“If You Graph It, They Will See It”

Identifying Root Issues From Product Testing to Production Crisis.

Splunk@Murex

For Test and Development

Robert Lynch | Global Splunk Manager & Performance Manager

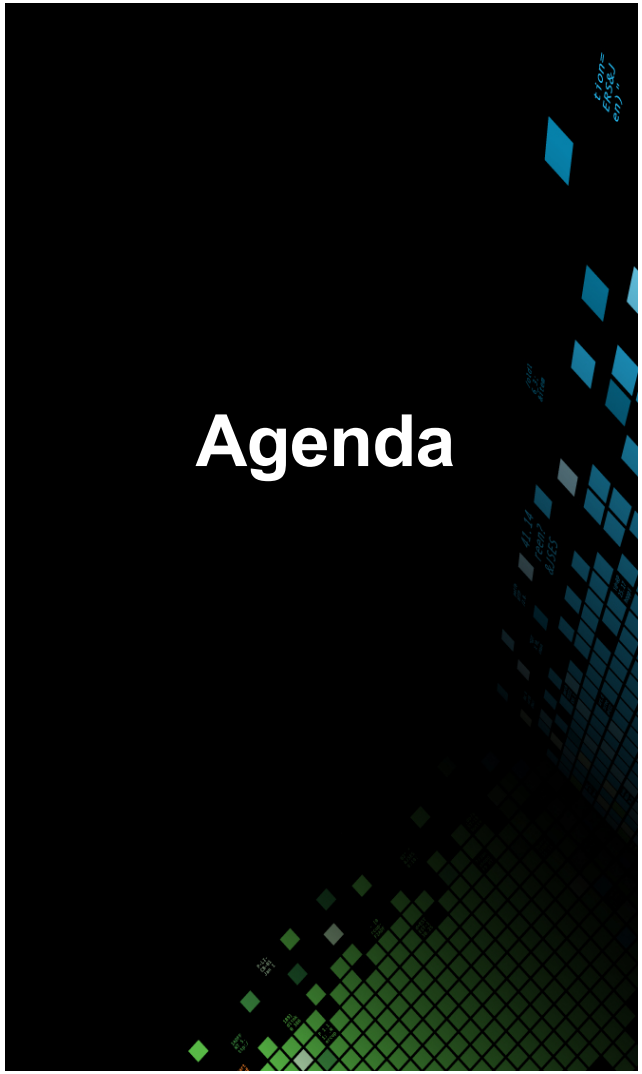
September 2017 | Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf2017



1. Who is Robert Lynch?
2. A little about Murex
3. Murex in-house solution had limitations!
4. How Murex uses **Splunk** for test and development
5. Some Use Cases:
   - The trillion dollar problem
   - Millisecond analysis
   - So, where is my test?

splunk> .conf2017

Who Am I ?

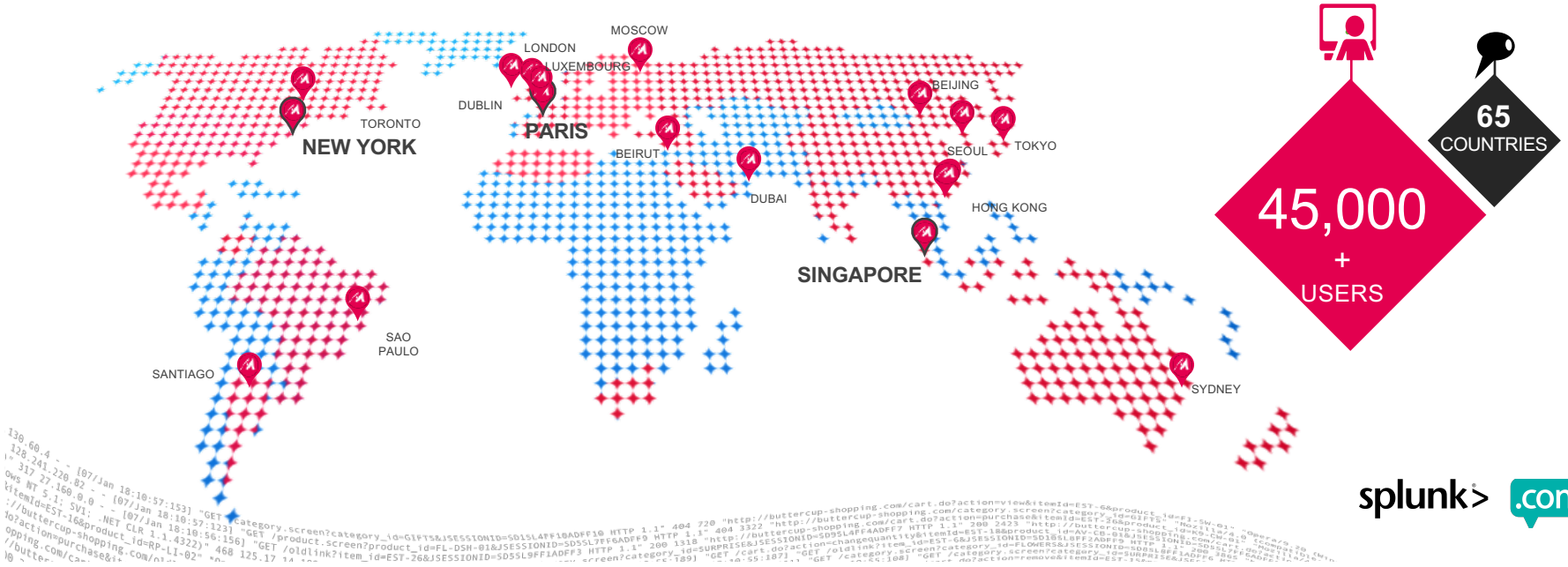
- Name:
  - Robert Lynch
- Current Position:
  - Global Splunk Manager @ Murex
  - Non-Functional Test Manager @ Murex
- History:
  - 13 years working at Murex in Non-Functional Testing
  - Masters “High Performance Computing”
  - Introduced Splunk to Murex 3 Years ago

Murex: Leading Vendor in Financial Software

- Murex is a leading vendor in the provision of integrated solutions for front office, back office and risk management for the capital markets industry
  - 30 years dedicated to our domain and clients
  - Around 45,000 users and 250+ clients in 65 countries
  - 3 Regional Hubs , 17 offices covering all time zones
  - Staff of 2,000, over 60 nationalities
  - Revenue of USD 0.5bn in 2016
  - Over USD 1 billion invested in R&D over the last 10 years
  - #1 Overall Vendor: Risk Magazine Technology Awards 2016
  - #7 in Chartis RiskTech100 for 2017 (highest ranked non-US headquartered company)

