Capture and Replay – create realistic tests

- **Curt Cotner, IBM Fellow**

Big Testing Challenges Faced by Most Customers

- **Most customers have only 10-15% of production workloads automated to run as a regression test.**
- **Often, test systems don't have access to the right mix of application servers to generate production-like transaction volumes.**
- **Even if you had all the right application servers, it is very expensive and labor intensive to actually run a comprehensive test workload that mimics production.**

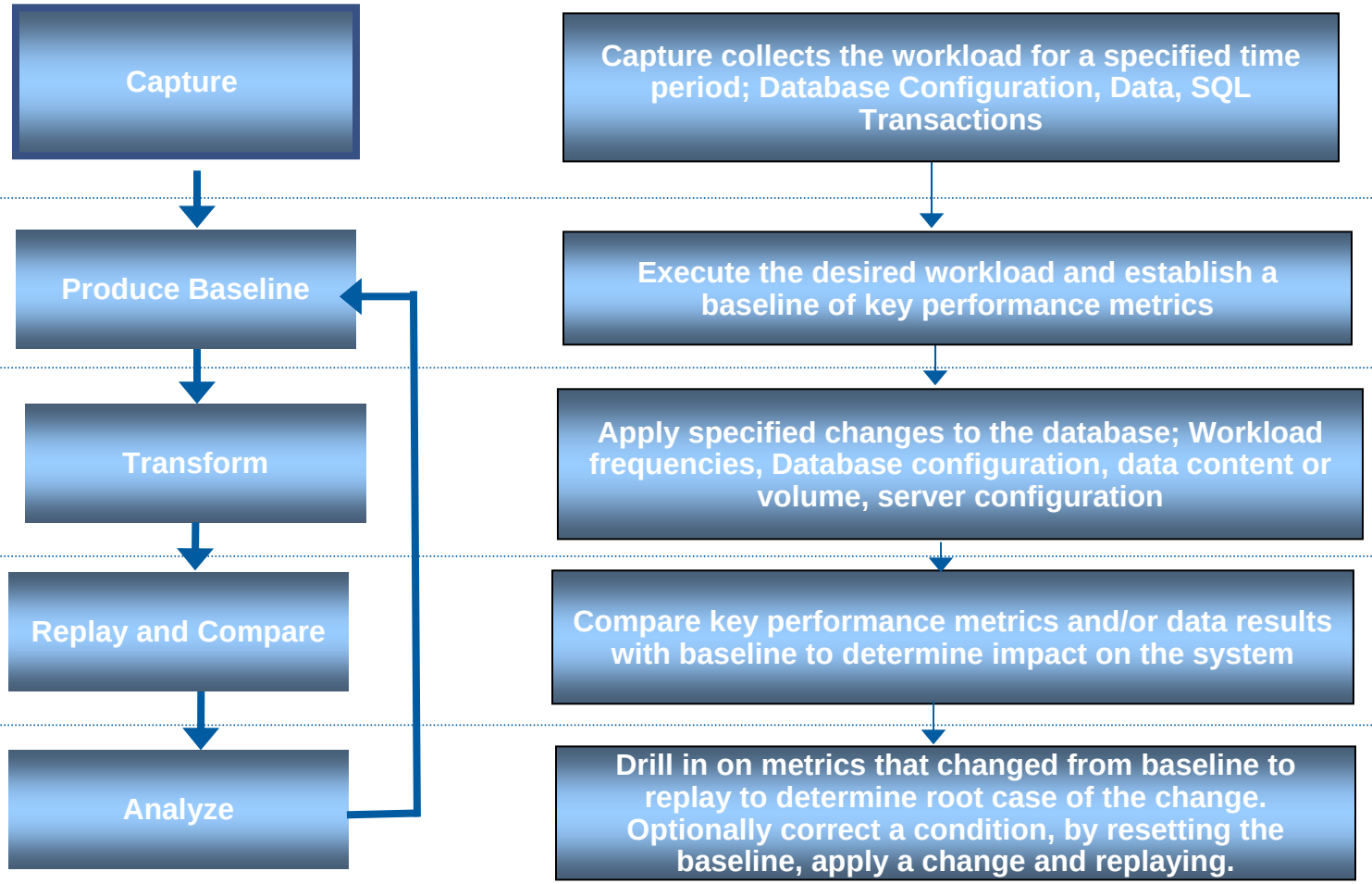
SQL Performance Testing Challenges

- **SQL query cost for a given statement can vary tremendously, which makes it tough to compare one run to another:**
  - Did you get the right access path?
  - Are your statistics current and chosen correctly?
  - Host variable inputs can change cost significantly due to data skew, etc.
  - Cost will also vary based on the number of rows returned by a given query.
  - Are the table conditions the same? (similar number of rows, similar index b-tree depth, etc.)
- **It's both an art and a science --**
  - A complex multi-variable experiment that must be heavily controlled to end up with repeatable results that can be used to make valid decisions…
  - Customers almost never know how to create a repeatable workload that they can use to evaluate performance impact.

