

Crack The Complexity Of Oracle Applications R12 Workload




Ajith Narayanan
Oracle ACE Associate 
8th- Nov -2014, Bangalore, India.

Who Am I



Ajith Narayanan

- ❑ 10 years of Oracle [APPS] DBA/Technical Architect experience.
- ❑ Oracle ACE Associate 
- ❑ Blogger :- <http://oracledbascriptsfromajith.blogspot.com>
- ❑ Member:-OAUG, AIOUG
- ❑ Website Chair (2011-2013) - Oracle RACSIG
<http://www.oracleracsig.org>
- ❑ AIOUG Real Application Clusters SIG Leader

Agenda

1. Introduction
2. Challenge
3. Workload Characterization
4. Instrumentation
5. Collection
6. Classifying Workload
7. Measure
8. Interpret
9. Conclusion
10. Q&A

Introduction

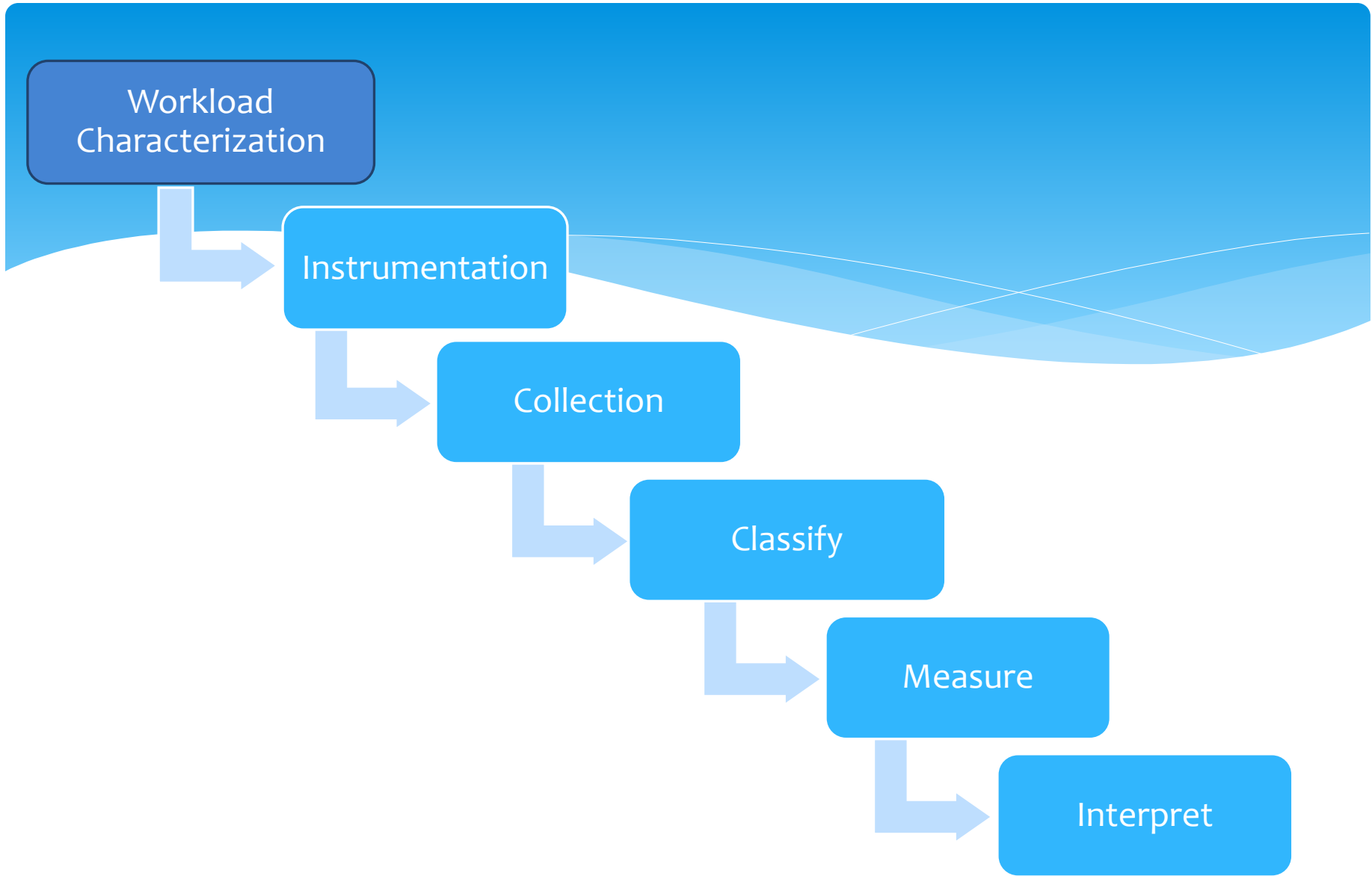
- * Oracle Applications R12 is one of the flagship products from Oracle .
- * Oracle Apps R12 implementations are quite large with more and more customizations done over a time.
- * Being a huge product with 200+ products packaged most of the architects and DBA's find it difficult to understand the workload trend of the system.
- * Customization's further contribute to the complexity.

