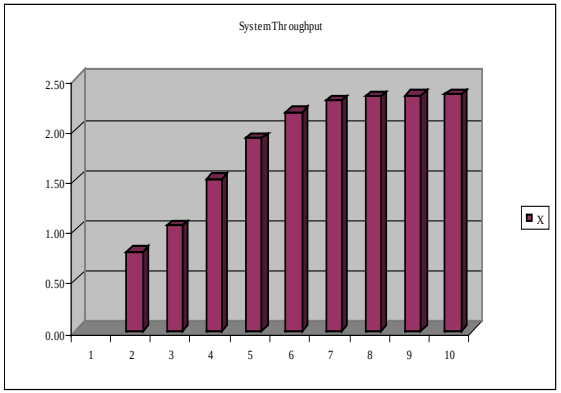
MVA—Example

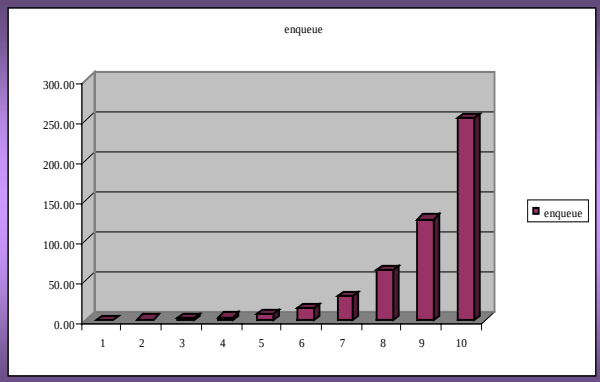
- Top wait events
  - before tuning
    - enqueue
    - db file sequential read
    - latch free
  - After tuning
    - db file sequential read
    - db file scattered read
    - latch free

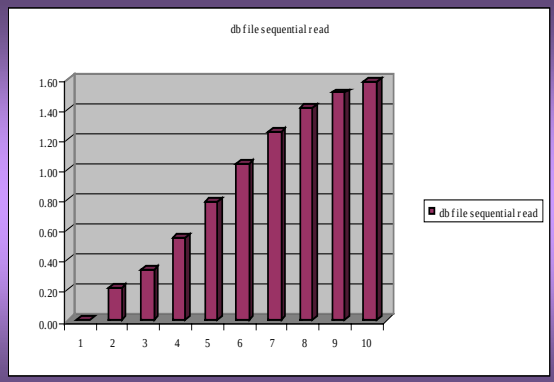
MVA—Example

Wait befor tuning	total waits	time in sec
enqueue	18,208	55,340
db file sequential read	4,106,472	17,412
latch free	17,111,757	9,520