Target scenarios – Workload Capture/Replay

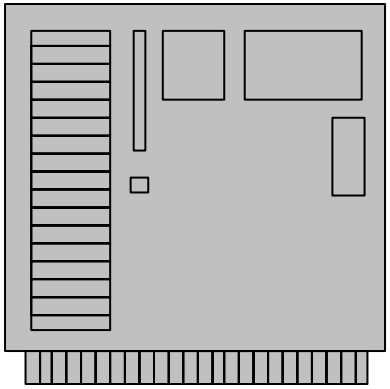
- **Change in Hardware**
  - Platform Switch (move to Linux)
  - O/S Upgrade
- **Change in Workload**
  - Increase in transactions due to expanded application
  - Increase in transactions due to more users
- **Change in Database**
  - Change in schema, index, tablespace, etc.
  - Change in configuration: buffer pool sizes, RUNSTATS, rebind packages, etc.
  - Increased data volume
  - Database upgrade – new version or fixpack
- **Change in Application**
  - Changes to application logic
  - Changes in SQL issued by app (new SQL, modified SQL, omitted SQL, different frequency of SQL statements)
- **Troubleshooting Production Problem**
- **Comparing one workload time period to another (why is Friday mid-day locking so heavy compared to Wed?)**

Proposed Workload Replay Solution - Breakdown

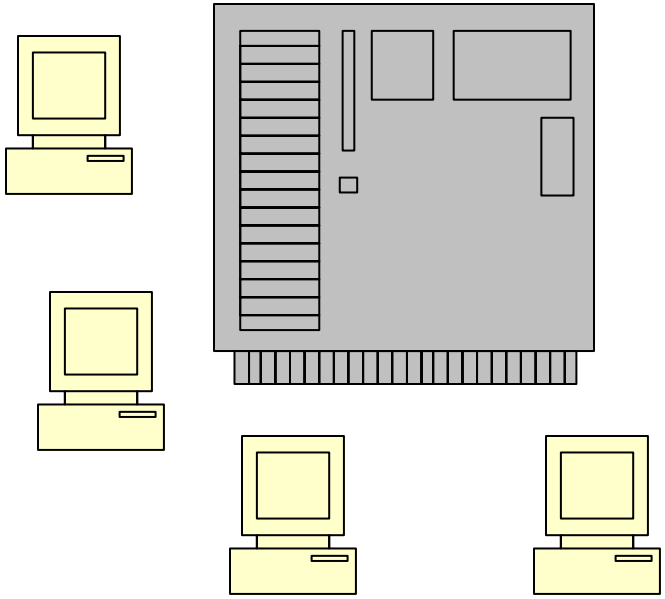


Test Topologies

Database server only



Database server and multiple app servers



Technical challenges – how to minimize capture overhead

- **Many customers run at high CPU utilization**
  - Has been a common practice on z/OS for many years.
  - With the advances in virtualization, this is now widespread on distributed systems also.
- **Capture needs to have minimal impact (3-5%?).**
- **You'd like to avoid duplicate "capture overhead" if you want capture/replay, and auditing, and performance monitoring, and …**

Technical challenges – how to reproduce workload?

- **DB2 workloads can be very complex, especially on z/OS:**
  - Number of DB2 connections can vary tremendously during the day.
  - SQL is submitted in somewhat random order across connections.
  - Different attach mechanisms: RRSAF, CAF, CICS, IMS, DDF, etc.
  - Things like SELECT statements can behave very differently inside DB2 depending upon number of FETCHes you issue, when you issue the FETCHes, whether the cursor is updateable, local vs. remote, etc.
  - All this is further complicated by parallel sysplex, where these things happen across multiple machines concurrently.
- **If your replay is going to be accurate, you need to be able to mimic all these things well.**

Technical challenges – test often differs from production

- **Hardware configuration**
  - Might have fewer data sharing members.
  - Might have less disk space.
  - Might have slower CPUs, less memory, etc.
- **Software configuration:**
  - Different userids/passwords compared to production.
  - Schema names and package collections might differ.
- **Data**
  - Might have only a subset of production data
  - Data might be masked due to PCI or other regulations.
- **How to get the production transaction replay to match the test data (literals, host variables, special registers, schema names, etc.)?**

Optim Solutions

Open | ▼ Welcome x Capture / Replay x

Create Test Database | SQL Workloads

Capture an SQL Workload running against one database and replay it against another database.

Capture... | Transform... | Replay... | Validate... | Report... | More Actions ▼ | Set Up...