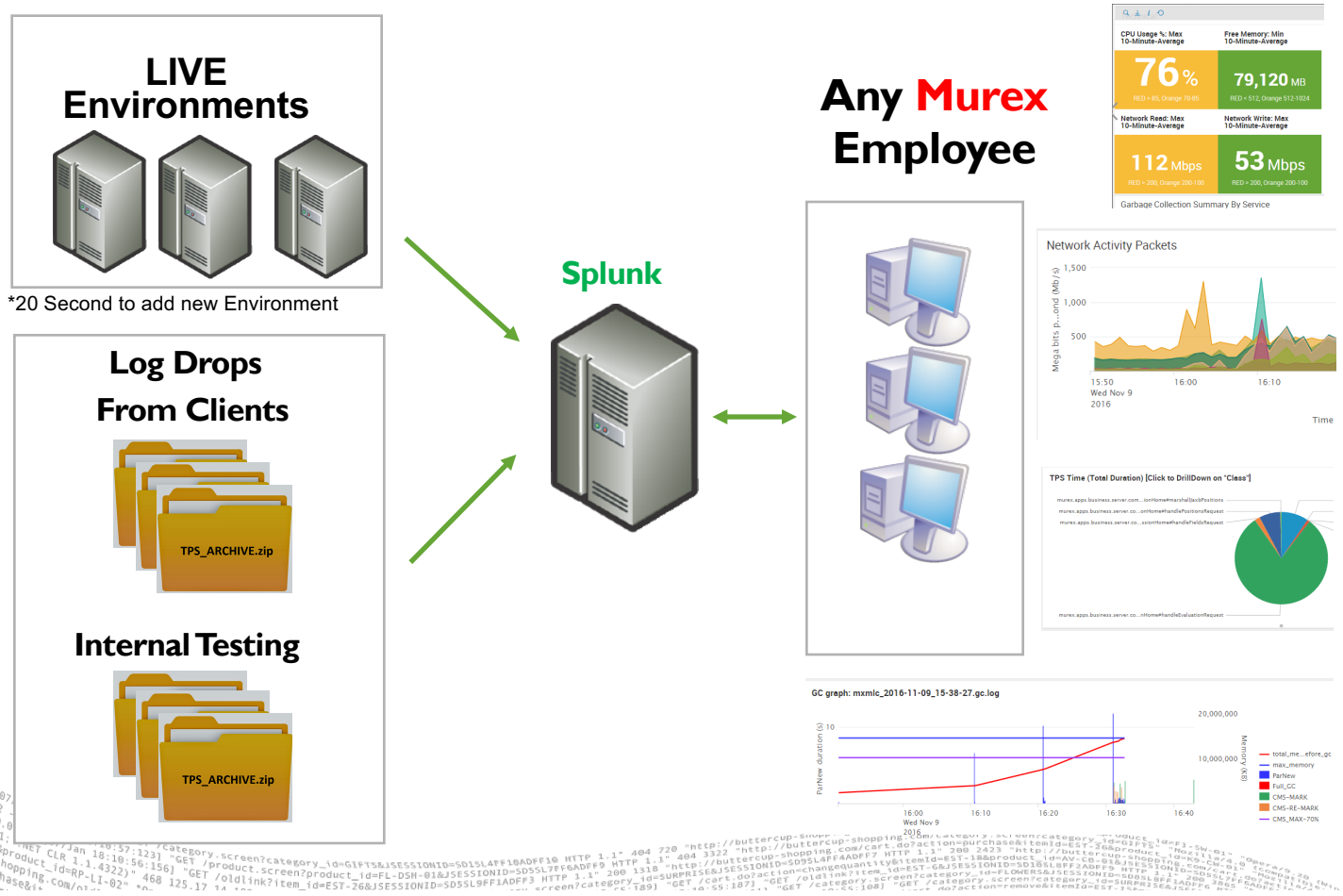
A Worldwide Presence

3 REGIONAL HUBS, 17 OFFICES COVERING ALL TIME ZONES



splunk> .conf2017

Splunk Deployment At Murex

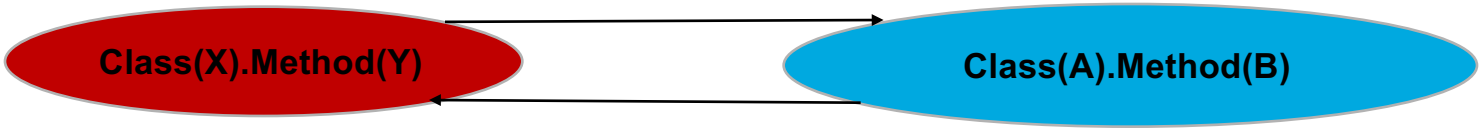


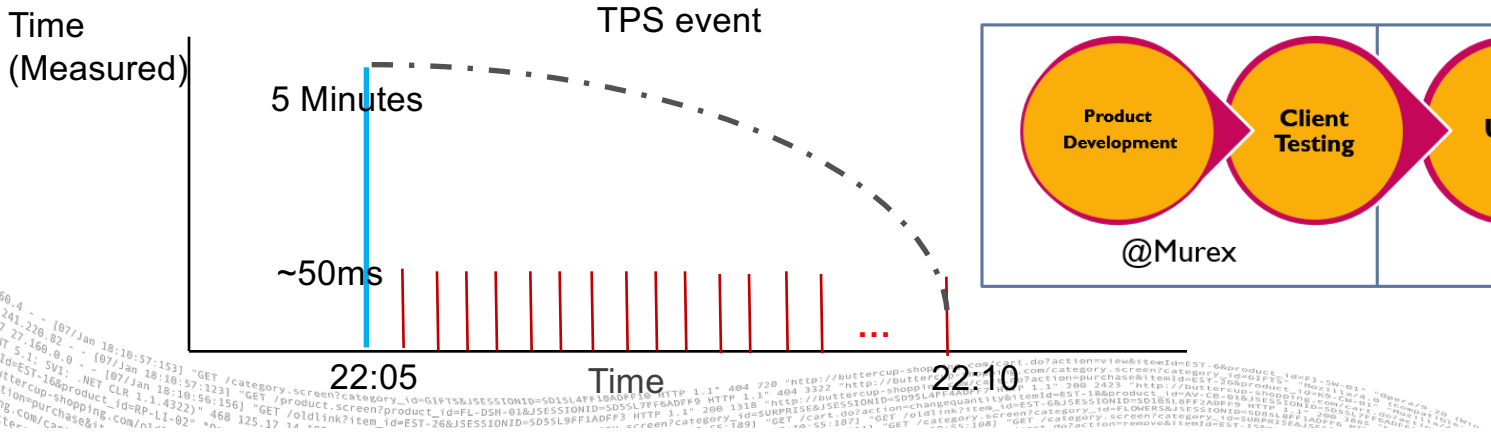
splunk> .conf2017

What is TPS tracing?

- TPS is light performance tracing across any JAVA service
- Code level timing on the Class.Method
  - Class(X).Method(Y) does action
  - It calls Class(A).Method(B) 6000 times

	Count	Average	Total
Class(X). Method(Y)	1	5 Minutes	5 Minutes
Class(A). Method(B)	6000	50 ms	5 Minutes

