Relaxed Releases with DB-Workload-Tests

NEW STARS OF DATA

08/20

Download at

https://tinyurl.com/mgstars

ABOUT ME

- Martin Guth (37)
- Working 11 years as a BI Engineer (built a new DWH from scratch)
- Working 5 years as a DBA (focus on performance tuning)
- Current Role „Data Swiss-Knife“
  at 3C Deutschland GmbH (Experian) in Heilbronn, Germany
  - Job Opening for a DBA (requires German B1)
- Excited about PASS since 2008
- Blog: www.martinguth.de
- Contact: martin_guth@hotmail.com

Agenda

1. Why Workload Testing?
2. Perform Workload Testing using Workload Tools
3. Demos
4. Tips & Tricks
5. Lessons Learned
6. Questions

WHY WORKLOAD-TESTING?

Photo by Edho Pratama on Unsplash

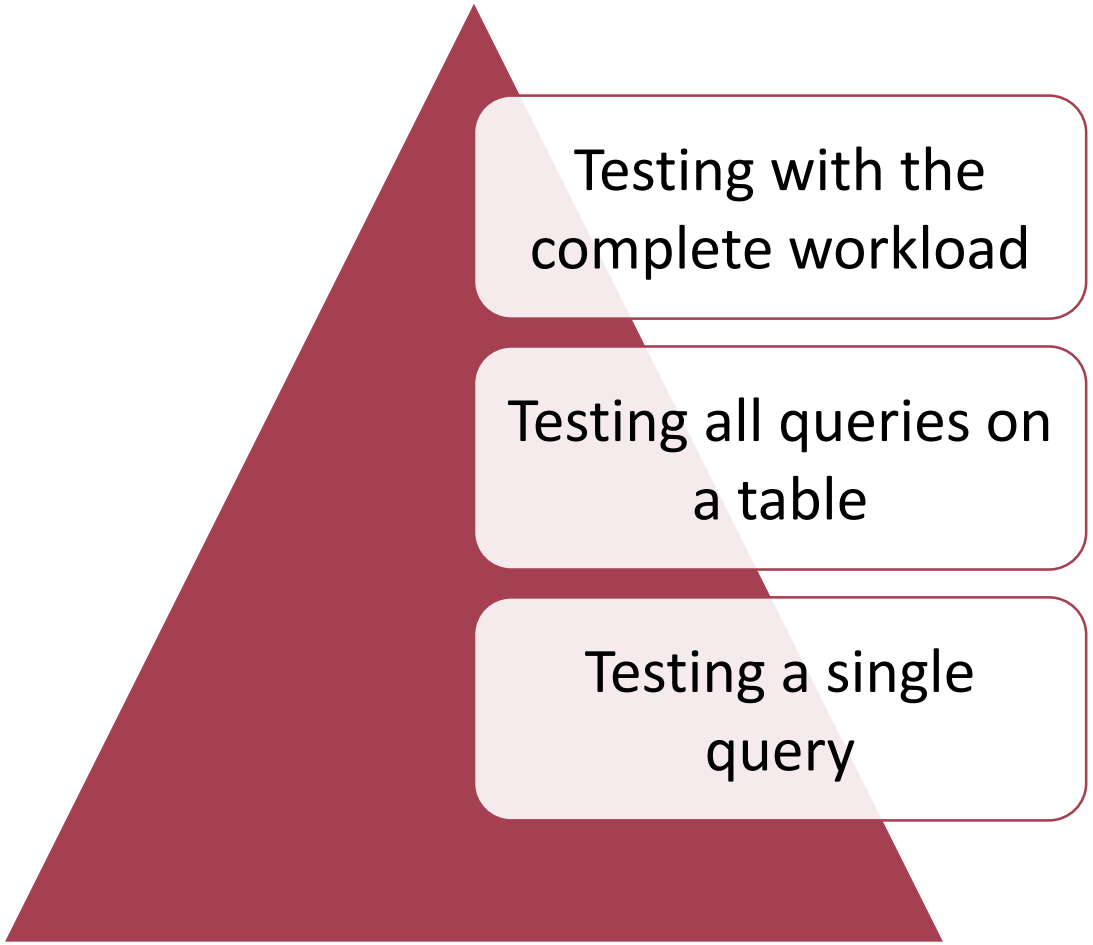
Photo by Linus Nylund on Unsplash

Small Changes Big Impact

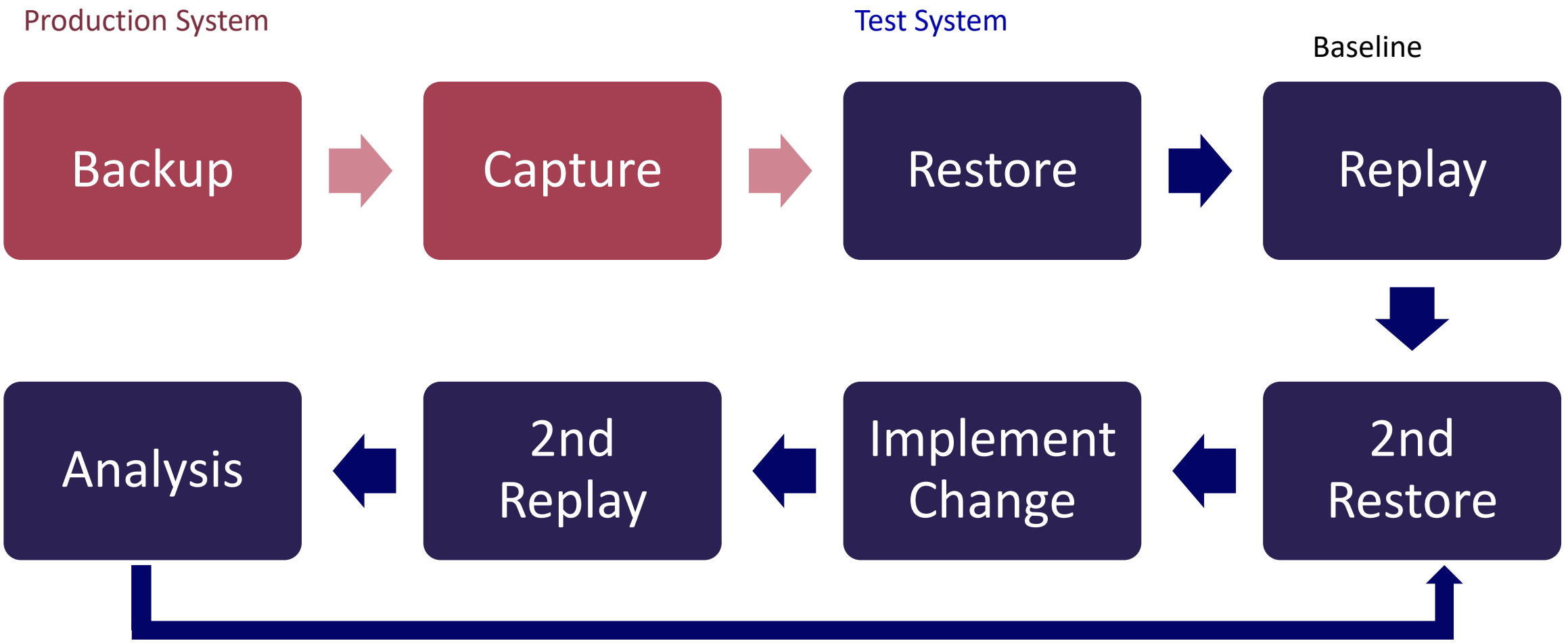
- Example performance tuning: Altering an index
  - Queries/DML getting faster
  - Queries/DML getting slower
- Some other examples
  - Usage of new features (e.g. ADR)
  - Adjustment of db-settings (e.g. MAXDOP)
- Courses of action
  - No risk no fun
  - Never change a running system
  - Test thoroughly in advance

Testing sounds fine…but how?

workload = everything which is being executed on the server in a specific time interval…in essence read **and write** activity



Conduction DB-Workload-Tests (A/B-Testing)