Transform SQL Workload: PeakOrders

Database Mapping:

Capture Database	Maps To	Replay Database	Type	Host Name	Port	User ID	Password
ORDERS	=	ORDERST1	DB2 LUW	test1.company.com	50001	DBA123	********
PORDERS	=	ORDERST1	DB2 LUW	test1.company.com	50001	DBA123	********
CUSTORD	=	ORDERST1	DB2 LUW	test1.company.com	50001	DBA123	********

Schema Mapping:

Capture Schema	Maps To	Replay Schema
PROD	=	TEST

Add

Remove

User ID Mapping:

Capture User ID	Maps To	Replay User ID	Replay Password
PRODUSER	=	TESTUSER	*********

Notes:

Mapped dbs, schemas, ids from prod to test

OK | Show Command | Cancel

Optim Solutions

Open | ▼ Welcome x Capture / Replay x

Create Test Database | SQL Workloads

Capture an SQL Workload running against one database and replay it against another database.

Capture... | Transform... | Replay... | Validate... | Report... | More Actions ▼ | Set Up...

Workload Name	Workload Type	Source	Status	Owner	Notes
PeakOrders[0]	Original Capture	ORDERS, ...	02/04/2011 4:00 pm	kmcbride	All peak time activity on the orders database
PeakOrders[1]	Replay Ready	PeakOrders[0]	02/05/2011 8:00 am	kmcbride	Mapped dbs, schemas, ids from prod to test

After transformation, the workload is Replay Ready. Replay… button is enabled.

Technical challenges – how do you uniquely identify transactions?

- **You'd like to be able to make requests like "replay the PAYROLL" workload**
- **Customers running workloads on CICS and IMS have a built-in solution:**
  - incoming transactions are tagged with a transaction name
  - end user names are often provided to DB2
  - static SQL is used heavily, so you usually have package names
- **It is a lot tougher for distributed workloads like WebSphere, Java, and .NET**
  - transaction names, end user names, and static SQL package names are often not available
  - unless you're using technology like pureQuery, you have very little to work with in naming transactions/workloads

Optim Solutions

Open | ▼ Welcome x Capture / Replay x

Create Test Database | SQL Workloads

Capture an SQL Workload running against one database and replay it against another dat

Capture... | Transform... | Replay... | Validate... | Report... | More Actions ▼

Workload Name

PeakOrders[0]
PeakOrders[1]
PeakOrders[2]

database
od to test

Transaction Classification Order helps us group transactions to show aggregate information.

Validate SQL Workload: PeakOrders[2]

Original Capture: PeakOrders

Replay Capture: PeakOrders[2] | ▼

Notes: PeakOrders[2] compared to PeakOrders Original Capture

▼ Transaction Classification Order

1: Client Application Name | ▼ — Not Masked | ▼

2: Client Accounting String | ▼ — Masked | ▼ — From position: 40 to: 65

3: Package Name | ▼

4: Order of SQL Statements | ▼

OK | Show Command | Cancel

Optim Solutions

Open | ▼ Task Launcher x Capture and Replay x **PeakOrders[3] Report** x

Replay Success and Response Time tabs appear when the report is complete. Use the links on those tabs to drill-down into rep

Save as New Replay Ready Workload...

Details | **Replay Results** | Response Time

Use replay success to evaluate how closely the replay workload matches the baseline workload.

SQL Replay Success Metric	Unique	Executions	Percentage	Description
Baseline SQL	10,000	1,000,000		All SQL statements in the baseline workload
Matched Replays - SQL	10,000	1,000,000	100%	SQL with the same return codes, rows returned, and rows updated
Unmatched Replays - SQL	0	0	0%	SQL with different return codes, rows returned, or rows updated
• Different SQL Return Codes	0	0	0%	SQL with different return codes in baseline and replay
• Different # Rows Returned	0	0	0%	SQL with different rows returned in baseline and replay
• Different # Rows Updated	0	0	0%	SQL with different rows updated in baseline and replay
• Missing SQL	0	0	0%	SQL that was in the baseline but is missing in the replay
New SQL	0	0		SQL that was not in the baseline, but was found in the replay

Transaction Replay Success Metric	Unique	Executions	Percentage	Description
Baseline Transactions	800	80,000		All transactions in the baseline workload
Matched Replays - Transactions	800	80,000	100%	Transactions where all SQL was successfully replayed
Unmatched Replays - Transactions	0	0	0%	Transactions where one or more SQL failed to replay
• Different SQL Return Codes	0	0	0%	Transactions where one or more SQL had different return codes
• Different # Rows Returned	0	0	0%	Transactions where one or more SQL had different rows returned
• Different # Rows Updated	0	0	0%	Transactions where one or more SQL had different rows updated
• Missing Transactions	0	0	0%	Transactions that were in the baseline but are missing in the replay
New Transactions	0	0		Transactions that were not in the baseline, but were found in the replay

Technical challenges – how to tell if replay performs and scales?

- **When replaying the workload, you'd like to understand how replay compares to the original workload:**
  - Are you seeing similar patterns in the workload peaks/valleys?
  - Are you encountering bottlenecks (peaks that get "flattened")?
  - Are you getting similar transaction throughput?