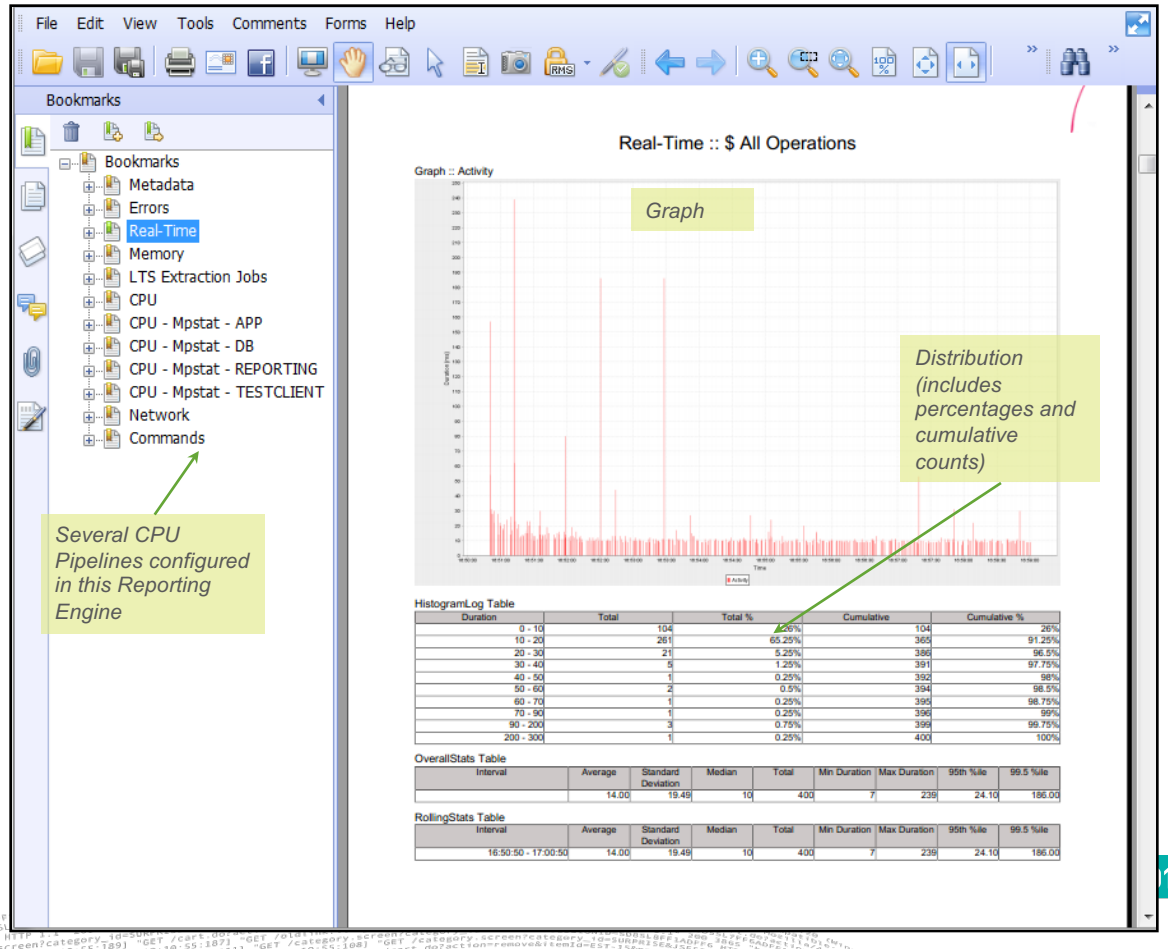


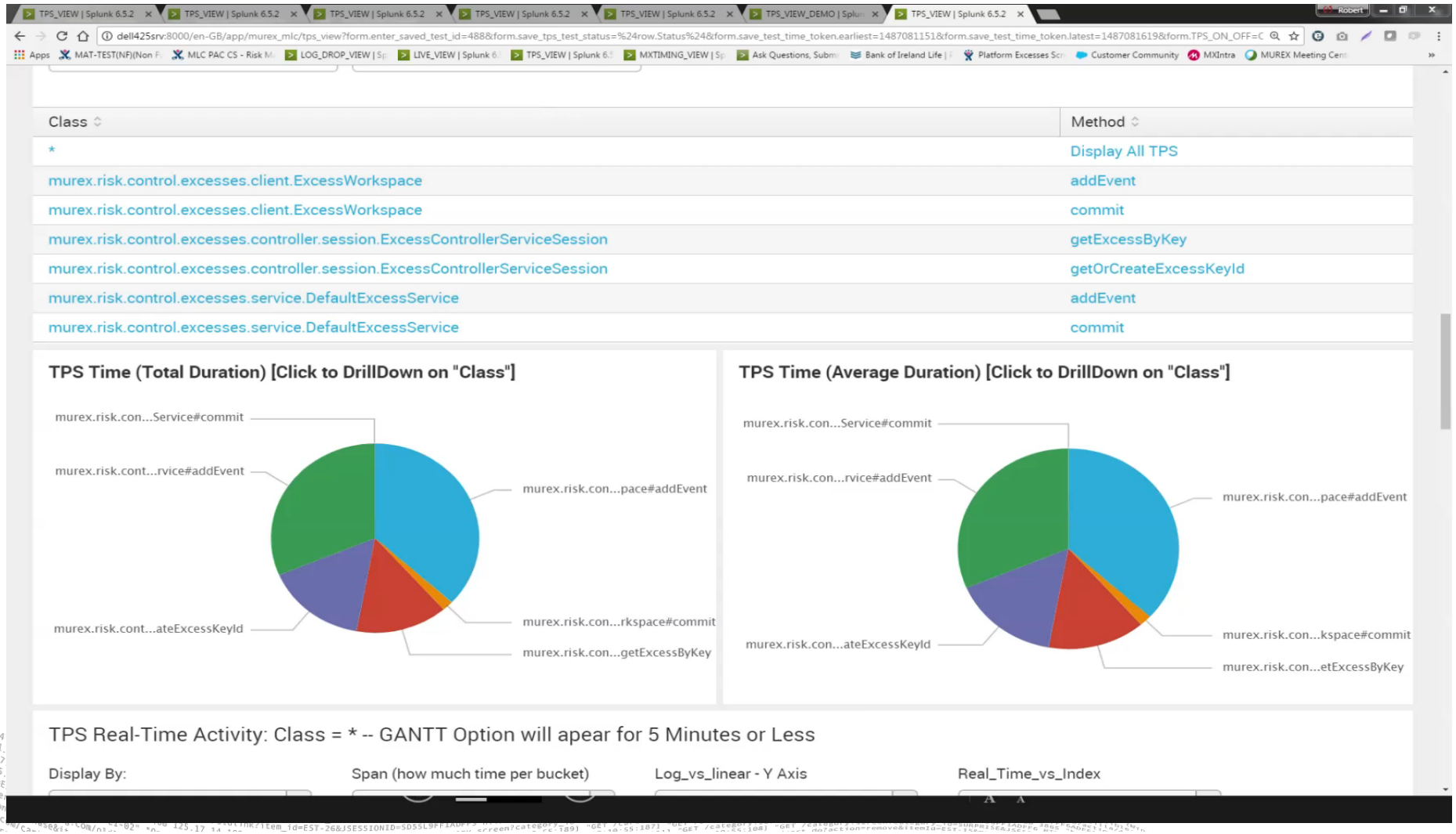


splunk> .conf2017

Murex in-house solution had limitations!

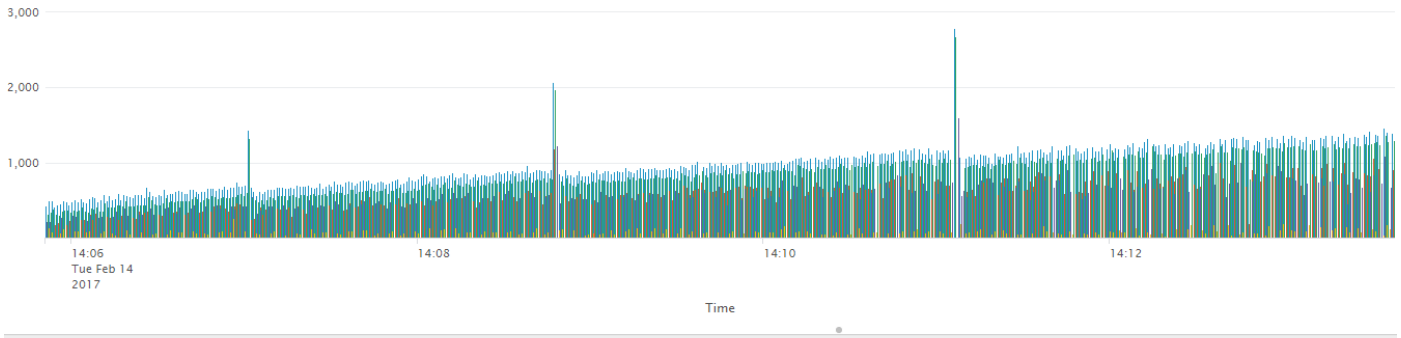
- TPS Report was a PDF = static report
- The existing PDF report was slow to graph millions of lines
- We could not zoom into areas of investigation
- Experts were always needed for analysis
- Introducing the new TPS_VIEW in Splunk




