



**SUNGARD**

# Performance Test Management

Praful Pillay

Vice President, SunGard Offshore Services

- SunGard Inc – a global leader in software products and services
- Targeted at financial services, public sector & higher education
- Revenues of \$4 billion
- Serves more 25,000 customers across 50 countries
  
- SunGard Offshore Services – based out of Bangalore and Pune.
- Extensive experience in Product Engineering
- Center of Excellence for Testing Services

- Significant investments
- Specialized skills – testing experts, tuning specialists...
- Making the process repeatable, re-run able
- Integrating it within the development lifecycle
- Making it part of application testing process
  - Ability to schedule performance testing
  - Involve developers while analyzing test results
  - Integrating with defect tracking system – report, track & measure
- **Meeting expectations of different stakeholders**

# Identify Key Performance Indicators

- Different strokes for different folks – question of semantics

## End Users



Interface rendering time



Number of transactions  
per unit of time

## Developers



Requests per second /  
CPU utilization

- Setting and managing expectations is critical

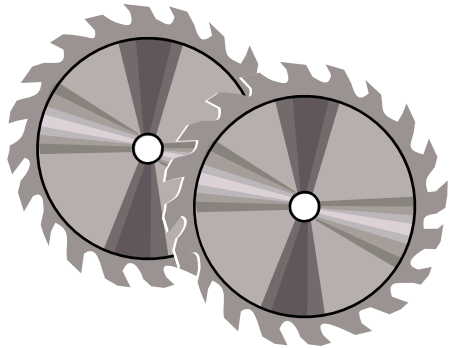
# Develop Usage Scenarios with End Users

- Start with the end user
- Develop Spreadsheets to capture workload characteristics (scenarios) <SAMPLE>
- Setting up data load
- Develop scripts for performance

# Have an Execution Engine

- Performance testing – for creating benchmarks or to measure against benchmarks
  - Benchmarking – observational in nature
  - Testing performance is about tuning
  
- Effective execution of scripts involves
  - Executing all scenarios for measurement
  - Tuning scenarios individually
  - Re-running all scenarios
  
- Requirements for an execution engine
  - Tests can be run on DEMAND !!
  - Reduce set-up time and therefore costs
  - Focus on Performance testing and not on infrastructure requirements