- **You'd like to be able to speed up or slow down the replay to study things like:**
  - Can my workload scale to 2X of my current peak workload?
  - Do I start to see I/O or locking problems?
  - If I encounter these problems, how do I isolate the cause?

Optim Solutions

Open | ▼ | Task Launcher x | Capture and Replay x | **PeakOrders[3] Report** x

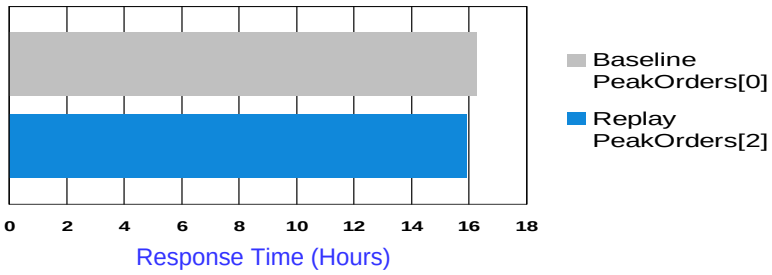
Replay Success and Response Time tabs appear when the report is complete. Use the links on those tabs to drill-down into rep

Save as New Replay Ready Workload...

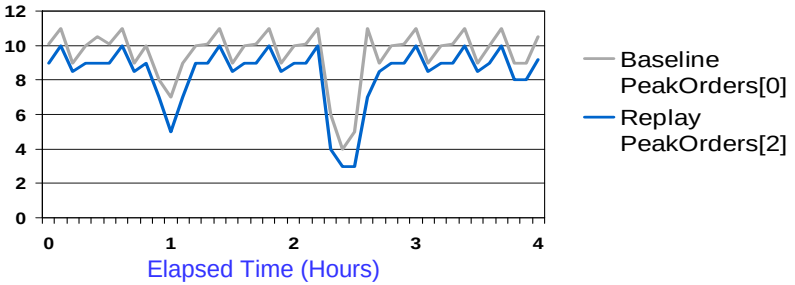
Details | Replay Results | **Response Time**

Use the response time tab to identify improvements and regressions in performance between the baseline and replay workloads.

Total Statement Response Time

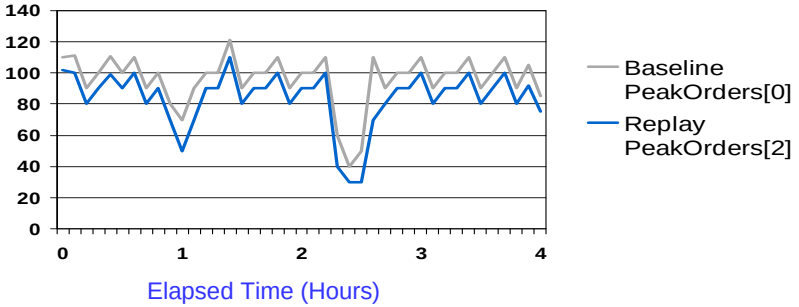


SQL Executions (1000 / second)



Metric	Value	Percentage
Total Response Time Difference	00:19:30	2%
Total Improvements	00:37:50	4%
Total Regressions	00:18:20	-2%
SQL >= 5% Improvement	300 / 10000	3%
SQL >= 5% Regression	200 / 10000	2%
Transactions >= 5% Improvement	8 / 800	1%
Transactions >= 5% Regression	16 / 800	2%
Baseline Elapsed Time	04:00:00	
Replay Elapsed Time	03:53:30	

Rows Returned (10,000 / second)



Optim Solutions

Open | ▾ | Task Launcher x | Capture and Replay x | **PeakOrders[3] Report** x

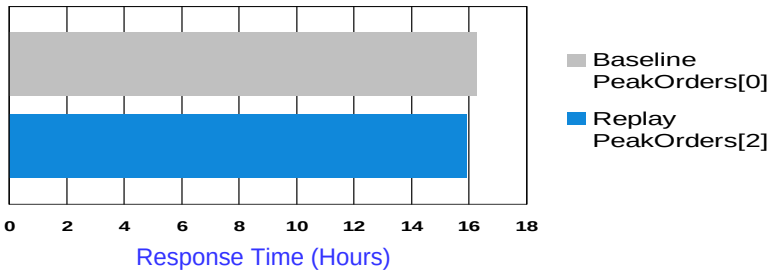
Replay Success and Response Time tabs appear when the report is complete. Use the links on those tabs to drill-down into rep

Save as New Replay Ready Workload...