TPS_VIEW | Splunk 6.5.2
TPS_VIEW_DEMO | Splu
dell425srv:8000/en-GB/app/murex_mlc/tps_view?form.enter_saved_test_id=488&form.save_tps_test_status=%24row.Status%24&form.save_test_time_token.earliest=1487081151&form.save_test_time_token.latest=1487081619&form.TPS_ON_OFF=C
Apps
MAT-TEST(NF)(Non F
MLC PAC CS - Risk M
LOG_DROP_VIEW | Sp
LIVE_VIEW | Splunk 6.
TPS_VIEW | Splunk 6.5
MXTIMING_VIEW | Sp
Ask Questions, Subm
Bank of Ireland Life |
Platform Excesses Scr
Customer Community
MXIntra
MUREX Meeting Cent
Class
Method
*
Display All TPS
murex.risk.control.excesses.client.ExcessWorkspace
addEvent
murex.risk.control.excesses.client.ExcessWorkspace
commit
murex.risk.control.excesses.controller.session.ExcessControllerServiceSession
getExcessByKey
murex.risk.control.excesses.controller.session.ExcessControllerServiceSession
getOrCreateExcessKeyId
murex.risk.control.excesses.service.DefaultExcessService
addEvent
murex.risk.control.excesses.service.DefaultExcessService
commit
TPS Time (Total Duration) [Click to DrillDown on "Class"]
murex.risk.con...Service#commit
murex.risk.cont...rvice#addEvent
murex.risk.con...pace#addEvent
murex.risk.cont...ateExcessKeyId
murex.risk.con...rkspace#commit
murex.risk.con...getExcessByKey
TPS Time (Average Duration) [Click to DrillDown on "Class"]
murex.risk.con...Service#commit
murex.risk.con...rvice#addEvent
murex.risk.con...pace#addEvent
murex.risk.con...ateExcessKeyId
murex.risk.con...kspace#commit
murex.risk.con...etExcessByKey
TPS Real-Time Activity: Class = * -- GANTT Option will apear for 5 Minutes or Less
Display By:
Span (how much time per bucket)
Log_vs_linear - Y Axis
Real_Time_vs_Index

“If You Graph It, They Will See It”

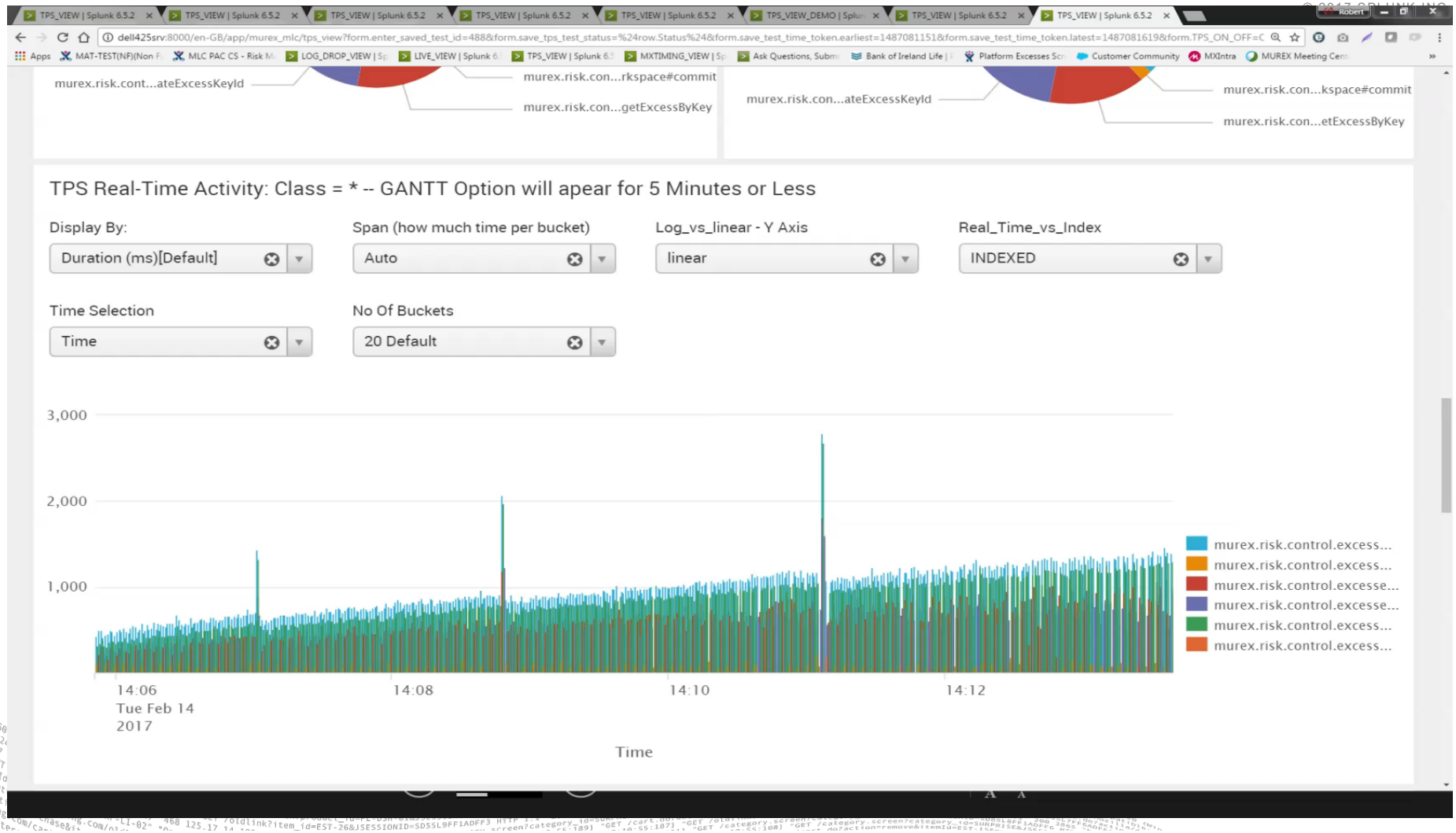
- When you graph an issue, it becomes more obvious
- Sometimes “Average” and “Mediums” and “Maximums” are not enough to see issues…
- In the below graph an obvious increase in the “class.method” over time can be seen
- This might not have been obvious without a visualization



TPS Real-Time Stats TPS_Class = * [For the time-period selected in the chart above]