Take Away

- *End of this session, we should be able get the functional perspective of our EBS system workload and its impact in 360°*

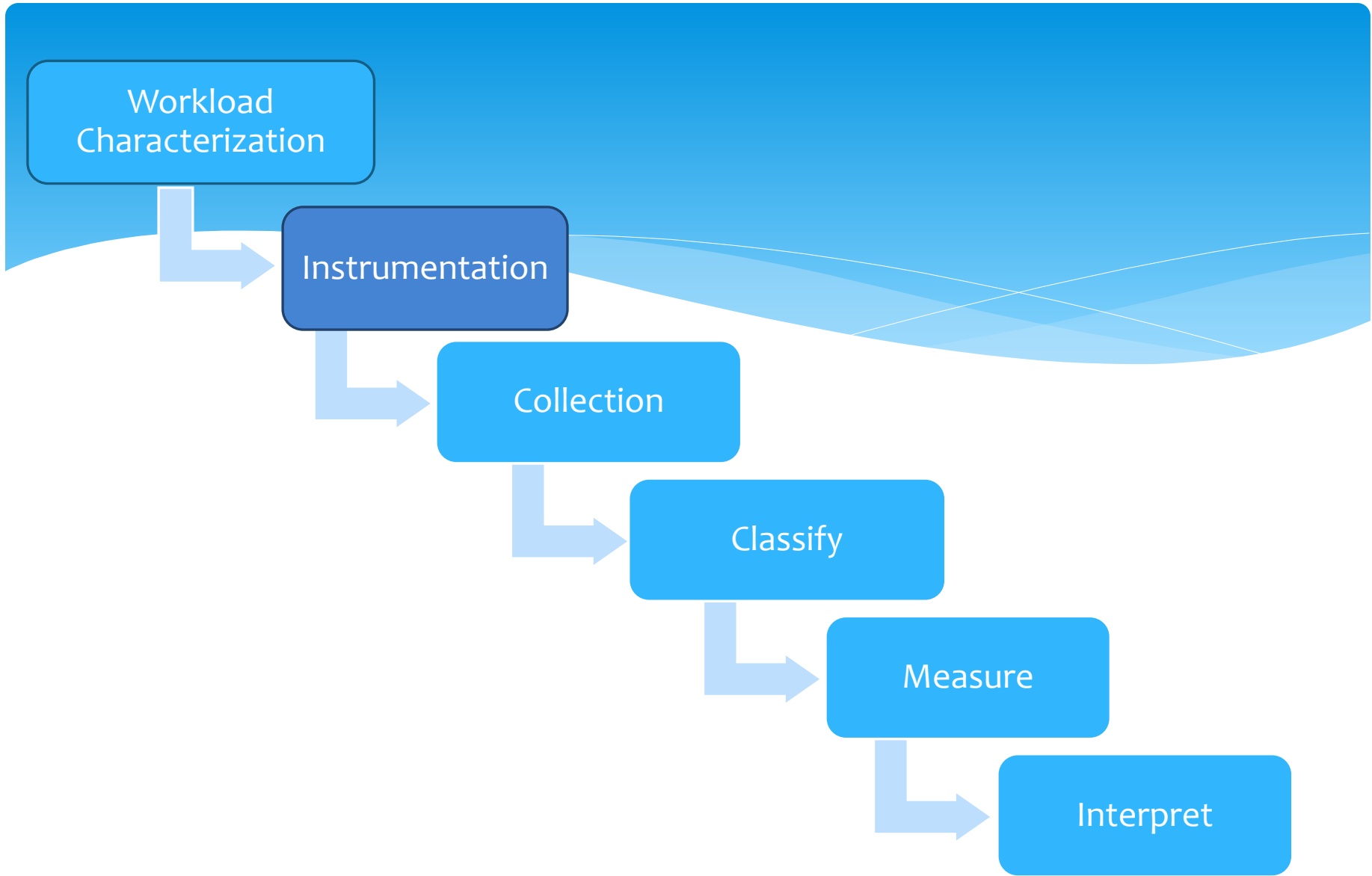
Challenge

- * Oracle Application R12 has a very rich techstack.
- * Understanding each components, analyzing the sizing requirements for each of these components is really tricky and tedious job.
 - *Forms, Reports, Self Service Applications, Concurrent Programs*
- * We will be trying to understand Workload characterization process by identifying classes of workload, measuring those classes and then identifying their impact to the business.