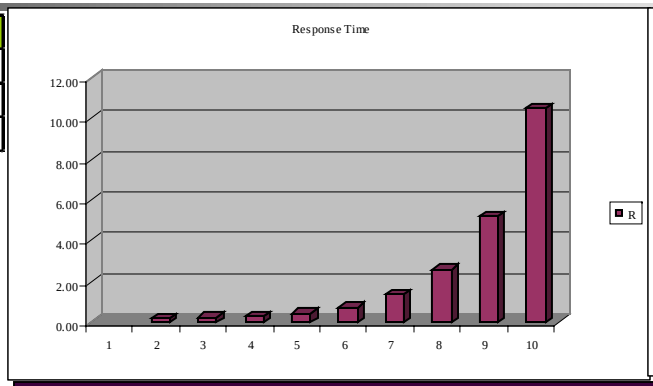


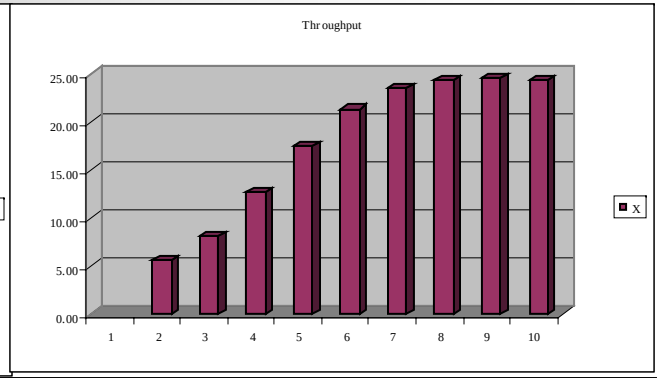


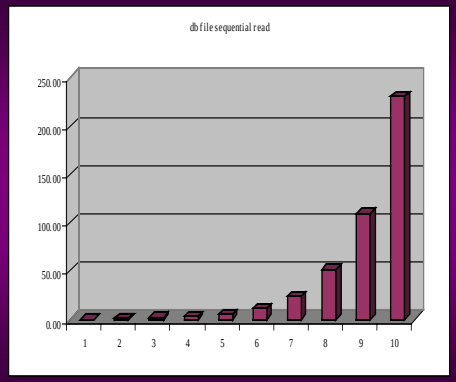


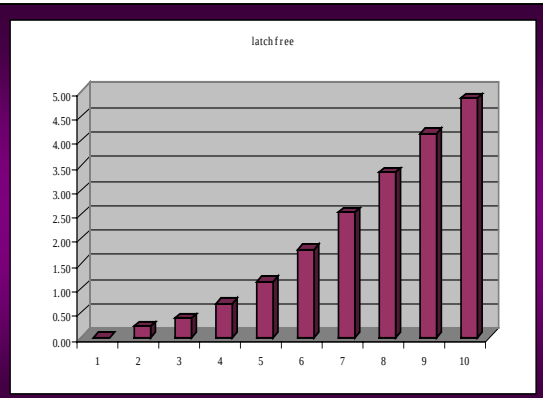
MVA—Example

Wait after tuning	total waits	time in sec
db file sequential read	6,465,304	5,548
db file scattered read	8,201,125	3,584
latch free	202,955	2,514



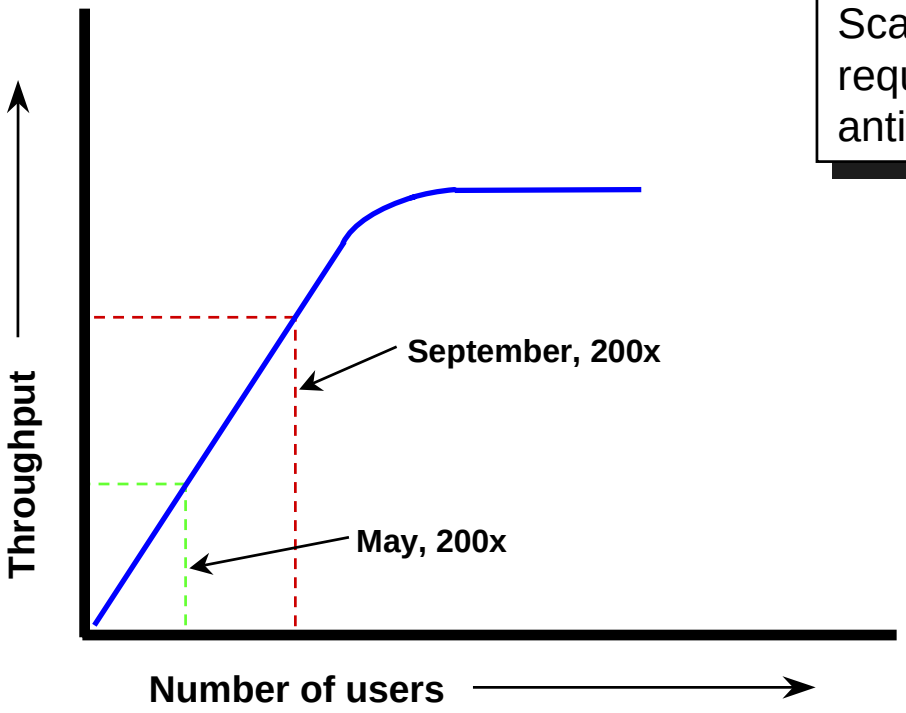






Data Server Load Testing- An Augmented Method

Goals: Scalability Projection

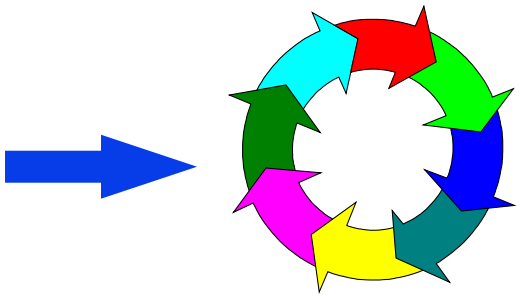


Scaling tests predict resource requirements to support anticipated loads.

Goals – Validate Success Initiatives

New Initiatives

- New Application Design & Development
- Application Upgrade Design & Planning
- Planned Enterprise Architecture Change/Upgrade
- Application/Server Patch Implementation



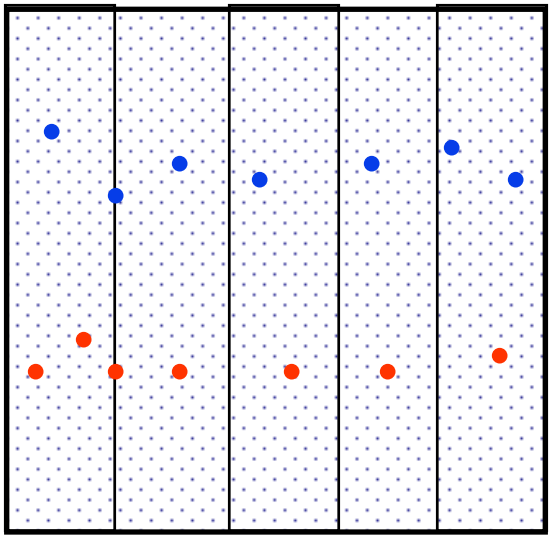
Performance-Based Analysis

Performance Analysis

- Current server config will support xxx users
- Application is CPU bound at yyy users
- Planned patch decreases I/O throughput by xx%
- Planned architecture will require xxx server platforms per 5,000 users