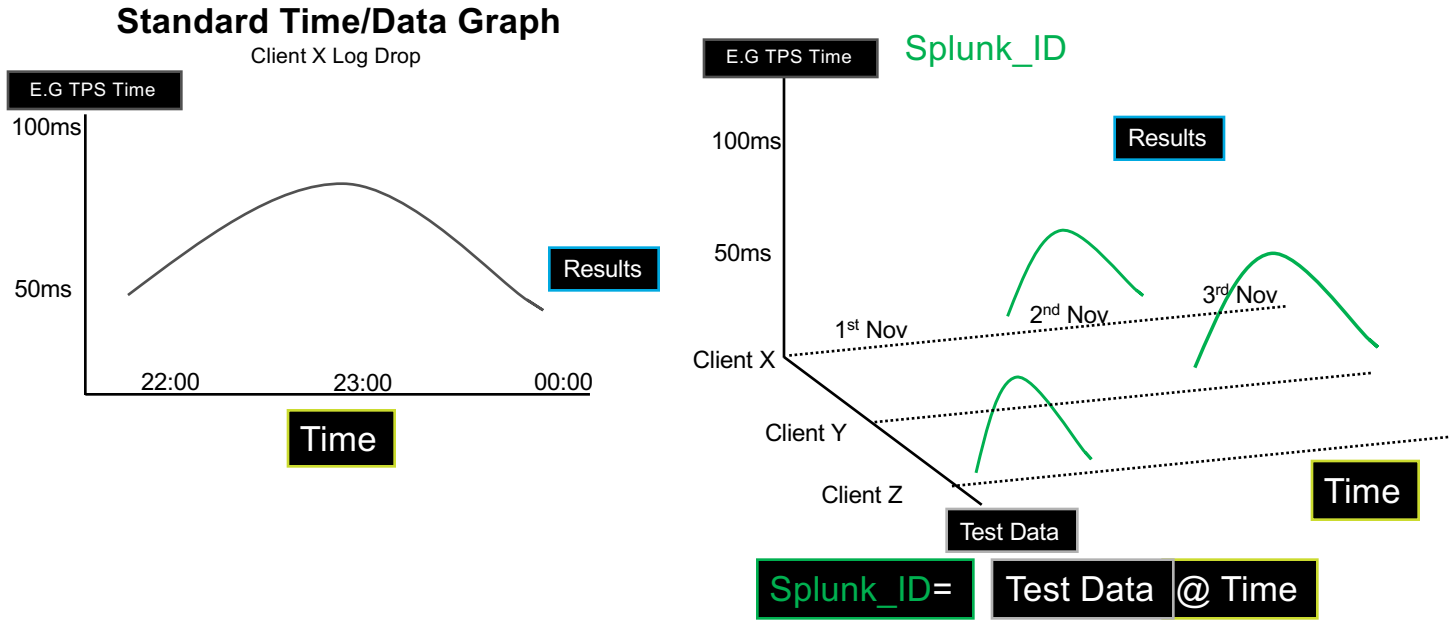
fullyQualifiedMethod	average	count	stdev	median	perc95	perc99.5	min
murex.risk.control.excesses.client.ExcessWorkspace#addEvent	810.30	15330	268.3	789	1230	1422	223
murex.risk.control.excesses.client.ExcessWorkspace#commit	38.80	15301	21.1	34	80	128	14
murex.risk.control.excesses.controller.session.ExcessControllerServiceSession#getExcessByKey	314.60	15330	181.9	288	630	878	34
murex.risk.control.excesses.controller.session.ExcessControllerServiceSession#getOrCreateExcessKeyId	346.10	15325	152.4	325	613	886	62
murex.risk.control.excesses.service.DefaultExcessService#addEvent	689.60	15330	268.5	670	1111	1297	162
murex.risk.control.excesses.service.DefaultExcessService#commit	0.10	15301	0.6	0	1	1	0

splunk> .conf2017


TPS_VIEW | Splunk 6.5.2
dell425srv:8000/en-GB/app/murex_mlc/tps_view?form.enter_saved_test_id=488&form.save_tps_test_status=%24row.Status%24&form.save_test_time_token.earliest=1487081151&form.save_test_time_token.latest=1487081619&form.TPS_ON_OFF=C
Apps
MAT-TEST(NF)(Non F
MLC PAC CS - Risk M
LOG_DROP_VIEW | Sp
LIVE_VIEW | Splunk 6.
TPS_VIEW | Splunk 6.5
MXTIMING_VIEW | Sp
Ask Questions, Subm
Bank of Ireland Life |
Platform Excesses Scr
Customer Community
MXIntra
MUREX Meeting Cent
murex.risk.cont...ateExcessKeyId
murex.risk.con...rkspace#commit
murex.risk.con...getExcessByKey
murex.risk.con...ateExcessKeyId
murex.risk.con...kspace#commit
murex.risk.con...etExcessByKey
TPS Real-Time Activity: Class = * -- GANTT Option will apear for 5 Minutes or Less
Display By:
Duration (ms)[Default]
Span (how much time per bucket)
Auto
Log_vs_linear - Y Axis
linear
Real_Time_vs_Index
INDEXED
Time Selection
Time
No Of Buckets
20 Default
3,000
2,000
1,000
14:06
Tue Feb 14
2017
14:08
14:10
14:12
Time
murex.risk.control.excess...
murex.risk.control.excess...
murex.risk.control.excesse...
murex.risk.control.excesse...
murex.risk.control.excess...
murex.risk.control.excess...

What is a Splunk_ID?

- Splunk_ID is a reference to your data in Splunk
- A numeric code that will reference your “Test Data” @ “Time” = Splunk_ID

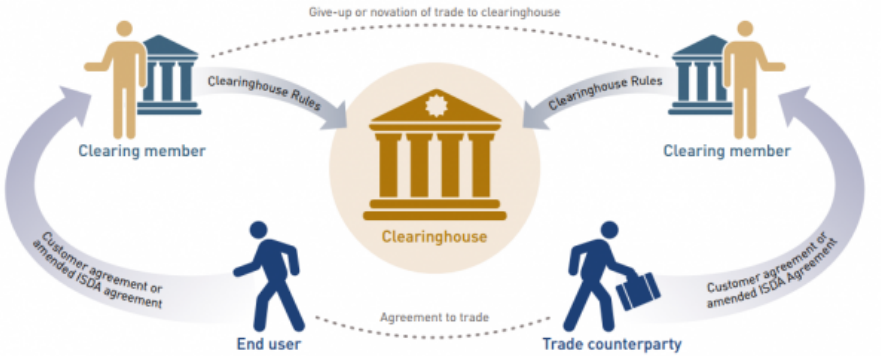


- This means we can create URL’s to a specific “Time” in a set of “Test Data”
- This is used by developers/testers to quickly share data with one another

splunk> .conf2017

Use Case 1: “The One Trillion Dollar Problem ”

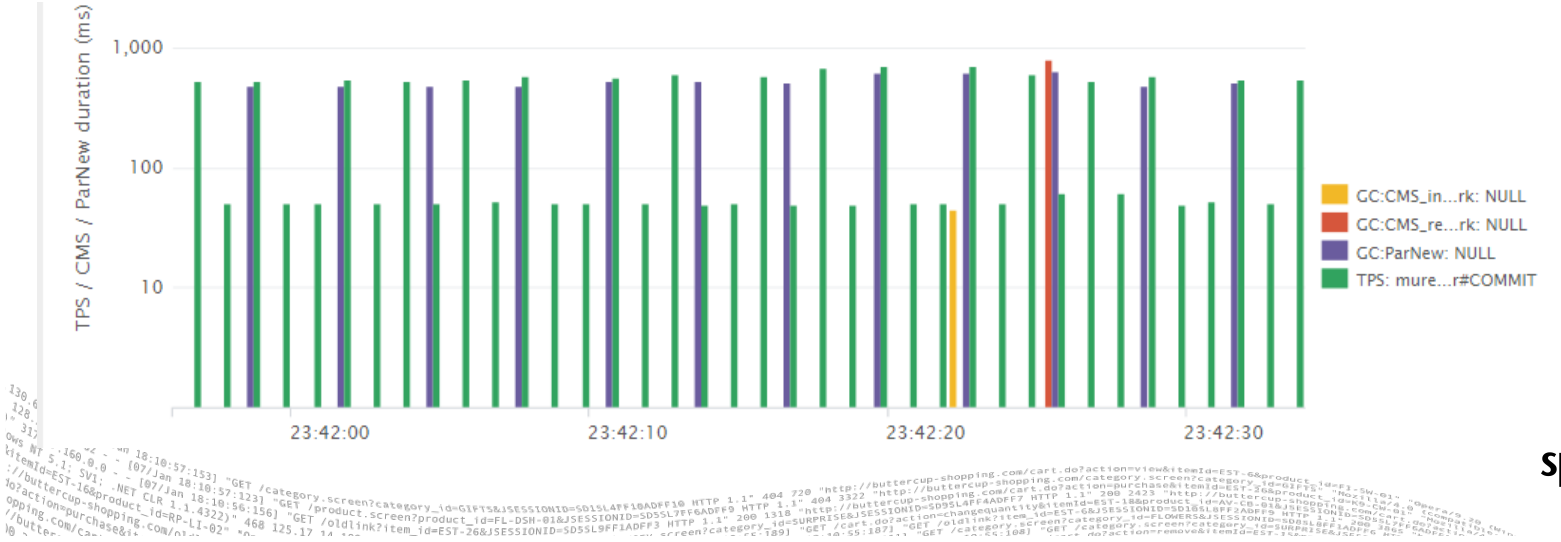
- What is a “Central Counterparty Clearing House” (CCP)?
  - “An organisation that becomes the counterparty to the buyer and the seller of a trade and guarantees the terms of that trade even if one party defaults on the agreement”
  - After 2007-2009 Financial Crash clearing became a regulatory priority
- Murex has a large CCP client in Europe were it needed to perform 60 position updates per second as part of its clearing and risk process
  - 17 Milliseconds per update (1000 Milliseconds = 1 Second)
- This client can clear up to One Trillion Dollars Notional Daily!