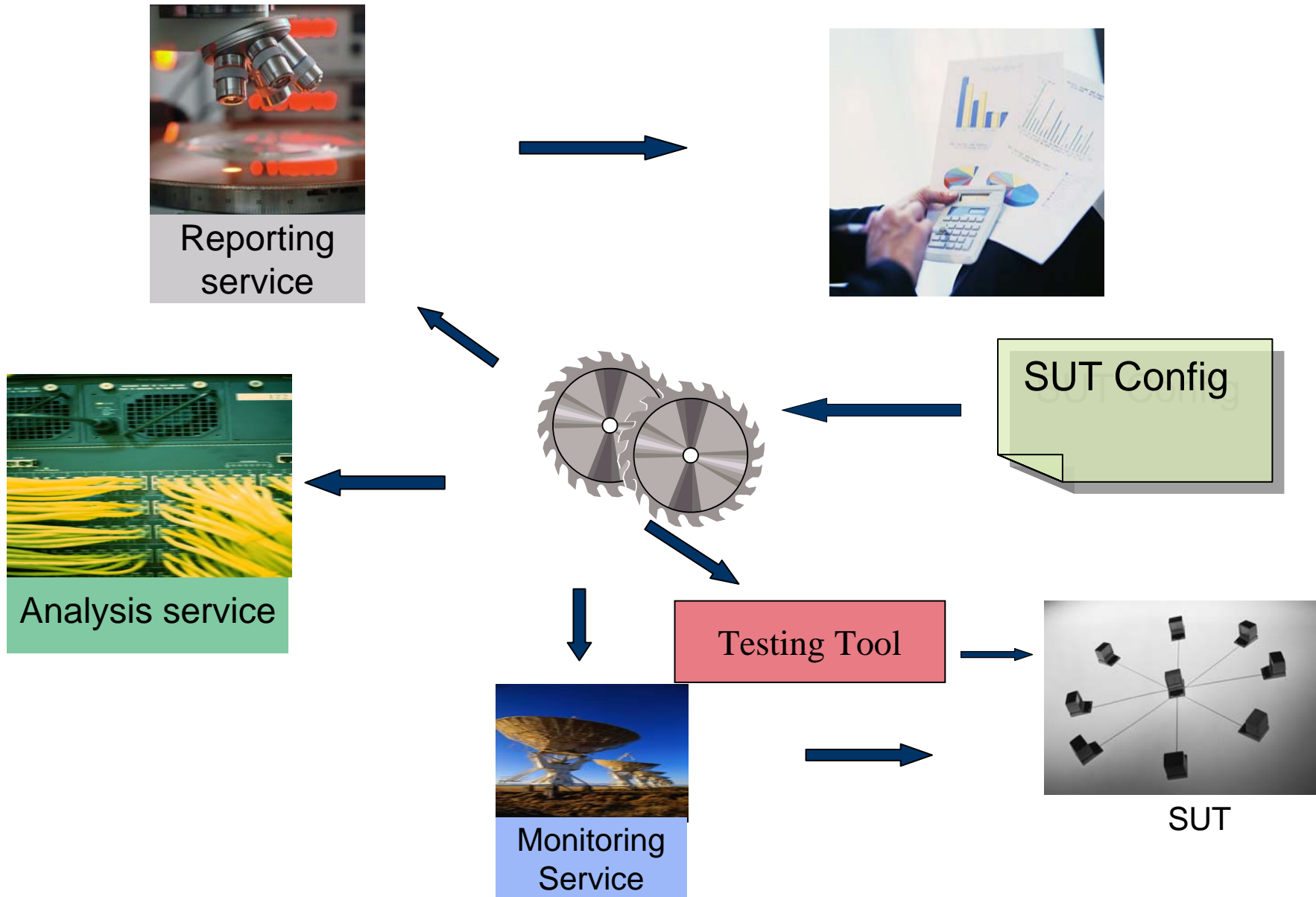
- An Execution Engine



### SUT Config

- S/w & H/w Settings
- Steady state of SUT
- Backup & restore details
- Data inputs
- Sequence of Test execution
- Increment factor for load

# So what that means....



# What more should the Execution Engine do

- Stop execution on reaching performance threshold
- Record performance characteristics at saturation level  
(Any tuning effort must delay reaching this level on load increase)
- Run Clean up activities
  - Backup and restore databases
  - Execute restore scripts of any
  - Bring the test system to the pre-run state
  - Prepare data inputs

# Make Analysis & Reporting Easy

## Data Collection

- Instrumentation & monitoring tools to probe SUT
  - Layer-wise instrumentation
  - Component-wise instrumentation
  - Application specific instrumentation
  - Infrastructure specific instrumentation
- Log Files – server level or database level