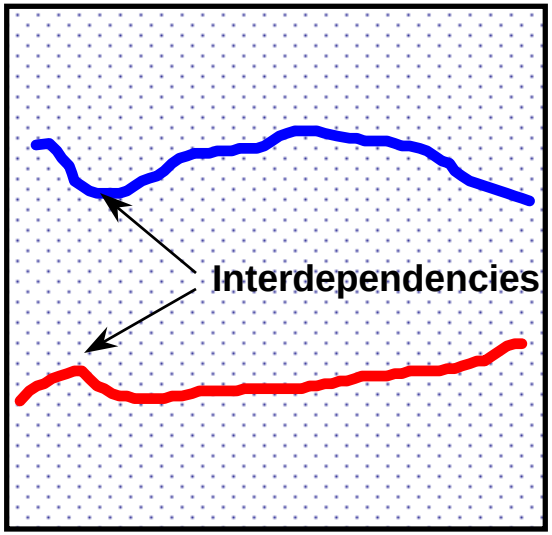
Summary

Step 1 - Gather Data Points



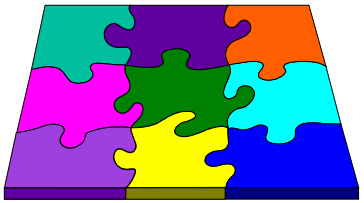
- Measure full suite of Oracle parameters
- Measure at key production windows
- Measure before/after transactions
- Snapshot of actual production workload

Step 2 - Find Patterns (Regression Analysis)



- Look for interdependencies
- Look for trends (regression)
- Look for key business transactions
- Determine “knee” points (break points)

Step 3 - Define and Deploy Production Load Profile



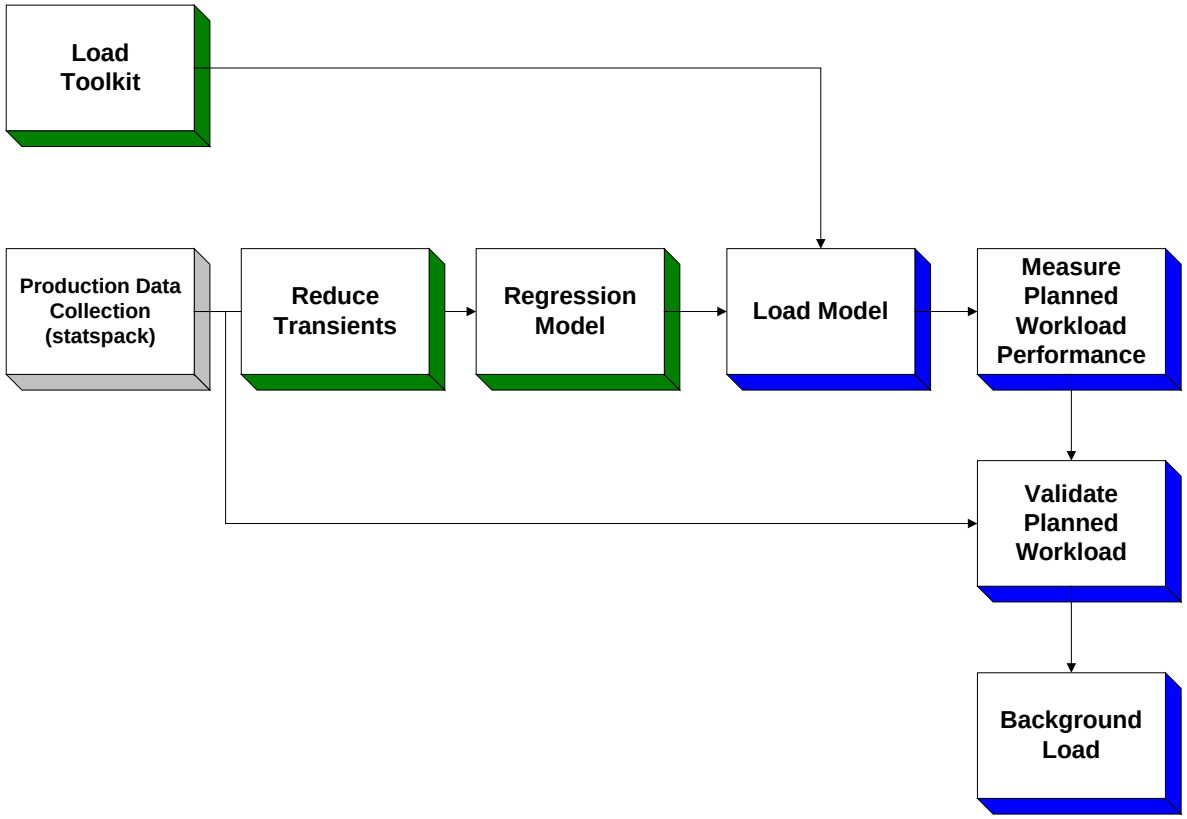
- **Key business transactions**
- **Key performance metrics**
- **Production load factors**
- **Business user profiles**
- **Optimized load inducers**

- What key transactions drive the business
- What resources limit performance
- What discrete, reproducible elements describe the production load
- Use specific load factors to do testing