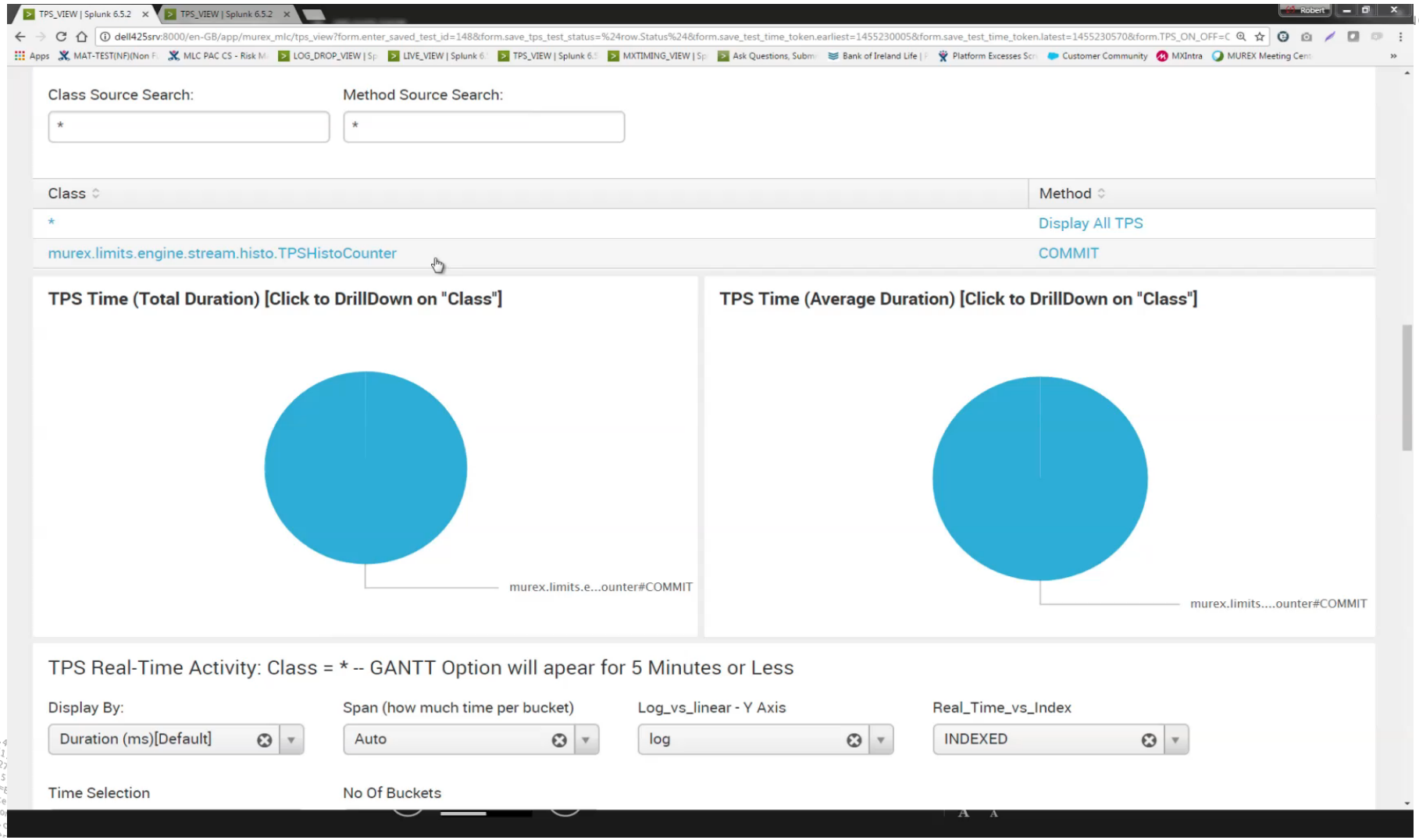
splunk> .conf2017

Use Case 1: Overlapping Different Data Sets

- By using Splunk we were able to overlap different data sets
  - TPS Tracing with Garbage Collection(GC) [GC is native to Java to clean memory, however if not handled correctly it can slow down performance]
- Below we can see the GC Overlapped with TPS
  - The green line is updates Insertion (COMMIT), a lot of times in parallel we can see purple, yellow or red lines, this is “Java GC” slowing down the COMMIT!



splunk> .conf2017


TPS_VIEW | Splunk 6.5.2
TPS_VIEW | Splunk 6.5.2
dell425srv:8000/en-GB/app/murex_mlc/tps_view?form.enter_saved_test_id=148&form.save_tps_test_status=%24row.Status%24&form.save_test_time_token.earliest=1455230005&form.save_test_time_token.latest=1455230570&form.TPS_ON_OFF=C
Class Source Search:
*
Method Source Search:
*
Class
Method
*
Display All TPS
murex.limits.engine.stream.histo.TPSHistoCounter
COMMIT
TPS Time (Total Duration) [Click to DrillDown on "Class"]
murex.limits.e...ounter#COMMIT
TPS Time (Average Duration) [Click to DrillDown on "Class"]
murex.limits....ounter#COMMIT
TPS Real-Time Activity: Class = * -- GANTT Option will apear for 5 Minutes or Less
Display By:
Duration (ms)[Default]
Span (how much time per bucket)
Auto
Log_vs_linear - Y Axis
log
Real_Time_vs_Index
INDEXED
Time Selection
No Of Buckets

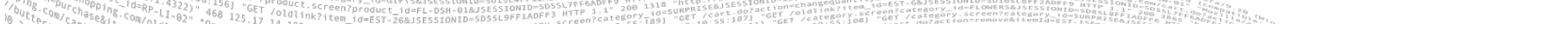
Use Case 1: Before optimizations

- In Splunk we could add up all the GC time divided by the TPS
  - **Par_new_%_of_TPS** = ∑ParNew(GC) / ∑TPS
  - **Par_new_%_of_total_time** = ∑ParNew(GC) / Duration
- Test 1: Before any optimizations
  - The average was 96 milliseconds [That is 10 updates per second, we needed 60!]
  - **Par_new_%_of_TPS = 37% [Major bottle neck identified]**
  - **Par_new_%_of_total_time = 20% [Major bottle neck identified]**

TPS Real-Time Stats TPS_Class = * [For the time-period selected in the chart above]