## Analysis

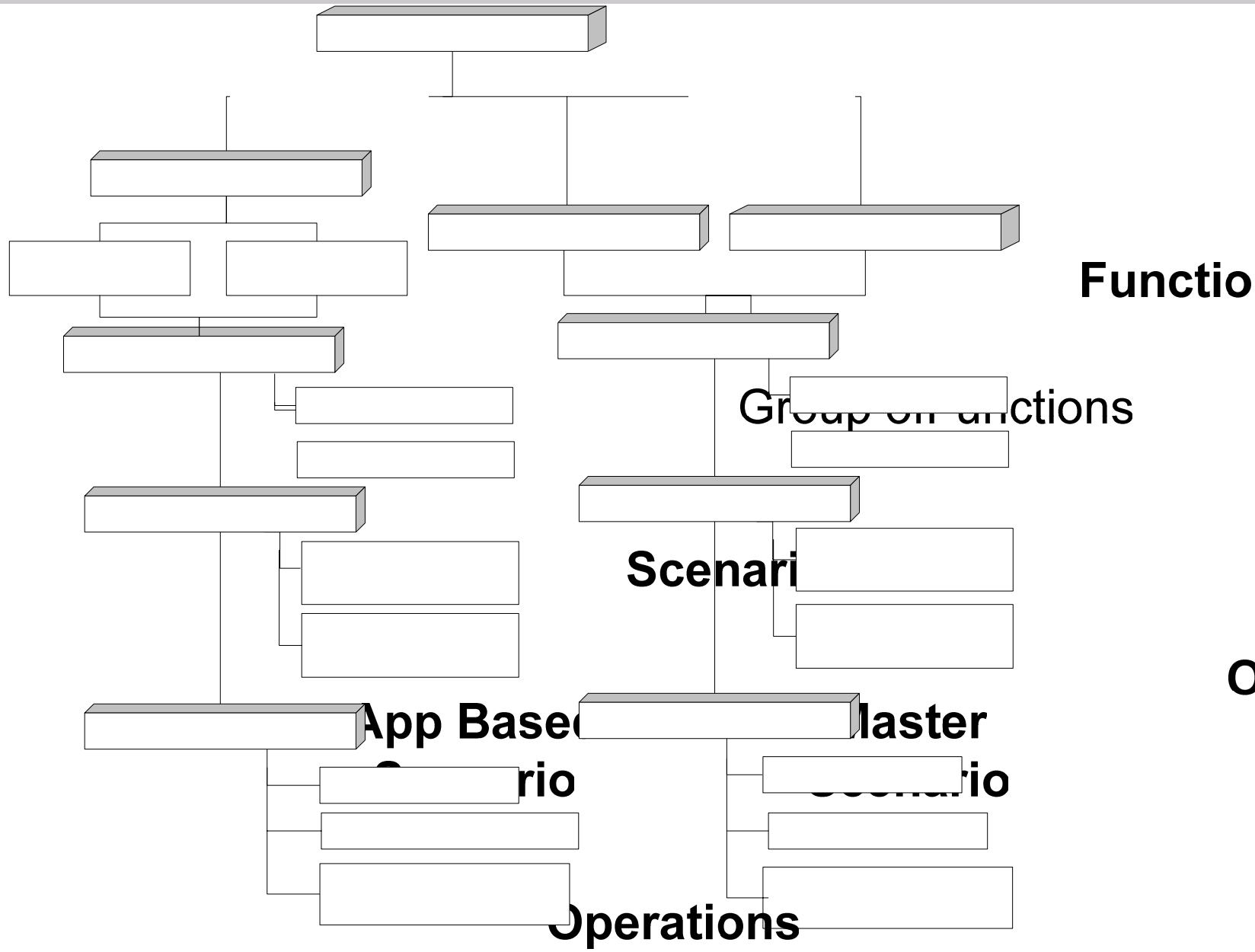
- Takes takes huge effort and time
  - Different formats
  - Correlation across log files & instrumented outputs
- Specialist skills are at a premium
- Spending time sifting through large log files is inefficient

So what we need are Analyzer tools <Sample>

## Data Store

- Store the results of a performance test
- Maintain results across cycles of performance runs
- Be able to query these results
- Store recommendations made by Tuning Specialists
- Track implementation of recommendations
- Do all this across version releases of applications

An Application to allow Developers and Performance Test Engineers to access the data store



- **Function Summary**
  - Scenarios Tested
  - Test Result Summary
  - Factors impacting performance
  - Data configuration

**What End Users Consume**

- **Hardware configuration**
- **Function Under Test**
  - *Phase 0 (Baseline)*
    - Test Results
    - Interpretations
    - Recommendations
  - *Phase 1*
    - Test Results
    - Comparison with Phase 0
    - Interpretations
    - Recommendations

**What Developers Consume**

# Where Performance Testing should take you

- In Development
  - Build knowledge base of recommendations
  - Develop patterns around these recommendations
  - Publish revised development standards
  - Integrate performance testing with the development lifecycle
  
- Planning Growth
  - Traffic Management
  - Capacity Planning [<Sample>](#)
  - Adopt testing experiences during deployment/configuring production systems

Questions ?