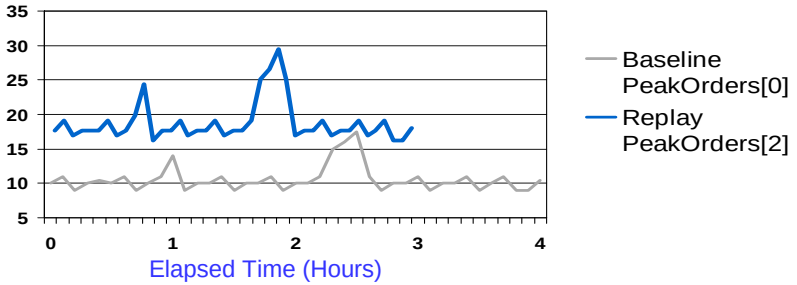
Details | Replay Results | **Response Time**

Use the response time tab to identify improvements and regressions in performance between the baseline and replay workloads.

Total Statement Response Time

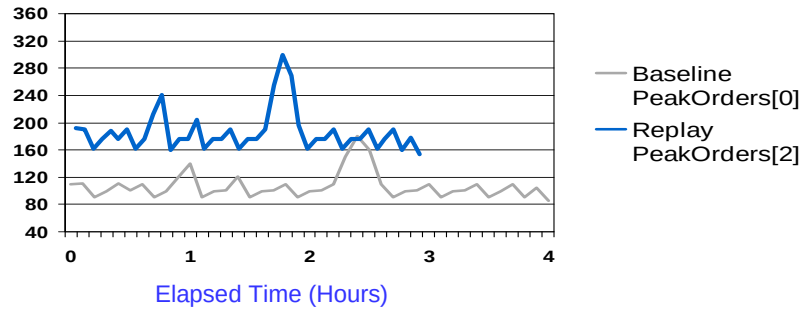


SQL Executions (1000 / second)



Metric	Value	Percentage
Total Response Time Difference	00:19:30	2%
Total Improvements	00:37:50	4%
Total Regressions	00:18:20	2%
SQL >= 5% Improvement	300 / 10000	3%
SQL >= 5% Regression	200 / 10000	2%
Transactions >= 5% Improvement	8 / 800	1%
Transactions >= 5% Regression	16 / 800	2%
Baseline Elapsed Time	04:00:00	
Replay Elapsed Time	03:00:30	

Rows Returned (10,000 / second)



Optim Solutions

Open | ▾ | Task Launcher x | Capture and Replay x | **PeakOrders[3] Report** x

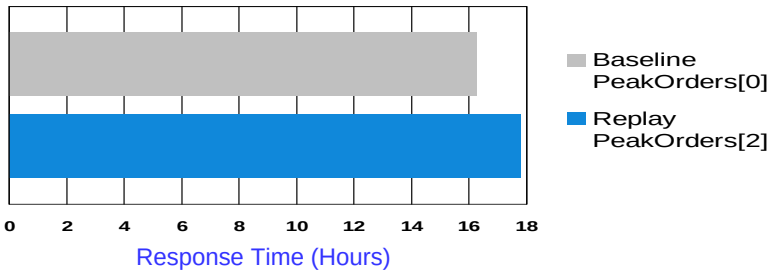
Replay Success and Response Time tabs appear when the report is complete. Use the links on those tabs to drill-down into rep

Save as New Replay Ready Workload...

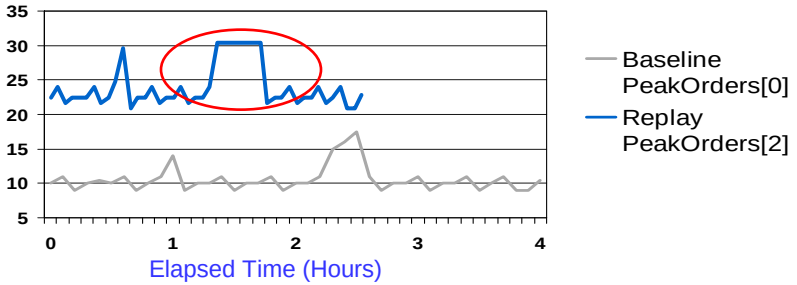
Details | Replay Results | **Response Time**

Use the response time tab to identify improvements and regressions in performance between the baseline and replay workloads.

Total Statement Response Time

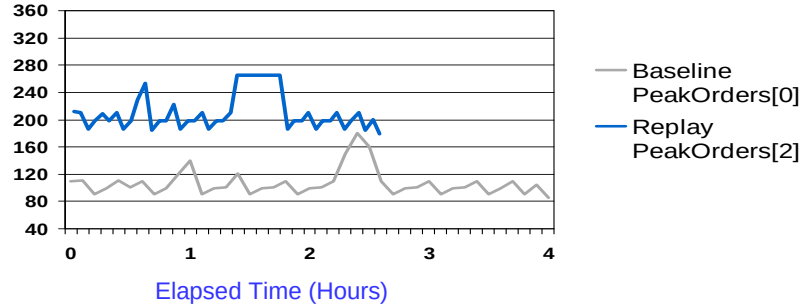


SQL Executions (1000 / second)



Metric	Value	Percentage
Total Response Time Difference	01:37:30	10%
Total Improvements	00:19:50	2%
Total Regressions	01:58:20	2%
SQL >= 5% Improvement	200 / 10000	2%
SQL >= 5% Regression	1000 / 10000	10%
Transactions >= 5% Improvement	8 / 800	1%
Transactions >= 5% Regression	40 / 800	5%
Baseline Elapsed Time	04:00:00	
Replay Elapsed Time	02:40:30	

Rows Returned (10,000 / second)



Optim Solutions

Open | ▼ Welcome x Capture / Replay x

Create Test Database | SQL Workloads

Capture an SQL Workload running against one database and replay it against another database.