Workload Characterization

- ▶ We should be starting our analysis from the database (Matrices related to our workload)
- * Understand the business process that is responsible for the peak workload.
- * Know the workload in functional perspective to have a clear picture (Batch, Forms, Self Service, Java, FMW, OA Framework Etc.)
- * For the critical business process and related workloads, we should be ensuring that enough resources are available, since they are not available for compromises.



Instrumentation

Forms Workload -Must be explained on the degree at which it impacts business.

- apps.fnd_login_resp_forms*
- apps.fnd_logins*
- apps.fnd_form_tl*
- apps.fnd_login_resp_forms*
- apps.fnd_application_tl*

Form Work Load (Forms Workload)

DATE	CNT
20-AUG-14	86545
24-AUG-14	7341
04-SEP-14	86527
06-AUG-14	78177
07-AUG-14	76764
11-AUG-14	82277
18-AUG-14	83819
30-AUG-14	10098
12-AUG-14	85184
16-AUG-14	7903
17-AUG-14	2336
23-AUG-14	11478
01-SEP-14	55959
06-SEP-14	12958
21-AUG-14	89256
25-AUG-14	66664
03-SEP-14	88154
08-AUG-14	72425
09-AUG-14	10116
14-AUG-14	82828
15-AUG-14	59315
26-AUG-14	71879

Top Application Modules (Forms - By top App)

APPL_NAME	MAX	AVG
Inventory	10159	6365
Purchasing	8184	5054
Payables	6534	3056
Order Management	5380	3053
Application Object Library	5345	3149
Receivables	4196	2808
Shipping Execution	3545	1978
Work in Process	3326	1781
Master Scheduling/MRP	1869	1076
Receibmento Integrado	1701	661
General Ledger	1092	373
Bills of Material	927	666
Quality	360	151
Subledger Accounting	341	196
Depot Repair	340	215
Warehouse Management	266	173
Quoting	196	153
Advanced Pricing	175	133
Assets	163	163
Engineering	157	124

Instrumentation

* Batch workload

Apps.fnd_concurrent_requests

```
=====
Concurrent High Frequency (Conc High Freq Monthly)
Concurrent High Impact (Conc High Impact Monthly)
Date on which Concurrent MAX TIME Taken
Concurrent MAX AVG MONTHLY (Data only for analysis)
=====
```

* Self-Service Applications

apps.icx_sessions

* DBMS_APPLICATION_INFO package

set_client_info

set_module

set_action procedures

Instrumentation

- Identify foreground sessions with "SYS\$USERS" & background sessions with "SYS\$BACKGROUND".

USERNAME	SID	SERIAL#	CLIENT_IDENTIFIER	MACHINE	CLIENT_INFO	MODULE	ACTION	SERVICE_NAME
	1	5		ebstdbnode1				SYS\$BACKGROUND
APPS	3	115		ebscmmnode1	2	0 STANDARD_SHORT_REQS@ebscmmnode1 (TNS V		SYS\$USERS
IDENT_USER	4	19309		Development\Win781		PL/SQL Developer	SQL Window-select *	DELTA1
APPS	5	17137		Development\Win901		PL/SQL Developer	SQL Window-New	DELTA1
APPS	6	43		ebscmmnode2.lab.com	2	0 JDBC Thin Client		SYS\$USERSAPPS
	7	77		ebscmmnode1	270	0 REPORTS@ebscmmnode1 (TNS V1-V3)		SYS\$USERS
IDENT_USER	8	89		Development\Win321		PL/SQL Developer	SQL Window-SELECT SEG	DELTA1
APPS	10	15379		ebscmmnode1	2	0 STANDARD_SHORT_REQS@ebscmmnode1 (TNS V		SYS\$USERS
APPS	11	7		ebscmmnode2	2	0 CRITICAL_SHORT_REQS@ebscmmnode2 (TNS V		SYS\$USERS

- * Additional attributes to extended SQL trace data as well.

```

*** ACTION NAME:() 2013-08-26 09:18:18.647
*** MODULE NAME:(SQL*Plus) 2013-08-26 09:18:18.647
*** SERVICE NAME:(SYS$USERS) 2013-08-26 09:18:18.647
*** SESSION ID:(202.11594) 2013-08-26 09:18:18.647
APPNAME mod='SQL*Plus' mh=3669949024 act='' ah=4029777240
    
```