Some Thoughts on the Test System

- Sizing
  - Provide sufficient storage capacity (disks can be slower than in production)
  - Provide the same amount of RAM as in production
  - Provide the same CPU-Config as in production
- Set SQL-Server-Settings for the instance (e.g. MAXDOP) identically as in production
- Due Diligence: Watch out for GDPR/CCPA and security concerns
- Workaround
  - Separate DB on your production instance
  - Run workload tests in off-peak-hours (e.g. weekend)
  - CAUTION: Ensure that the workload test doesn‘t access your production databases (e.g. separate login w/o access on prod)

PERFORM WORKLOAD TESTING USING WORKLOAD TOOLS

Photo by Ēriks Irmejs on Unsplash

Workload Tools

- Developed by MVP Gianluca Sartori (spaghettidba)
- Capture using Profiler, Trace or XEvents
- Supports SQL Server from Version 2000 (tested with 2005 by the author)
- Capture gets saved as SQLite DB
- Integrated tool for replay
- Download via Github

Components

SqlWorkload

- Capture and replay of workloads
- Command line

WorkloadViewer

- Comparison of test-results
- GUI

ConvertWorkload

- Converting trace files
- Command line

WorkloadWizard

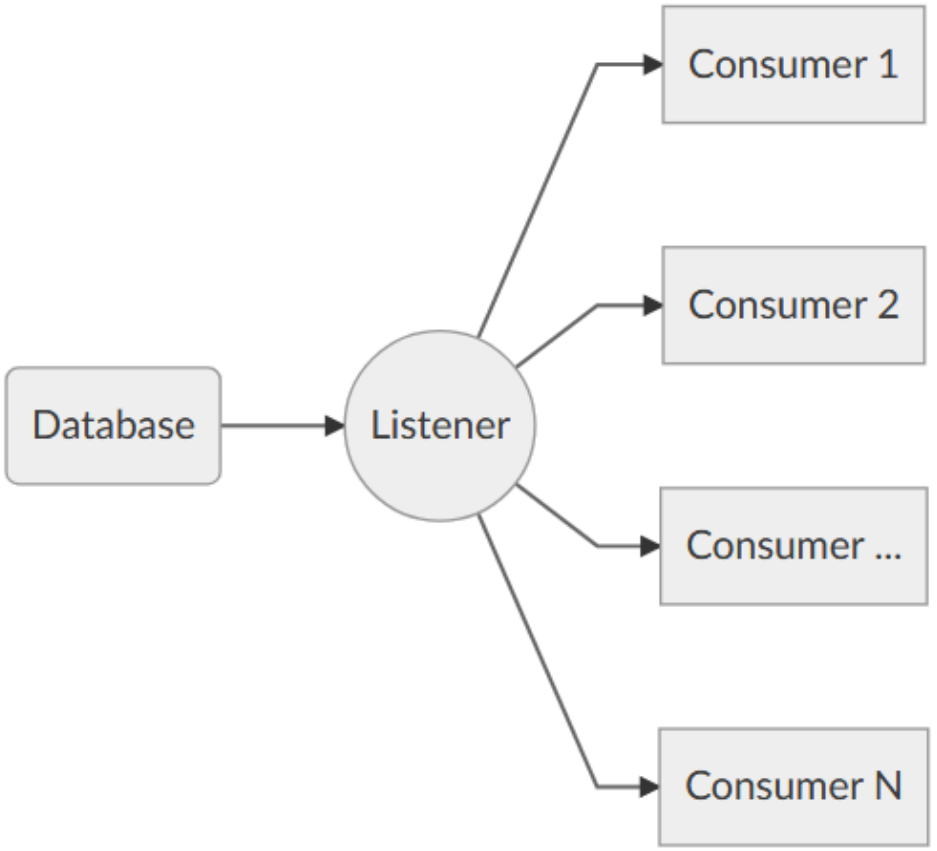
- GUI for easier config
- Not yet available

Configuration

Listener (always one)

Capture from Source
- ExtendedEventsWorkloadListener
- ProfilerWorkloadListener
- SqlTraceWorkloadListener

Read from Capture
- FileWorkloadListener



Consumer (0..n)