Capture... | Transform... | Replay... | Validate... | Report... | More Actions ▼ | Set Up...

Workload Name	Workload Type	Source	Status	Owner	Notes
PeakOrders[0]	Original Capture	ORDERS, ...	02/04/2011 4:00 pm	kmcbride	All peak time activity on the orders database
PeakOrders[1]	Replay Ready	PeakOrders[0]	02/05/2011 8:00 am	kmcbride	Mapped dbs, schemas, ids from prod to test
PeakOrders[2]	Replay Capture	PeakOrders[1]	02/05/2011 2:00 pm	kmcbride	Baseline replay test
PeakOrders[3]	Validation Report	PeakOrders[2]	02/06/2011 9:00 am	kmcbride	PeakOrders[2] compared to PeakOrders Original
PeakOrders[4]	Replay Ready	PeakOrders[1]	02/06/2011 10:00 am	kmcbride	Invalid transactions removed from PeakOrders[1]
PeakOrders[5]	Replay Capture	PeakOrders[4]	02/06/2011 2:00 pm	kmcbride	Replay with invalid transactions removed

Another replay capture appears. The user can select the Report… button for performance reports.

Optim Solutions

Open | ▼ Welcome x Capture / Replay x

Create Test Database | SQL Workloads | Validation Report x

Overview > Different Return Codes

Save Workload...

+100 Return Codes – The data from the original capture environment is not present in the replay environment.

☐	Statement Text	Original RC	New RC	Description
☐	UPDATE DBPARTITION...	0	+100	Row not found or end of cursor.
☐	INSERT T1.AGENT_ID ...	0	+100	Row not found or end of cursor.
☐	UPDATE DBPARTITION...	0	+100	Row not found or end of cursor.
☐	INSERT T2.AGENT_ID	0	+100	Row not found or end of cursor.
☐	SELECT * FROM T3 ...	0	+100	Row not found or end of cursor.

Select All

Deselect All

Remove Transactions

☐	Statement Text	Original RC	New RC	Description
☐	UPDATE DBPARTITION...	0	-204	Object not defined to DB2.
☐	INSERT T1.AGENT_ID ...	0	-204	Object not defined to DB2.
☐	UPDATE DBPARTITION...	0	-205	Column name not in table.
☐	INSERT T2.AGENT_ID	0	-206	Column name not in table.
☐	SELECT * FROM T3 ...	0	-206	Column does not exist in any table of the SELECT.

Select All

Deselect All

Remove Transactions

☐	Statement Text	Original RC	New RC	Description
☐	UPDATE DBPARTITION...	0	-551	Authorization failure
☐	INSERT T1.AGENT_ID ...	0	-551	Authorization failure
☐	UPDATE DBPARTITION...	0	-922	Authorization needed
☐	INSERT T2.AGENT_ID	0	-551	Authorization failure
☐	SELECT * FROM T3 ...	0	-551	Authorization failure

Select All

Deselect All

Remove Transactions

-551, -922 Return Codes – The result of the original SQL execution is different in the replay environment.

Optim Solutions

Open | ▼ | Task Launcher x | Capture and Replay x | **PeakOrders[9] Report** x

Replay Success and Response Time tabs appear when the report is complete. Use the links on those tabs to drill-down into rep

Save as New Replay Ready Workload...

Details | Replay Results | Response Time | **SQL Improvements** x

SQL with the greatest performance improvement – where response time is shorter in the replay than in the baseline.