Instrumentation

- **Client Identification:- Use, DBMS_SESSION.SET_IDENTIFIER**

```
SQL> exec dbms_session.set_identifier('SANGAM14');
```

```
PL/SQL procedure successfully completed.
```

USERNAME	SID	SERIAL#	CLIENT_IDENTIFIER	MACHINE	CLIENT_INFO	MODULE	ACTION	SERVICE_NAME
APPS	1995	19719	SANGAM14	ebsscmnode1		SQL*Plus		SYS\$USERS

* Application Information

```
SQL> exec dbms_application_info.set_client_info('AJITHNARAYANAN');
```

```
PL/SQL procedure successfully completed.
```

USERNAME	SID	SERIAL#	CLIENT_IDENTIFIER	MACHINE	CLIENT_INFO	MODULE	ACTION	SERVICE_NAME
APPS	1995	19719	SANGAM14	ebsscmnode1	AJITHNARAYANAN	SQL*Plus		SYS\$USERS

Instrumentation

- **Module Information (With action performed)**

```
SQL> exec dbms_application_info.set_module('AjithModule','step1');
```

```
PL/SQL procedure successfully completed.
```

USERNAME	SID	SERIAL#	CLIENT_IDENTIFIER	MACHINE	CLIENT_INFO	MODULE	ACTION	SERVICE_NAME
APPS	1995	19719	SANGAM14	ebsscmnode1	AJITHNARAYANAN	AjithModule	step1	SYS\$USERS

- **ASH - Active Session History**

- ❑ *Once a session disconnects the details are not available through the V\$SESSION based views,*
- ❑ *ASH overcomes this problem with ability to catch sessions that might have otherwise been missed using the V\$SESSION view.*
- ❑ *The data is sampled at 1 second interval, so the data is more granular than most attempts at sampling the V\$SESSION based views.*

Instrumentation

- **Tools Advantage (SQL*Plus commands)**

```
SQL> set pages 9999
SQL> show appinfo
appinfo is ON and set to "SQL*Plus"
SQL> set appinfo on;
SQL> show appinfo
appinfo is ON and set to "SQL*Plus"
SQL> set appinfo "AjithNarayanan"
SQL> variable mod varchar2(30)
SQL> variable act varchar2(30)
SQL> execute dbms_application_info.read_module(:mod,:act);
```

PL/SQL procedure successfully completed.

```
SQL> print mod
```

MOD

```
-----
AjithNarayanan
```

```
SQL> print act
```

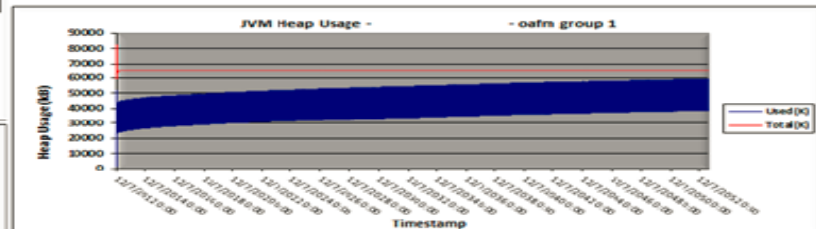
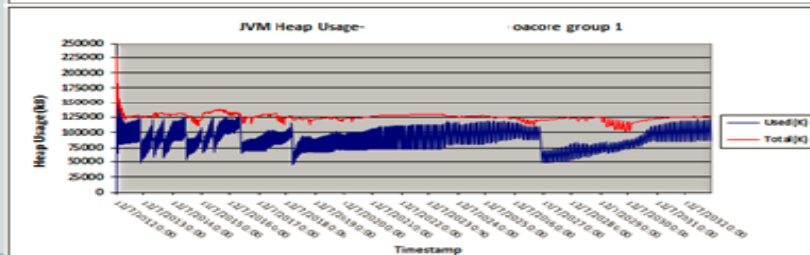
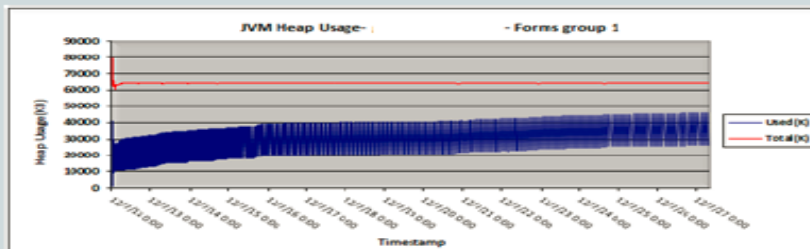
ACT