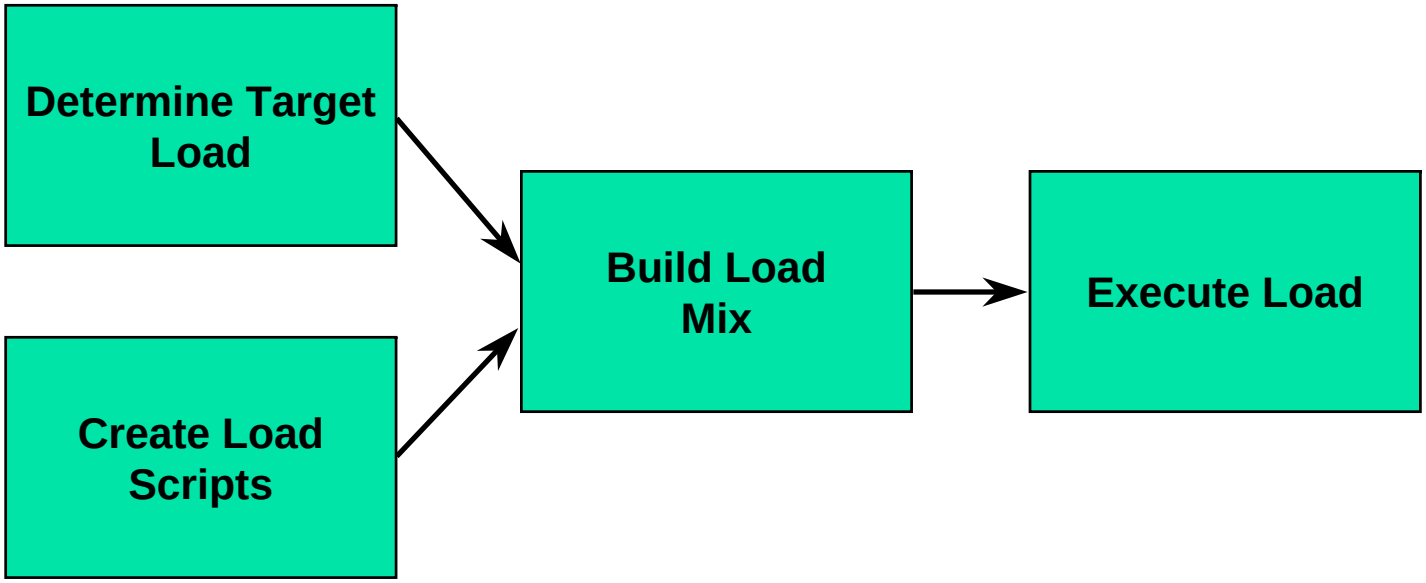
Three Phases :

1. **Assessment - gather data points, interview business users**
2. **Analyze and test data. Regression analysis. Optimize load factors**
3. **Deploy load profile to answer business questions - scaling, sizing, performance**