SQL Improvements

Statement Identifier	Statement Text	Baseline Executions	Replay Executions	Change in Executions	Baseline Total Response Time	Replay Total Response Time	Total Response Time Change ▼	Percentage Total Response Time Change	Baseline Average Response Time	Replay Average Response Time	Average Response Time Change	Percentage Average Response Time Change
AABCCD	SELECT T2.AGENT_ID ...	100	100	0	00:30:50.8	00:14:20.8	-00:15:55.3	-50%	00:00.085234	00:00.059234	-00:00.027	-50%
AABCWR	SELECT T1.AGENT_ID ...	345	345	0	00:16:35.4	00:05:55.4	-00:11:30.5	-70%	00:13.633456	00:12.433456	-00:01.208	-10%
ZZZHGD	SELECT DBPARTITIONN...	15454	15454	0	00:30:55.6	00:22:30.6	-00:08:25.9	-5%	00:01.393567	00:01.223567	-00:00.176	-5%
ZZZH45	SELECT T2.AGENT_ID ...	4443	4443	0	00:15:28.3	00:08:08.3	-00:07:22.4	-32%	00:01.013432	00:00.821342	-00:00.286	-32%
Z35HTR	SELECT DBPARTITIONN...	56	56	0	00:05:01.7	00:03:35.7	-00:01:15.7	-27%	00:00.695432	00:00.565432	-00:00.133	-27%
Q89EDS	SELECT T1.AGENT_ID ...	345	345	0	00:16:04.4	00:14:55.4	-00:01:09.8	-10%	00:14.133434	00:12.433434	-00:01.208	-10%
ZRZH77	SELECT DBPARTITIONN...	15454	15454	0	00:30:35.6	00:29:30.6	-00:01:05.3	-5%	00:01.473232	00:01.223232	-00:00.176	-5%
RBDEDS	SELECT T2.AGENT_ID ...	4443	4443	0	00:05:06.3	00:04:08.3	-00:00:59.5	-10%	00:13.333234	00:12.433234	-00:01.208	-10%
PJZHGD	SELECT DBPARTITIONN...	56	56	0	00:04:30.5	00:03:35.7	-00:00:55.9	-5%	00:01.453453	00:01.223453	-00:00.176	-5%
GGDED	INSERT T2.AGENT_ID ...	307	307	0	00:15:32.7	00:14:55.4	-00:00.48.1	-32%	00:01.123768	00:00.821768	-00:00.286	-32%

1 – 10 of 100 ◄ **1** 2 3 4 5 6 7 8 9 ... 10 ► **10** | 25 | 50 | 100

SQL Statement Comparison Drill-down

Optim Performance Manager

SQL Statement Comparison Report

Compare performance details of this statement across the two workload runs

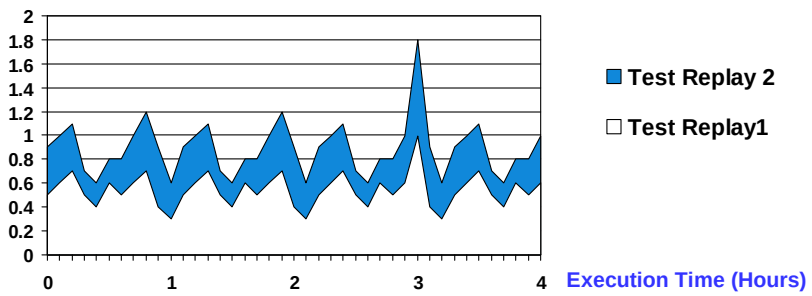
SQL Statement

SELECT B.COL1, B.COL3, B.COL5, B.COL6, B.COL12 FROM T1.SETLMNT, BRANCH B, ADDR A WHERE S.TRANS_NO = ?, AND S.TRANS_PROC_DT < '9999-12-31' AND YEAR (S.TRANS_TARGET_DT) = '2002' AND S.TRANS_TYPE IN ('A1', 'A2', 'A3', 'Z9') AND S.TRANS_CD IN ('EOD', 'IMD', 'UGT') AND S.TRANS_SETL_DT = ? AND B.BRANCH_EFF_DT <= ? AND B.BRANCH_INACTIVE_DT > ?

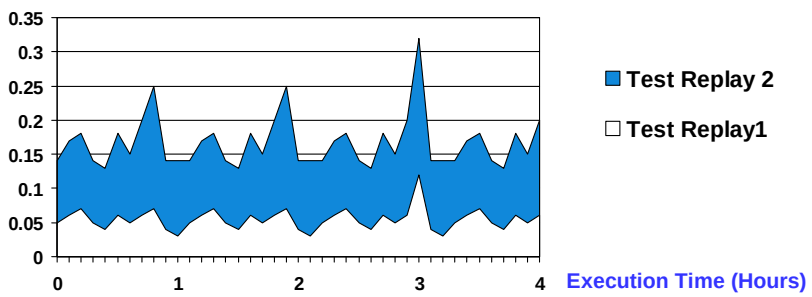
Tune SQL

Metric	Test Replay 1	Test Replay 2	% Change
Executions	508	508	0%
Average Elapsed Time (sec)	0.567	0.876	+45%
Total Elapsed Time (sec)	254.453	367.463	+45%
Average CPU Time (sec)	0.0567	0.1376	+275%
Total CPU Time (sec)	25.4567	69.876	+275%
Average System CPU Time (sec)	0.0062	0.0121	+175%
Total System CPU Time (sec)	2.3445	6.6503	+175%
Average User CPU Time (sec)	0.0434	0.1221	+275%
Total User CPU Time (sec)	20.432	57.876	+275%
Average Get Pages	4.01	4.40	+15%
Total Get Pages	2000	2300	+15%
Sorts	0	0	0%
Table Scans	0	0	0%