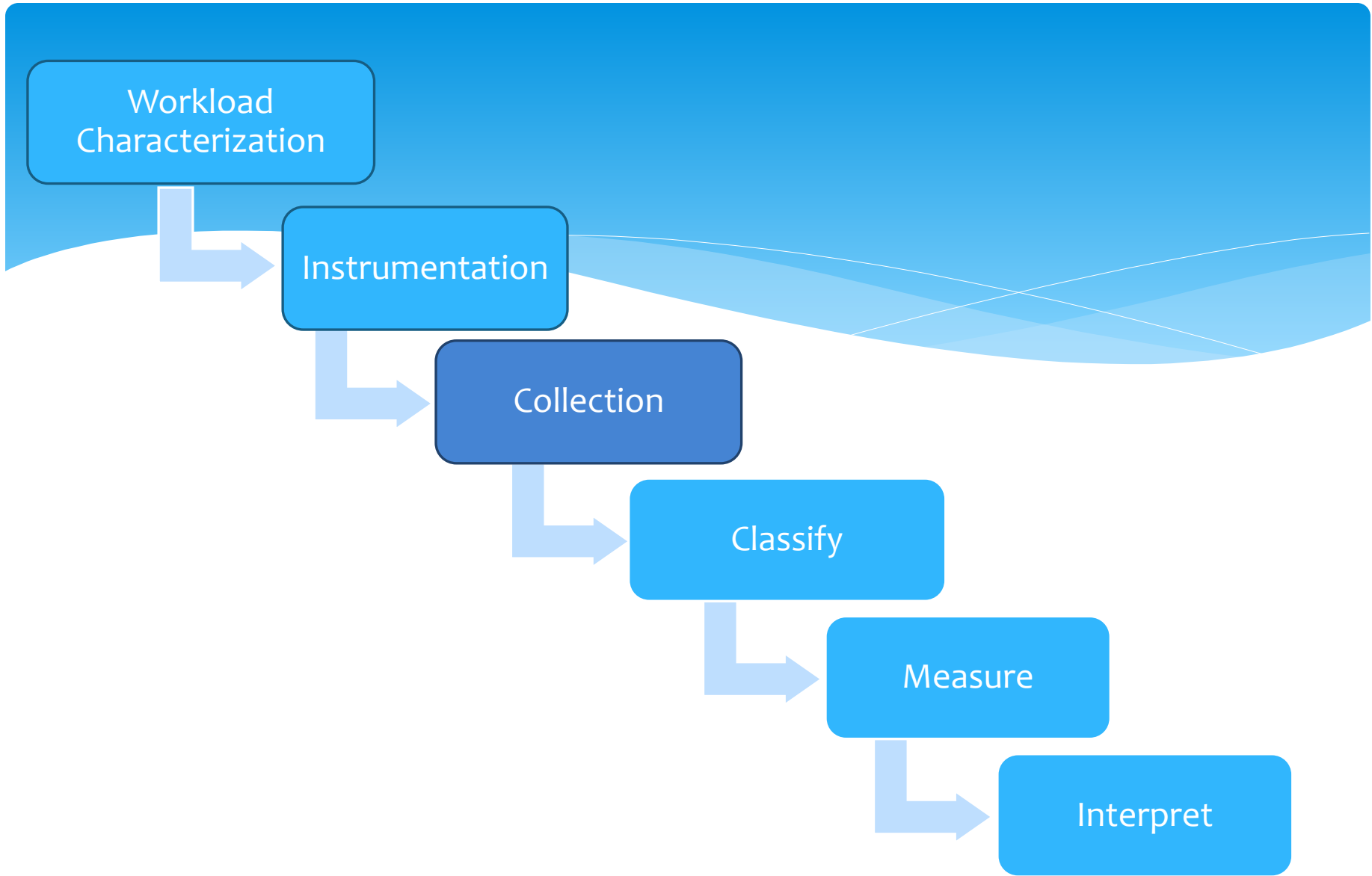
```
-----
SQL> show appinfo
appinfo is ON and set to "AjithNarayanan"
SQL> set appinfo off
SQL>
```

Instrumentation

- OACore JVM usage can provide a 360 degree view of the Application workload using GCViewer.