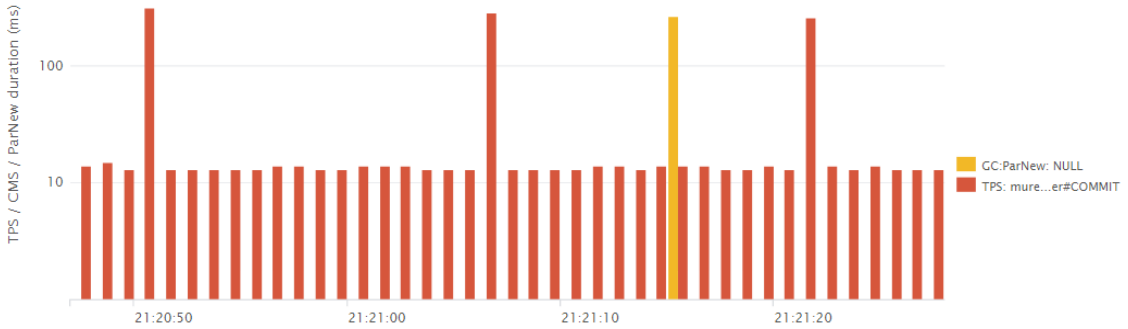
fullyQualifiedMethod	average
murex.limits.engine.stream.histo.TPSHistoCounter#COMMIT	96.30

par_new_%_of_tps	par_new_%_of_total_time
37.75	20.82



Use Case 1: After optimizations

- After multiple binary fixes and garbage collection tuning we hit the 17ms mark
- Some GC peaks we still present, however massive improvement was seen
  - We can see from the graph below that the GC peaks in yellow have reduced and we have also reduced the COMMIT peaks
  - **Par_new_%_of_TPS = 37% Before, 7.8% After**
  - **Par_new_%_of_total_time = 20% Before, 0.89% After**



TPS Real-Time Stats TPS_Class = murex.limits.engine.stream.histo.TPSHistoCounter [For the time-period sel

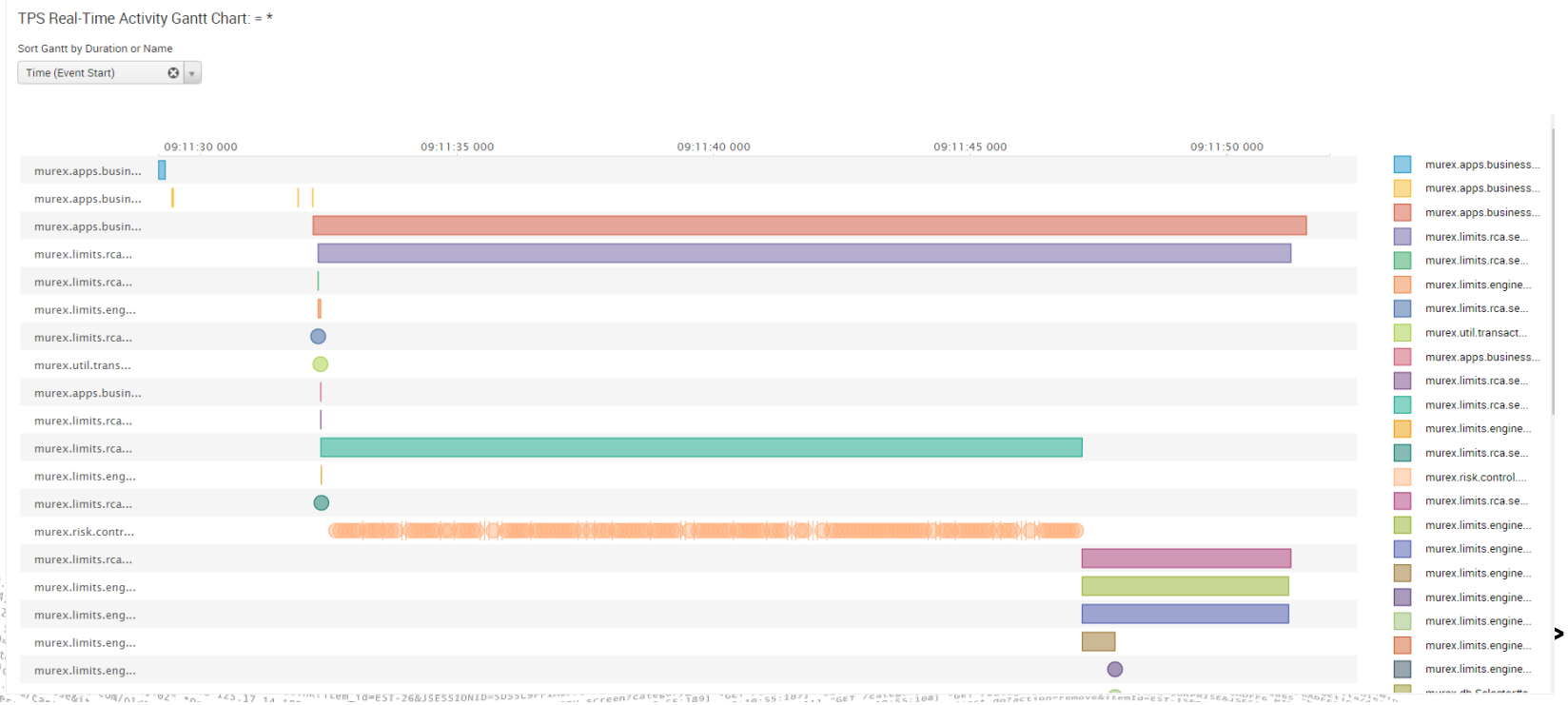
fullyQualifiedMethod	average
murex.limits.engine.stream.histo.TPSHistoCounter#COMMIT	16.50

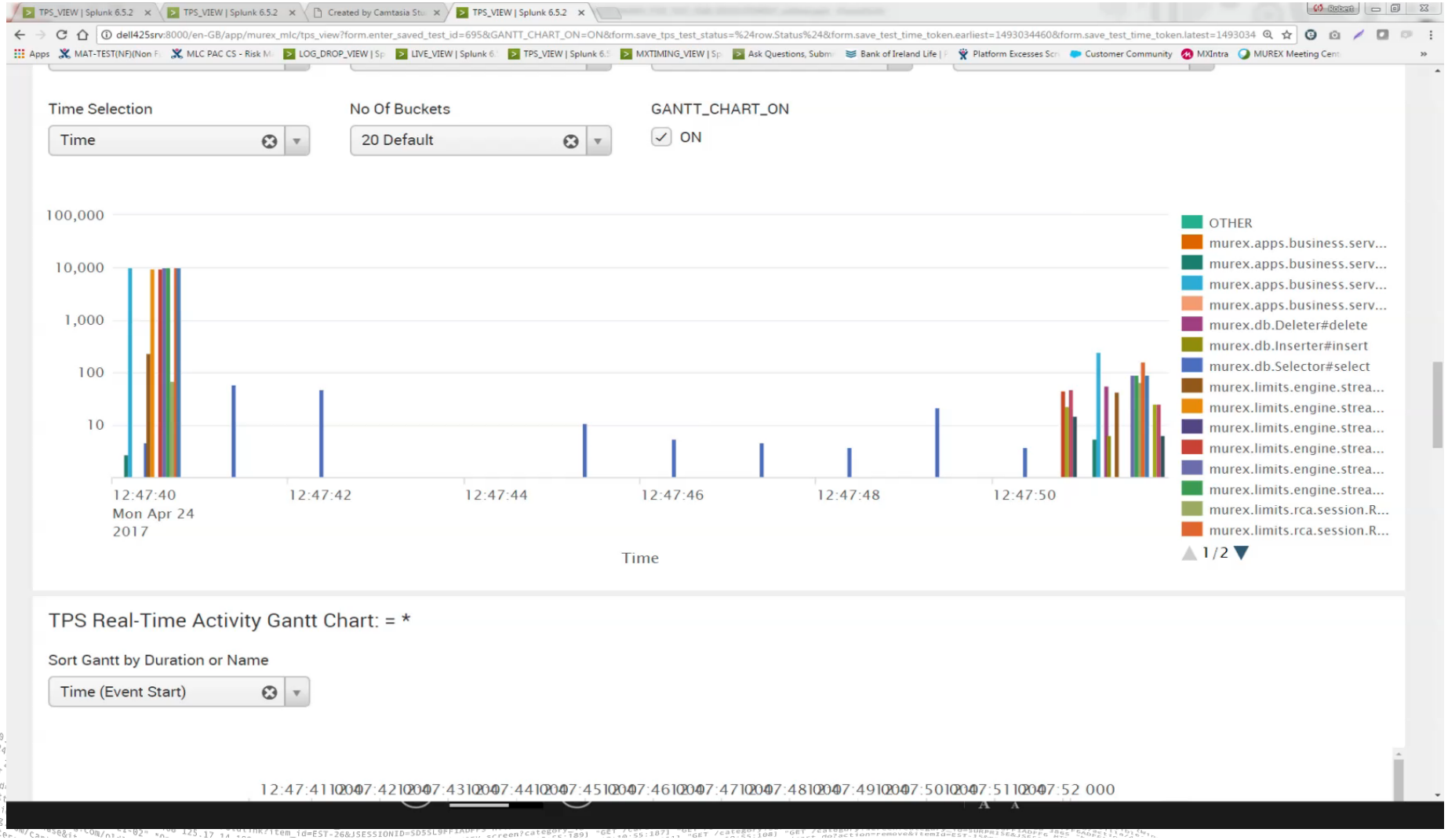
par_new_%_of_tps	par_new_%_of_total_time
7.88	0.89