Write workload to a file
- WorkloadFileWriterConsumer

Gather analysis data
- AnalysisConsumer

Replay the workload
- ReplayConsumer

JSON-Format

Source: spaghettidba.com

Sample Configuration for Capture

```
{    "Controller": {
        "Listener":
        {
            "__type": "ExtendedEventsWorkloadListener",
            "ConnectionInfo":
            {
                "ServerName": "mydbserver"
            },
            "DatabaseFilter": "mydb"
        },
        "Consumers":
        [
            {
                "__type": "WorkloadFileWriterConsumer",
                "OutputFile": "C:\\temp\\mydb.sqlite"
            }       ]    }}
```

Explanation:

① Recording the workload using XEvents

② Connecting to DB-Server *mydbserver* via Windows-Authentication

③ Recording just the workload on database *mydb* (optional)

④ Saving the workload to a SQLite DB for subsequent replay

Complete reference on Github

Some words on the replay-process

- Each recorded session (SPID) gets it‘s own ReplayWorker-Process
- Executes commands in this session one after another
- Possible problem: Sequence of execution on the test system may differ from the one captured on production
  - ReplayWorker-Processes run independently from one another
  - Example: SPID 100 INSERT of new row; SPID 101 UPDATE of this new row
  - Replay can result in FK violation, if replay for SPID 101 is faster than replay for SPID 100

Replay Options

- Sync (synchronizationMode = true)
  - Respects pauses in the workload
  - Replay-Duration resembles capture-duration
- Async (synchronizationMode = false)
  - Fires commands as quickly as possible
  - Replay-Duration can be significantly shorter than capture -duration
  - Capacity of test-system pushed harder
  - Errors due to wildly different order of execution more likely (e.g. FK violations)

18

Sample Configuration for Replay

```
{    "Controller": {
       "Listener":
       {
           "__type": "FileWorkloadListener",
           "Source": "C:\\temp\\mydb.sqlite",
           "SynchronizationMode": "true"
       },
       "Consumers":
       [
         {
            "__type": "ReplayConsumer",
            "ConnectionInfo":
            {
               "ServerName": "replayServer",
               "DatabaseName": "mydb"
            },
           "ConsumeResults": "false",
           "QueryTimeoutSeconds": 180
         }
```

①

②

③

④

```
{
   "__type": "AnalysisConsumer",
   "ConnectionInfo":
    {
      "ServerName": "replayServer",
      "DatabaseName": "WorkloadAnalysis",
      "SchemaName": "baseline"
    },
    "UploadIntervalSeconds": 60
}]    }}
```

⑤

Explanation:

① Read a captured workload

② Synchronous replay

③ Replay gets executed on *replayServer* in *mydb (integrated authentication)*

④ Additional configuration for replay (optional):

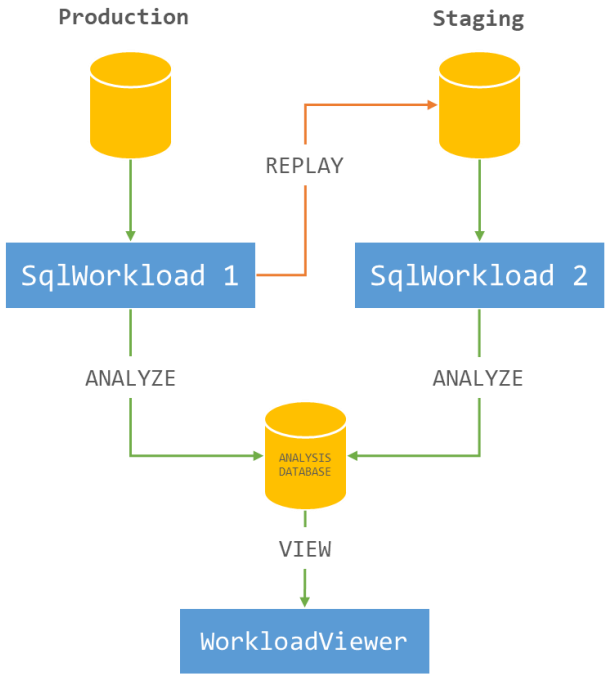
- Discard results after execution (consumeResults *false*)
- Command Timeout after 3 minutes