Average Elapsed Time (seconds)



Average CPU Time (seconds)



Optim Solutions

Open | ▾ | Welcome x | Capture / Replay x

Create Test Database | SQL Workloads | Performance Report x

Top 'N' Transaction Comparison

Sort by: Total Response Time Change | ▾ Number of Statements: 5 | ▾ Show: Both Regressions and Improvements | ▾

Transaction Regressions

Transactions	Type	SQL Statements	Total Response Time			Average Response Time			Rows Updated (changes)	Rows Returned (changes)	Return Code (Changes)
			Baseline (sec)	Change (sec) ▼	Change (%)	Baseline (sec)	Change (sec)	Change (%)			
APPNAME23	App Name	25	200.849	+100.427	+50%	0.059	+0.027	+50%	0	0	0
ACCTSTR456	App Name	5	896.433	+90.708	+10%	12.433	+1.208	+10%	0	0	0
ACCTSTR789	Acnt Str	73	1765.623	+85.676	+5%	1.223	+0.176	+5%	0	0	0
PKGNM123	Package	15	248.321	+78.786	+32%	0.821	+0.286	+32%	0	0	0
SQL_SEQ_567	SQL Seq	75	215.765	+75.653	+27%	0.565	+0.133	+27%	0	0	0

Transaction Improvements

Transactions	Type	SQL Statements	Total Response Time			Average Response Time			Rows Updated (changes)	Rows Returned (changes)	Return Code (Changes)
			Baseline (sec)	Change (sec) ▼	Change (%)	Baseline (sec)	Change (sec)	Change (%)			
SQL_SEQ_765	SQL Seq	15	1874.321	-195.427	-12%	10.874	-22.337	-12%	0	0	0
SQL_SEQ_988	SQL Seq	43	135.987	-120.7083	-95%	0.421	-0.398	-95%	0	0	0
ACCTSTR333	Acnt Str	20	1201.787	-55.676	-5%	0.123	-0.059	-5%	0	0	0
ACCTSTR555	Acnt Str	1	86.874	-20.786	-23%	0.013	-0.007	-23%	0	0	0
APPNAME767	App Name	56	753.765	-15.653	-2%	15.345	-1.334	-2%	0	0	0

Optim Solutions

Open | ▼ Welcome x Capture / Replay x

Create Test Database | SQL Workloads | Performance Report x

Top N Transactions Report > SQL List for Transaction APPNAME23

SQL List for Transaction APPNAME23

Statement Text	Baseline Executions	Change in Executions	Total Response Time			Average Response Time			Rows Updated (changes)	Rows Returned (changes)	Return Code (Changes)
			Baseline (sec)	Change (sec) ▼	Change (%)	Baseline (sec)	Change (sec)	Change (%)			
UPDATE DBPARTITION...	10050	0	200.849	+100.427	+50%	0.059	+0.027	+50%	0	0	0
INSERT T1.AGENT_ID ...	25	0	896.433	+90.708	+10%	12.433	+1.208	+10%	0	0	0
UPDATE DBPARTITION...	2234	0	1765.623	+85.676	+5%	1.223	+0.176	+5%	0	0	0
INSERT T2.AGENT_ID ...	307	0	248.321	+78.786	+32%	0.821	+0.286	+32%	0	0	0
SELECT * FROM T3 ...	529	0	215.765	+75.653	+27%	0.565	+0.133	+27%	0	0	0
SELECT T2.AGENT_ID ...	100	0	1874.321	-195.427	-12%	10.874	-22.337	-12%	0	0	0
SELECT T1.AGENT_ID ...	345	0	135.987	-120.7083	-95%	0.421	-0.398	-95%	0	0	0
SELECT DBPARTITION...	15454	0	1201.787	-55.676	-5%	0.123	-0.059	-5%	0	0	0
SELECT T2.AGENT_ID ...	4443	0	86.874	-20.786	-23%	0.013	-0.007	-23%	0	0	0
SELECT DBPARTITION...	56	0	753.765	-15.653	-2%	15.345	-1.334	-2%	0	0	0
SELECT T2.AGENT_ID ...	100	0	1874.321	-195.427	-12%	10.874	-22.337	-12%	0	0	0
SELECT T1.AGENT_ID ...	345	0	135.987	-120.7083	-95%	0.421	-0.398	-95%	0	0	0
SELECT DBPARTITION...	15454	0	1201.787	-55.676	-5%	0.123	-0.059	-5%	0	0	0

Thank You for Joining Us today!

Go to
www.ibm.com/software/systemz/events/calendar **to:**

- Replay this teleconference
- Replay previously broadcast teleconferences
- Register for upcoming events

Learn more

- **Additional resources for InfoSphere Optim Query Capture and Replay for DB2 on z/OS**
  - Product webpage
  - eBook: "Enterprise change testing in DB2 for z/OS: A confidence endeavor"
  - Online demo on developerWorks also available on YouTube