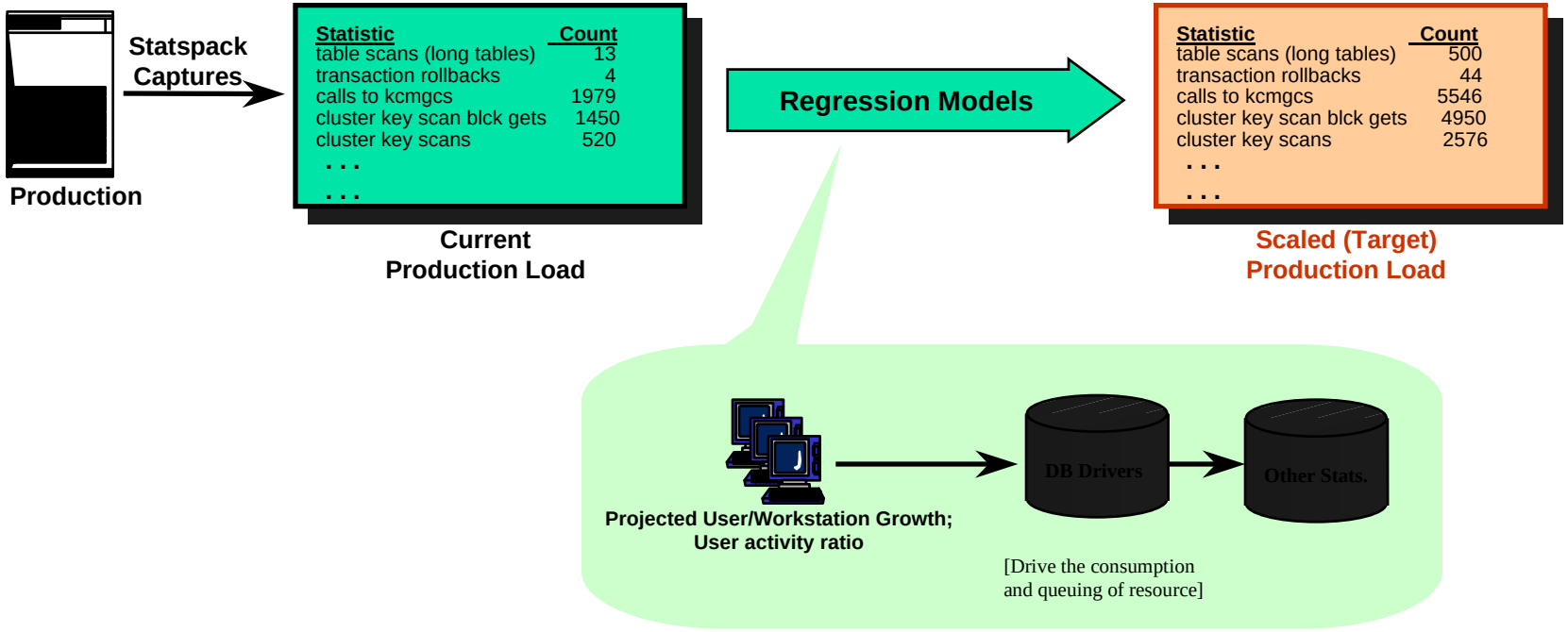
Process Flow



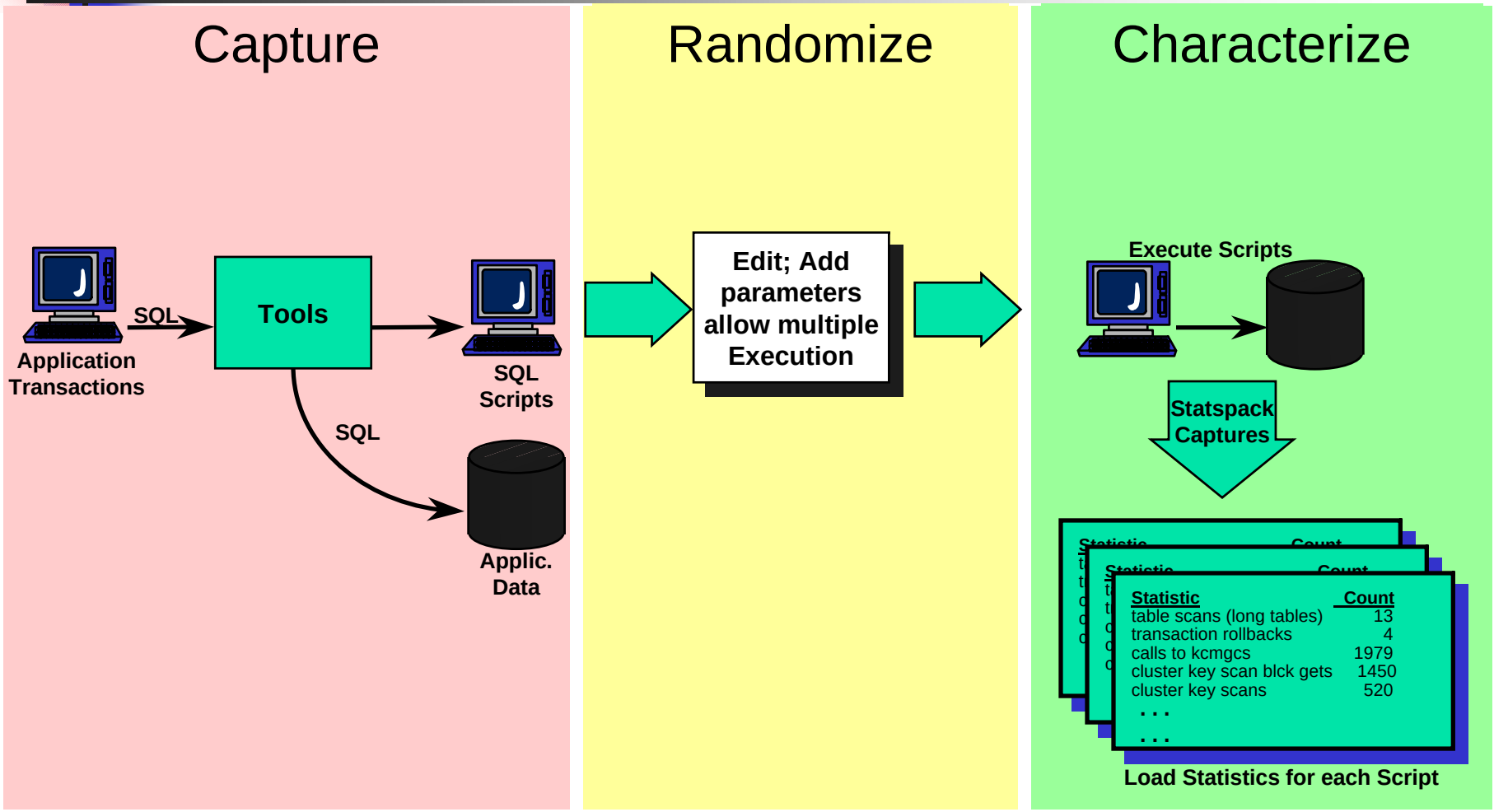
Data Server Load Method



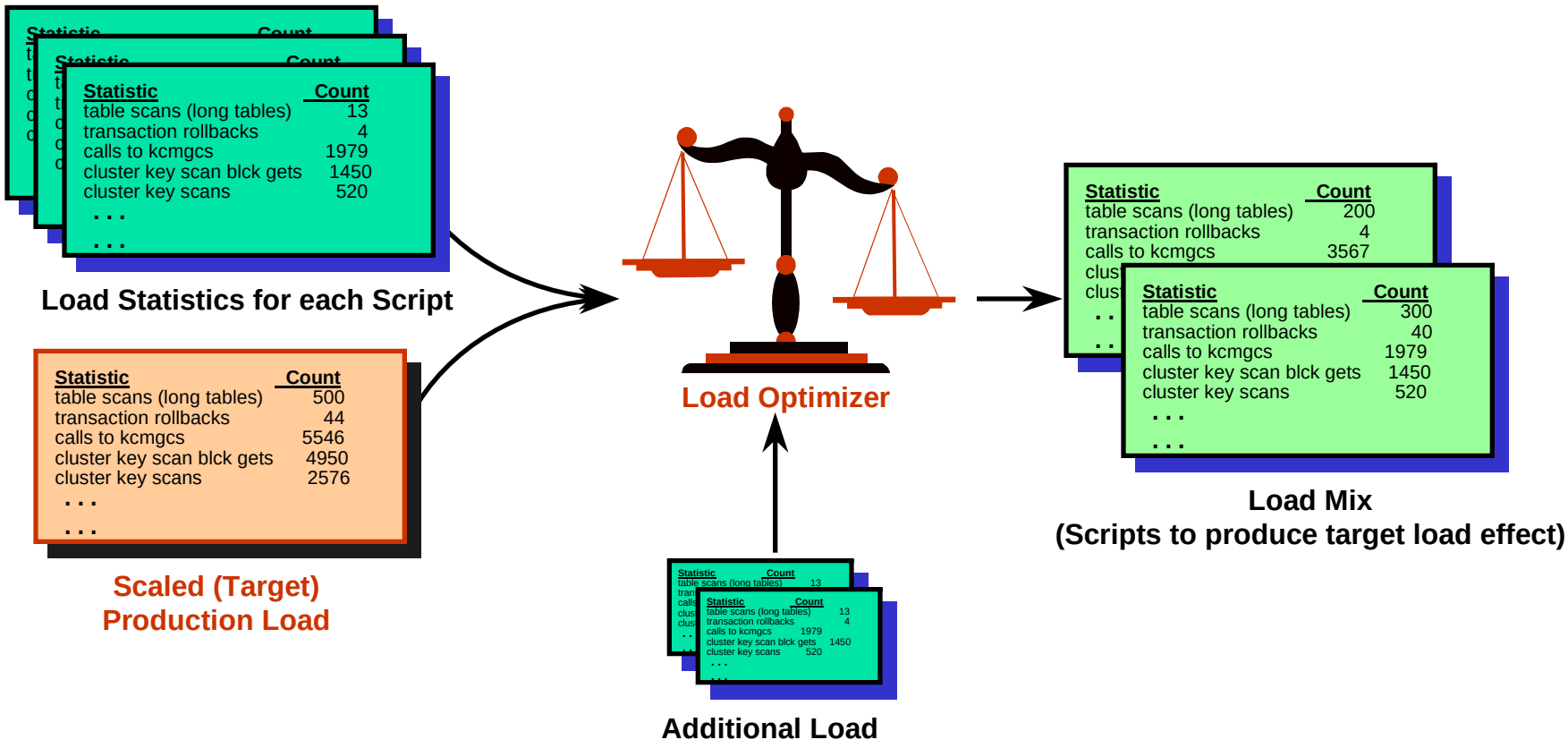
Determine Target Load



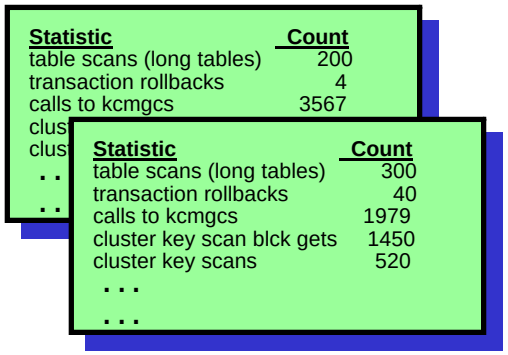
Create Load Scripts



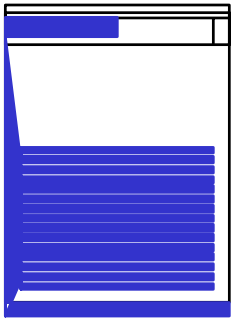
Build Load Mix



Execute Load & Analyze

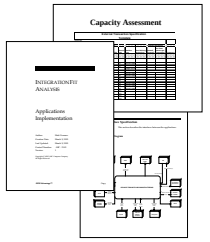