⑤ Workload-Analysis uploaded to schema *baseline* of database *WorkloadAnalysis* every minute

Complete reference on Github

Configuration: Discover the possibilities

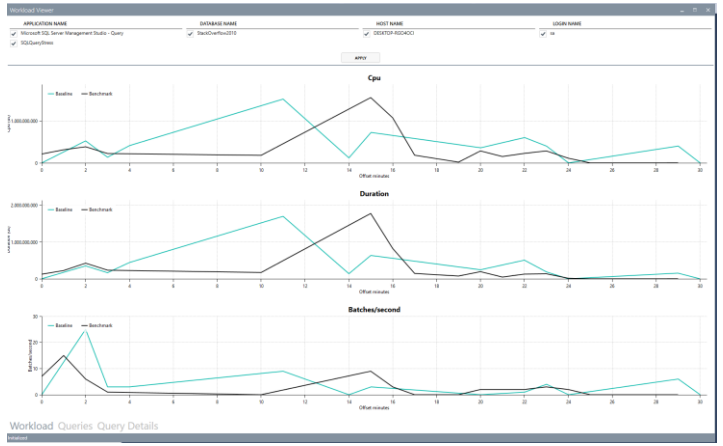
- Advanced Configuration
  - Workload-Analysis during Capture → mother of all baselines
  - Real-Time-Replay