**SUNGARD**

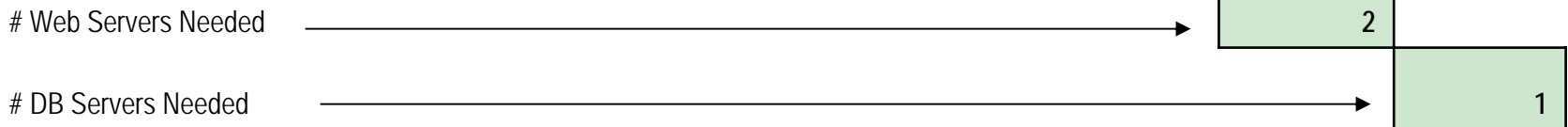
# Performance Test Management

Praful Pillay

Vice President, SunGard Offshore Services

H/w Specs	WEB		DB	
CPU Cycles	2500	Mhz	2500	Mhz
# Processors	2		4	

	Performance Measures at Saturation					Profile function requirements		
Application Profile	R/S	CPU WEB	CPU DB	Cost WEB	Cost DB	Desired R/S	WEB CPU Budget	DB CPU Budget
Application 1 (Function 1)	12	80%	30%	333	250	4	1333	1000
Application 2 (Function 2)	10	90%	40%	450	400	3	1350	1200
Application 3 (Function 3)	14	75%	45%	268	321	5	1339	1607
Application 4 (Function 4)	8	85%	50%	531	625	3	1594	1875
Application 5 (Function 5)	21	65%	30%	155	143	5	774	714
Application 6 (Function 6)	10	90%	25%	450	250	4	1800	1000
<b>Total Requirement</b>							<b>8190</b>	<b>7396</b>



Online Payment		Notes
Sample Operation 1: Online Payment Module		
Number of concurrent users accessing the system	20	Enter the expected number of concurrent users accessing the function.  A value of 20 is given as an example
Expected response time	6	Enter the expected response time. This is the time by which page load requests, made by the above number of users is completely processed.  A value of 6 secs is given as an example.
Total number of load operations in one hour	12000	This column calculates the through-put from the system in one hour .  Example: A value of 12000 signifies that 12000 page-load-operation requests are processed by the system in an hour. The assumption made here is that, the requests are processed in batches of 20 users. Thus, 10 batches of 20 users each will be processed in a minute (since 20 users have a 6 sec response time as per the example) and 600 batches of 20 users each will be processed in an hour. <b>Formula Used :</b> <b><math>((\text{Total number of secs in an hour}) / (\text{Expected Response time})) * \text{Number of concurrent users accessing the system.}</math></b>

Working hours / day ( in hours )	2	<p>Enter the number of working hours the Web site will be available for the operation.</p> <p>Ideally, the site would be available for 24 hours, but for the purpose of simulating a peak load (assuming that all the users submit payment requests during the last few days before the due date), a value of 2 hours is taken as example.</p>
Total number of operations in one day	24000	<p>Total number of operations in one day denotes the amount of page load operations completed in a day.</p> <p><b>Formula Used :</b>  <b>( Working hours / day ) * ( Total number of operations in one hour )</b></p>
Total student population doing online payment	100000	<p>Enter the total number users who wish to make online payment.</p> <p>A value of 100000 is given as an example.</p>
Number of working days required for all the users to load the payment screen ( in days )	4.17	<p>Typically the response time expected out of this operation is largely based on user perception (For example, I may need a response time of 5 secs, every time I click on the Summary screen. ); hence, this productivity measure is directly proportional to the load experienced by the server system with a given number of concurrent users.</p> <p><b>Mathematically, Productivity = Number of concurrent users * Response Time.</b></p> <p>Since Response Time is constant and is based on user perception, Productivity is directly a measure of number of concurrent users.</p>