Load Mix



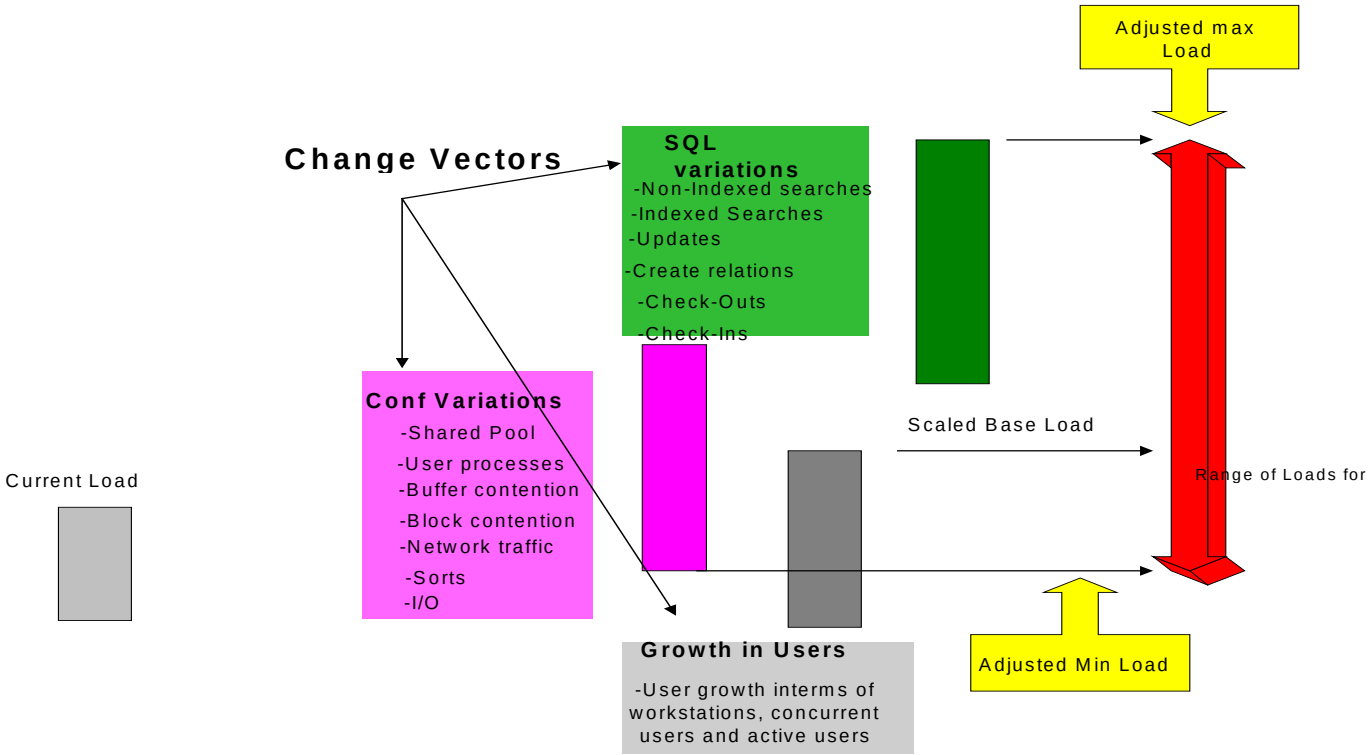
Execute Test

PEP
PM2
Pre Vue
Measurements

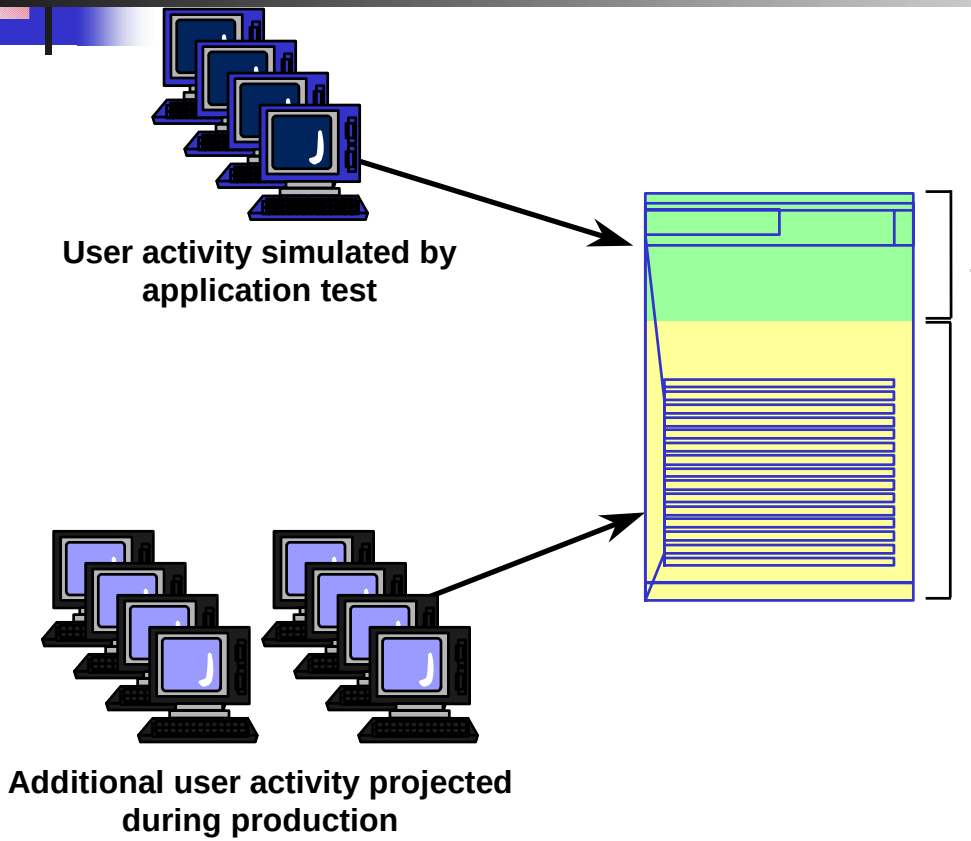


Analyze/Report Results

ODSS Application Configuration and SQL variants



Define Background Load



Data Server load resulting from test scripts

Background Load

(= Full production load - load from application test)

Background load is required for accurate test results.