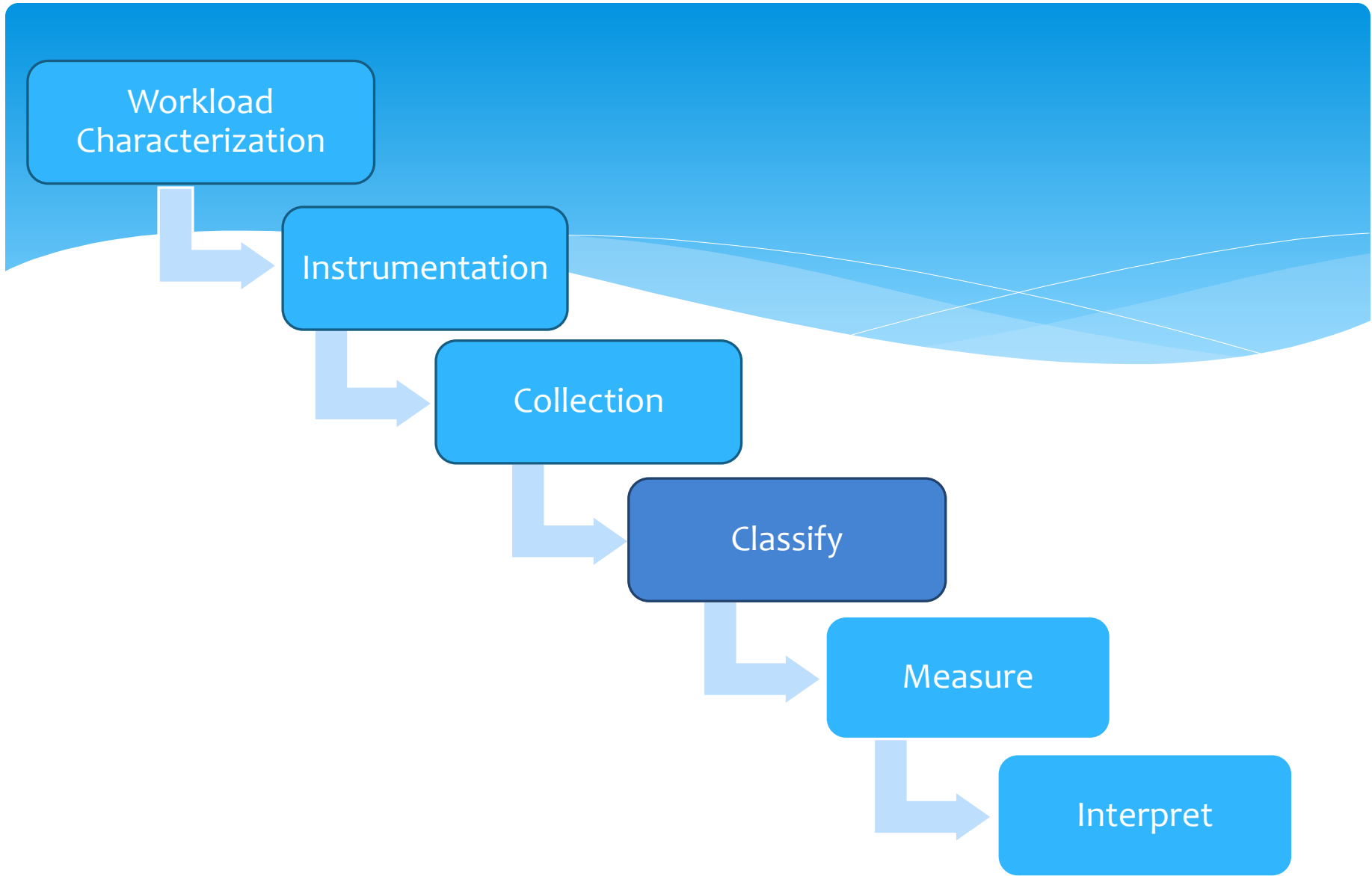
Node id:xyzvzrdo1	Full GC (avg) Secs	Full GC (max) Secs	Full GC (min) Secs	Count	Pause Duration (Secs)
FORMS	0.004344638	0.018386	0.002458	398	1.729166
OACORE	0.012064052	0.08705	0.001699	542	6.538716
OAFM	0.004308295	0.027928	0.002614	1056	4.549559





Collection

- In an Oracle Applications system for example, concurrent manager jobs have a broad enough mix of runtimes that collection by interval is usually not very productive.
- Forms users on the other hand tend to stay connected for a longer period of time and a logoff trigger misses the peaks and valleys of that type of workload.
- OACore JVM usage by modules can provide a 360° view (GCViewer).