Source: spaghettidba.com

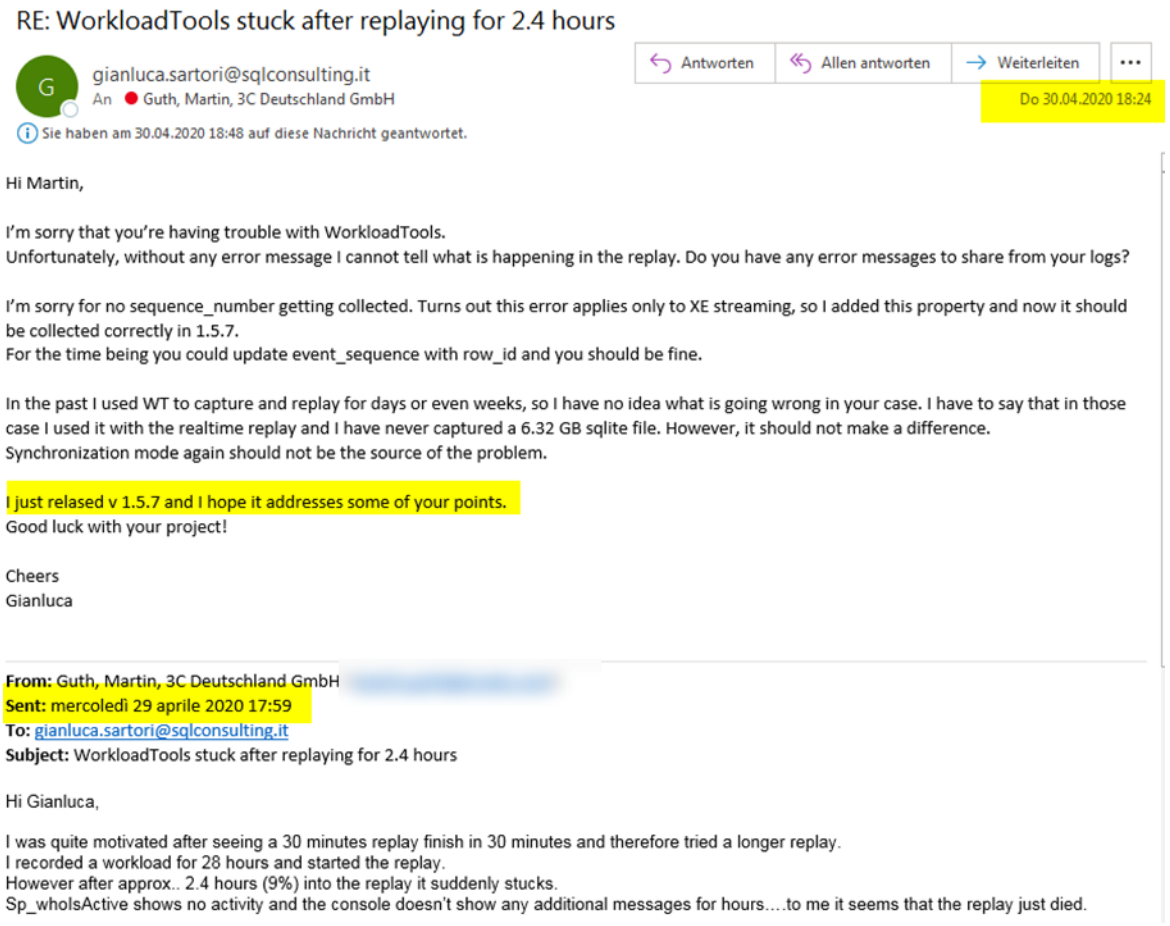
Analysis of Workload-Test-Results

- WorkloadViewer by Gianluca Sartori
  - Integrated in WorkloadTools
  - Core metrics CPU time and elapsed time
  - No execution plans (as of now)
- Query Store Comparison Script by John Sterrett
  - Analysis covers reads and writes as well
  - Execution Plans available from Query Store fo in-depth analysis



ExecutionD...	CompareExecutions	BaselineExecutions	DurationDelta	CPUDelta	ReadDelta	WriteDelta	CompareReads	BaselineReads	CompareCPU	BaselineCPU	CompareDuration	BaselineDuration	query_hash
0	1767	1767	-12053358	-11903430	-25271941	0	264366	25536307	2953516	14856946	4364968	16418326	0x6438E6BF71C4803B
0	1852	1852	-553623498	-78045527	-4780080	-160	703384	5483464	40556040	118601567	52823103	606446601	0xA113FB5A17EF3178
0	233	233	-251657794	261882578	-3259908	4	8661903	11921811	598405809	336523231	107232174	358889968	0x10B583976D0C9236
0	1032	1032	-18036467	-318656	-1618626	-129	25687755	27306381	17704250	18022906	30964475	49000942	0xA2B99CCEE3D8BDAA
0	10	10	-1056570	-1045266	-1568211	188	60514	1628725	426665	1471931	429464	1486034	0x24161325E430E58F
0	2110	2110	-160298299	225964768	-1470963	0	336332698	337803661	783307080	557342312	462053775	622352074	0xF3806AB866D675B6
0	5738344	5738344	-1786884	-2002862	-1111501	-58536	7308981	8420482	106641230	108644092	107676557	109463441	0x174B21F541E8EE1B
0	77638	77638	918103	-2602495	-1060178	0	1040605	2100783	13626641	16229136	47252596	46334493	0x8D3A11BE85E403EE
0	1027	1027	-660475	-1005781	-827656	1	2500845	3328501	24499825	25505606	25502623	26163098	0x7720AAA63CC6F5C2
0	16	16	-2070010	-12602682	-476589	0	8181248	8657837	11218099	23820781	27000301	29070311	0x7F1ADC76D4F3D9CE
0	2	2	-9885624	-2403985	-394093	0	2289156	2683249	2340884	4744869	2869153	12754777	0xA6DA6495969D7CF3
0	24953	24953	40361021	-5037302	-235511	-6339	35677917	35913428	117990782	123028084	323677505	283316484	0x14E5E89315BBE1A4
0	9	9	-3640557	-757363	-235345	0	2478264	2713609	1650313	2407676	1654330	5294887	0xDDA4A52435265197
0	18729	18729	24220434	-1652632	-203289	-5325	8204526	8407815	50045378	51698010	109420843	85200409	0x0A45CBD850200611
0	43404	43404	5963941	1332979	-169461	-5492	1092898	1262359	22851019	21518040	28891675	22927734	0x89F23668C40B40AB
0	180051	180051	61581	25652	-109752	587	989575	1099327	32235129	32209477	32334203	32272622	0x5C87FDC013E447DF

Community Spirit



RE: WorkloadTools stuck after replaying for 2.4 hours

G gianluca.sartori@sqlconsulting.it
An Guth, Martin, 3C Deutschland GmbH

Antworten | Allen antworten | Weiterleiten | ...