# SQL Trace Analyzer : In action

Analysis Report : MSDEVDB\JHU\_NIGHT\_Part3\_Aug03
151 Rows

BatchStatement	Occurrences	TotalDurati...	AverageD...	DurationSt...	TotalReads	AverageR...	ReadsStd...	TotalCPU	AverageC...	CPUStdDev
SELECT SNAPSHOT_IN_DEFAULTFOLDER,ALT_SNAP...	3	16	5	9.24	165	55	0.00	16	5	9.24
CMN_STATUSTRANSITIONSLISTEX	2	172	86	14.14	2814	1407	0.00	172	86	9.90
OFF SET PARSEONLY OFF				2.60	0	0	0.00	0	0	0.00
SELECT SWS_USERSID FROM SWS_USERS INNER JOI...				11.31	226	113	60.81	16	8	11.31
SELECT PLAN_NAME, REMOTE_HISTORY_SERVER FR...				0.00	30	30	0.00	0	0	0.00
CMN_COHORTSLISTEX	144	4367	30	18.97	102393	711	263.47	3969	27	14.94
SAS_STUDENTPOSTREQUIREMENTS	112	3846	34	32.04	112261	1002	1045.84	3717	33	31.89
SELECT * FROM SAS_STUDENTINSTANCES WHERE S...	99	312	3	6.14	3703	37	4.27	330	3	6.46
SELECT DISTINCT CMN_RULESRULEGROUPSID FROM...	123	2			36714	298	148.28	2295	18	10.94
CMN_STATUSESLISTEX	5				4785	957	197.18	187	37	8.32
DBCC SQLPERF(LOGSPACE)	86	40	0	2.78	172	2	0.00	31	0	2.35

**Normalized Command Text**

```
SET NOCOUNT ON SET TRANSACTION ISOLATION LEVEL READ COMMITTED EXEC DBO.SAS_STUDENTPOSTREQUIREMENTS @EVALUATIONTYPE = {STR},
@CMN_APPLICATIONINSTANCESID = {STRN},@APPLICATIONSID = {##},@USERGUID = {BS},@ENTITYIDLIST = {STR}
```

Advanced Sort

**Occurrences**

**Pattern Graph**

Occurrences	Duration	Reads	CPU
112 Rows	30	952	31
	46	998	47
	33	772	31
	30	374	
	33	374	

**Original Command Text**

```
Set NoCount On SET TRANSACTION ISOLATION LEVEL READ COMMITTED
EXEC dbo.SAS_StudentPostRequirements @EvaluationType = 'Download',
@CMN_ApplicationInstancesID = 'N':0;84943,85758,60564',
@ApplicationsID = 3,@UserGUID =
0x56e4a2428a60134db882425fd7d0676b,@EntityIDList =
'84943,85758,60564'
```

**Original Batch Command as executed**

Statements for selected Occurrence

Statements	Duration	Reads	CPU
16	16	67	16
0	0	7	0
0	0	42	0
0	0	7	0

**Sample Statement**

```
SELECT
@ReqCompleted = ReqCompleted
FROM
SAS_StudentInstances
WHERE
CMN_ApplicationInstances
@CMN_ApplicationInstancesIDTemp
```

**Normalized Statement Text**

```
SELECT @REQCOMPLETED = REQCOMPLETED
FROM SAS_STUDENTINSTANCES WHERE
CMN_APPLICATIONINSTANCESID =
@CMN_APPLICATIONINSTANCESIDTEMP
```

**Normalized Statement**