Classifying Workload

- Once we have instrumentation in place & data collection done, We need to classify the workload with a functional perspective.
- Some important modules used by the E-Business R12 are:
 - Oracle Financials: Accounts Payable, Accounts Receivable, Fixed Assets, General Ledger
 - Human Resource Management System
 - Customer Support: Service
 - Supply Chain Management (SCM): Logistics, Order Management, Purchasing, Shipping
 - Self-Service Applications: Employee Self-Service, iExpenses, iProcurement, Oracle Time & Labor

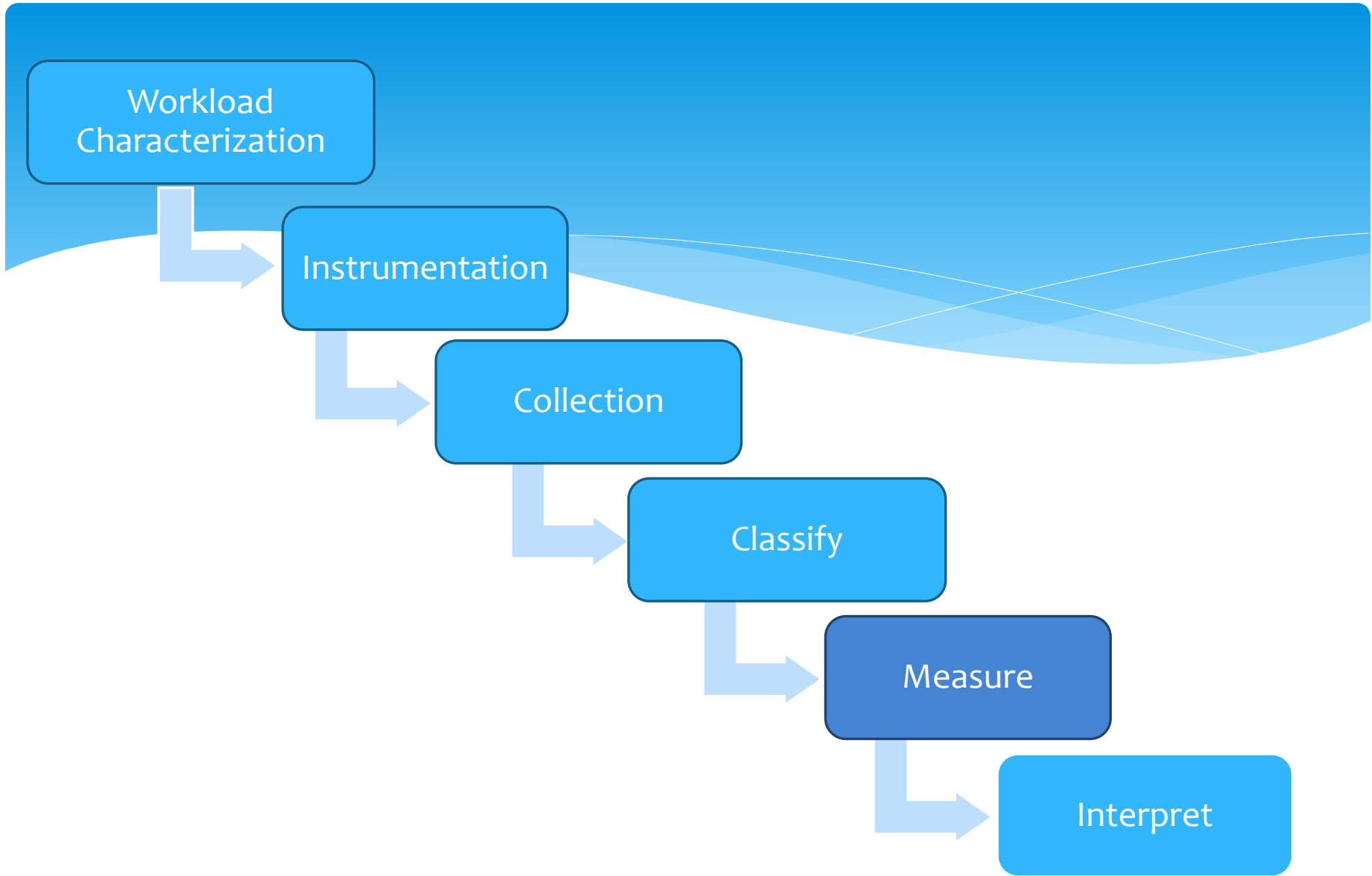
Classifying Workload

- Sample of collecting useful information for classifying the workload.