Do 30.04.2020 18:24

Sie haben am 30.04.2020 18:48 auf diese Nachricht geantwortet.

Hi Martin,

I'm sorry that you're having trouble with WorkloadTools.
Unfortunately, without any error message I cannot tell what is happening in the replay. Do you have any error messages to share from your logs?

I'm sorry for no sequence_number getting collected. Turns out this error applies only to XE streaming, so I added this property and now it should be collected correctly in 1.5.7.
For the time being you could update event_sequence with row_id and you should be fine.

In the past I used WT to capture and replay for days or even weeks, so I have no idea what is going wrong in your case. I have to say that in those case I used it with the realtime replay and I have never captured a 6.32 GB sqlite file. However, it should not make a difference.
Synchronization mode again should not be the source of the problem.

I just relased v 1.5.7 and I hope it addresses some of your points.
Good luck with your project!

Cheers
Gianluca

From: Guth, Martin, 3C Deutschland GmbH
Sent: mercoledì 29 aprile 2020 17:59
To: gianluca.sartori@sqlconsulting.it
Subject: WorkloadTools stuck after replaying for 2.4 hours

Hi Gianluca,

I was quite motivated after seeing a 30 minutes replay finish in 30 minutes and therefore tried a longer replay.
I recorded a workload for 28 hours and started the replay.
However after approx.. 2.4 hours (9%) into the replay it suddenly stucks.
Sp_whoIsActive shows no activity and the console doesn't show any additional messages for hours....to me it seems that the replay just died.

Amazing: Gianluca ships a new release in just one day

Photo by Bill Oxford on Unsplash

DEMOS

Photo by John Schnobrich on Unsplash

TIPS & TRICKS

A Logging-Table can come in handy

- Recipe for a logging-table (if you don‘t have one used by your system)
  - Create a new table with at least one DateTime column
  - Add a new record to the table every minute during the capture (e.g. via an agent-job)
- Example:

Results | Messages

	zeitpunkt
1	2020-05-09 11:39:00.290
2	2020-05-09 11:40:00.290
3	2020-05-09 11:41:00.290
4	2020-05-09 11:42:00.290
5	2020-05-09 11:43:00.290

- Especially useful for monitoring of the replay speed

Optimize the Restore

- Point-In-Time Restore using Log-Backups
  - Run a marked transaction right at the beginning of your capture

```
1 BEGIN TRANSACTION CaptureStart WITH MARK 'CaptureStart';
2 SELECT GETDATE();
3 COMMIT TRANSACTION CaptureStart;
```

  - Perform a point-in-time-restore from logs using *WITH STOPATMARK = 'CaptureStart'*
  - Alternatively use WITH STOPAT and the exact start time of the capture
- Leverage database snapshots to speed up subsequent restores after the first restore from backups (available in Standard Edition starting with SQL Server 2016 ☺)

Photo by JESHOOTS.COM on Unsplash

LESSONS LEARNED

Lessons Learned Workload Tests

- Complex Topic → make sure to reserve enough time
- Interesting things to discover in your „own“ databases (e.g. attempts to access tables which have been deleted months ago)
- Occasionally it may make sense to manipulate the workload
- Be aware that all activity on the database(s) is recorded
  - Watch out for backup commands → at least ensure that the replay can‘t overwrite production backups

Lessons Learned Presenting

- Less is more: Question the scope of your presentation
- Demos needed more time as originally thought
- Creating a (repeatable) workload is more challenging as originally thought

Resources

- https://github.com/spaghettidba/WorkloadTools
- https://spaghettidba.com/
- https://sqlitebrowser.org/
- https://www.brentozar.com/archive/2019/04/free-sql-server-load-testing-tools/
  https://www.brentozar.com/archive/2019/01/how-to-load-test-a-database-application/

Photo by Edwin Andrade on Unsplash

QUESTIONS/ DISCUSSION

Photo by Melinda Gimpel on Unsplash

Martin Guth, Heilbronn

@sqlNewRow

martinguth.de (presentation download)

martin_guth@hotmail.com