Example of Performance Analysis

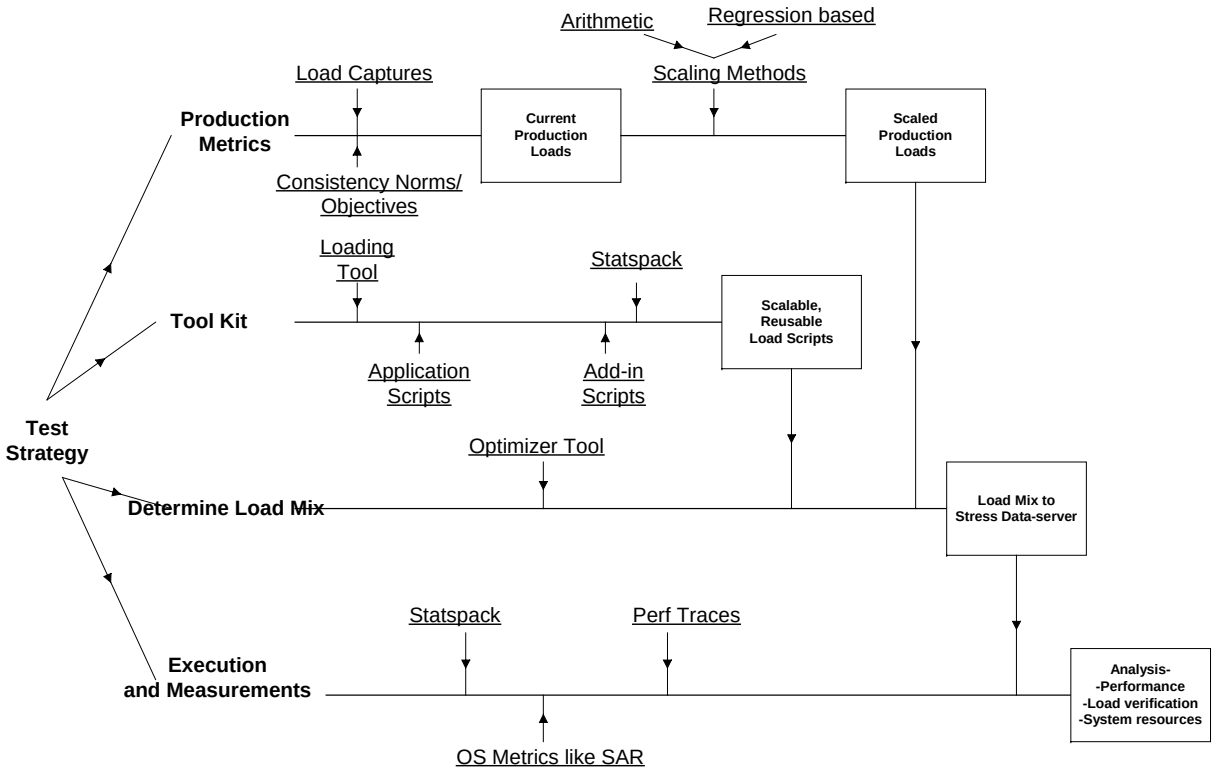
Summary of performance analysis

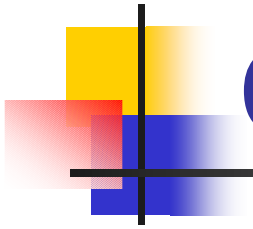
	100-200%loa	200-300%loa	300-400%loa	100-300%loa	100-400%load
Long Running Queries	159%	194%	214%	308%	661%
Medium range Queries	125%	163%	113%	200%	229%
Short queries	113%	155%	115%	175%	202%

Performance Impact Analysis:

1. Long running queries show severe degradation 1t 300 and 400% loads
2. Medium range and short queries deteriorate severely during 300% load growth and remain relatively

Strategic Elements





Conclusion and Contact

- Conclusion
  - An new approach to Capacity, Modeling and Scalability Testing
- Contact Information
  - Jb.Sastry@ge.com
  - shankar.jayaganapathy@oracle.com