User Name	Client OS User	Client Machine	Program	Module	Action	Logical Reads	Physical Reads
APPS	applmgr	app1	FNDSCSGN	US PO	Inquiry	3,023	13
APPS	applmgr	app2	POXPOVPO	FRM:AJITH	:US PO Inquiry	3,430	104
APPS	applmgr	app2	FNDSCSGN	US PO	O Inquiry	3,101	26
APPS	applmgr	app2	<u>httpd@app2</u> (TNS V1-V3) V1-V3)	httpd@ app2 (TNS V1-V3)		18,598	679

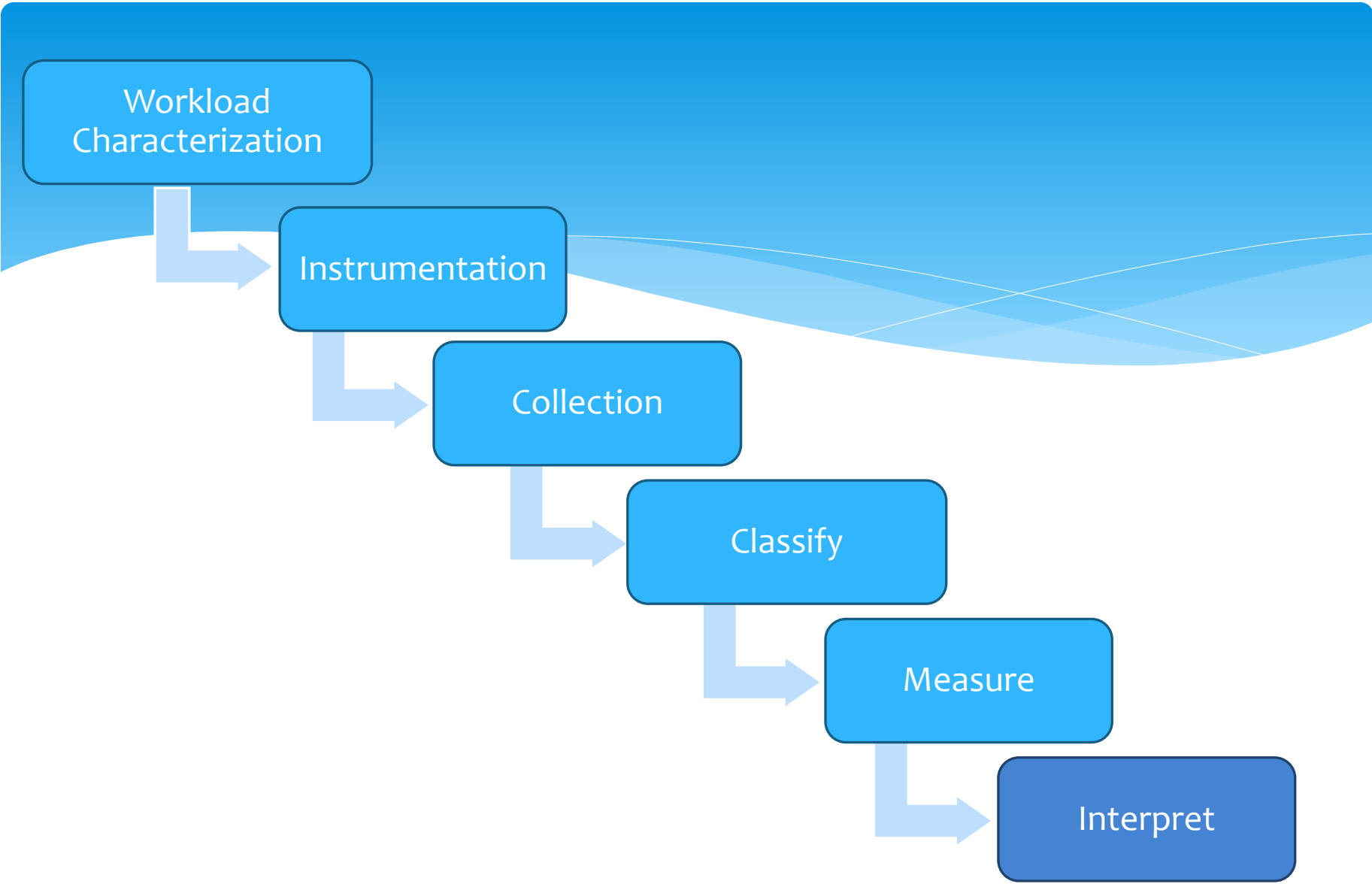


- Once the classification of our system workload is complete, We now will have. the functional perspective of the total work.



Measure

- After workload classes are identified then measurements can be grouped by class and summarized.



Interpret

- Interpretation leads to
- Understanding the overall impact of each workload class on the total workload.
- Charts can be easily created by co-relating the workload class and the resource usage metrics
- Decisions can then be made about work shift allocations or redistribution, and about possibly re-scheduling work to non-critical time periods.

Conclusion

- This session brings awareness of few basic and already existing, small & powerful features in Oracle that can do a lot of help in understanding our system workload,
- These features can provides us with all necessary leads to start tuning the system for optimal performance and sustainability.



Q&A



Thank You!

Contact:- ajithpathiyil@gmail.com