splunk> .conf2017

Use Case 2 – Millisecond analysis

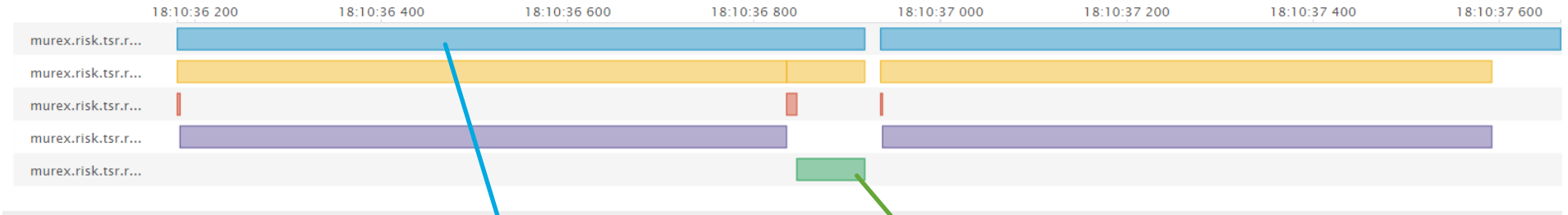
- Viewing the code line in chronological order, helped to improve visualizations
- We can zoom into the Millisecond to see the sequence of method calls




TPS_VIEW | Splunk 6.5.2
Time Selection
Time
No Of Buckets
20 Default
GANTT_CHART_ON
ON
100,000
10,000
1,000
100
10
12:47:40
Mon Apr 24
2017
12:47:42
12:47:44
12:47:46
12:47:48
12:47:50
Time
OTHER
murex.apps.business.serv...
murex.apps.business.serv...
murex.apps.business.serv...
murex.apps.business.serv...
murex.db.Deleter#delete
murex.db.Inserter#insert
murex.db.Selector#select
murex.limits.engine.strea...
murex.limits.engine.strea...
murex.limits.engine.strea...
murex.limits.engine.strea...
murex.limits.engine.strea...
murex.limits.engine.strea...
murex.limits.rca.session.R...
murex.limits.rca.session.R...
1/2
TPS Real-Time Activity Gantt Chart: = *
Sort Gantt by Duration or Name
Time (Event Start)

Use Case 2: Sequential vs. parallel testing

- During product development we have to stress code to make sure it can take Parallel load
- Sequential Import:
  - #addpoint 734ms
  - #storeExisting 73ms

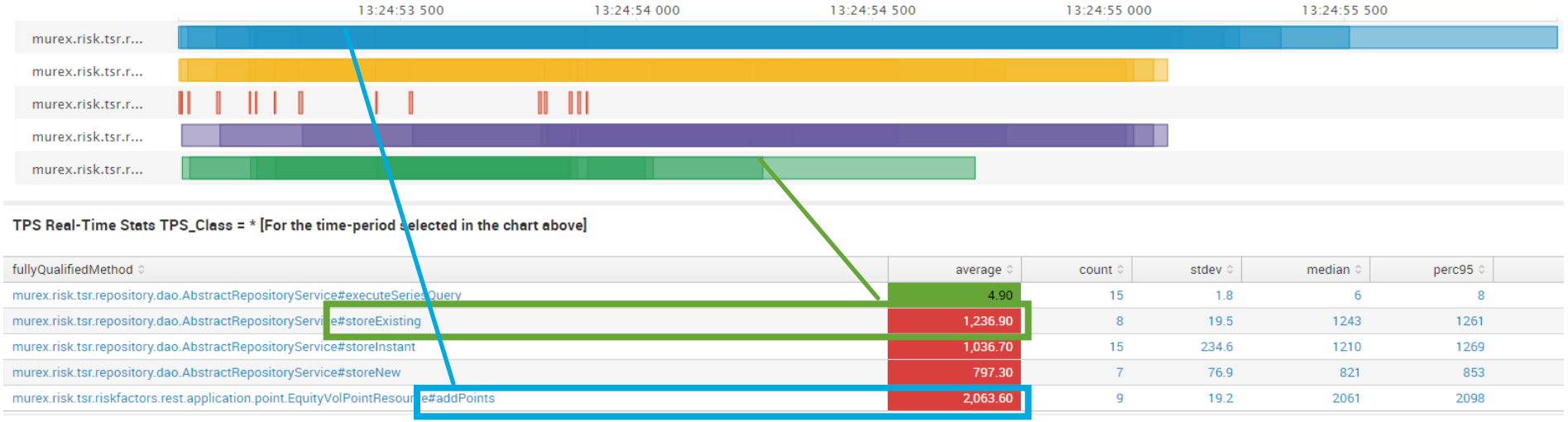


TPS Real-Time Stats TPS_Class = * [For the time-period selected in the chart above]

fullyQualifiedMethod	average	count	stdev	median	perc95
murex.risk.tsr.repository.dao.AbstractRepositoryService#executeSeriesQuery	5.30	3	4.9	3	11
murex.risk.tsr.repository.dao.AbstractRepositoryService#storeExisting	73.00	1	0.0	73	73
murex.risk.tsr.repository.dao.AbstractRepositoryService#storeInstant	464.70	3	329.7	654	656
murex.risk.tsr.repository.dao.AbstractRepositoryService#storeNew	652.50	2	2.1	654	654
murex.risk.tsr.riskfactors.rest.application.point.EquityVolPointResourc#addPoints	734.00	2	5.7	738	738

Use Case 2: Sequential vs. Parallel Testing

- Parallel Import: 15 Imports in Parallel
  - #addpoint Increased from 734ms to 2,063ms => 3 time slower – Not Bad!
  - #storeExisting Increased from 73ms to 1,236ms => 17 Times slower – Possible Issue Found!



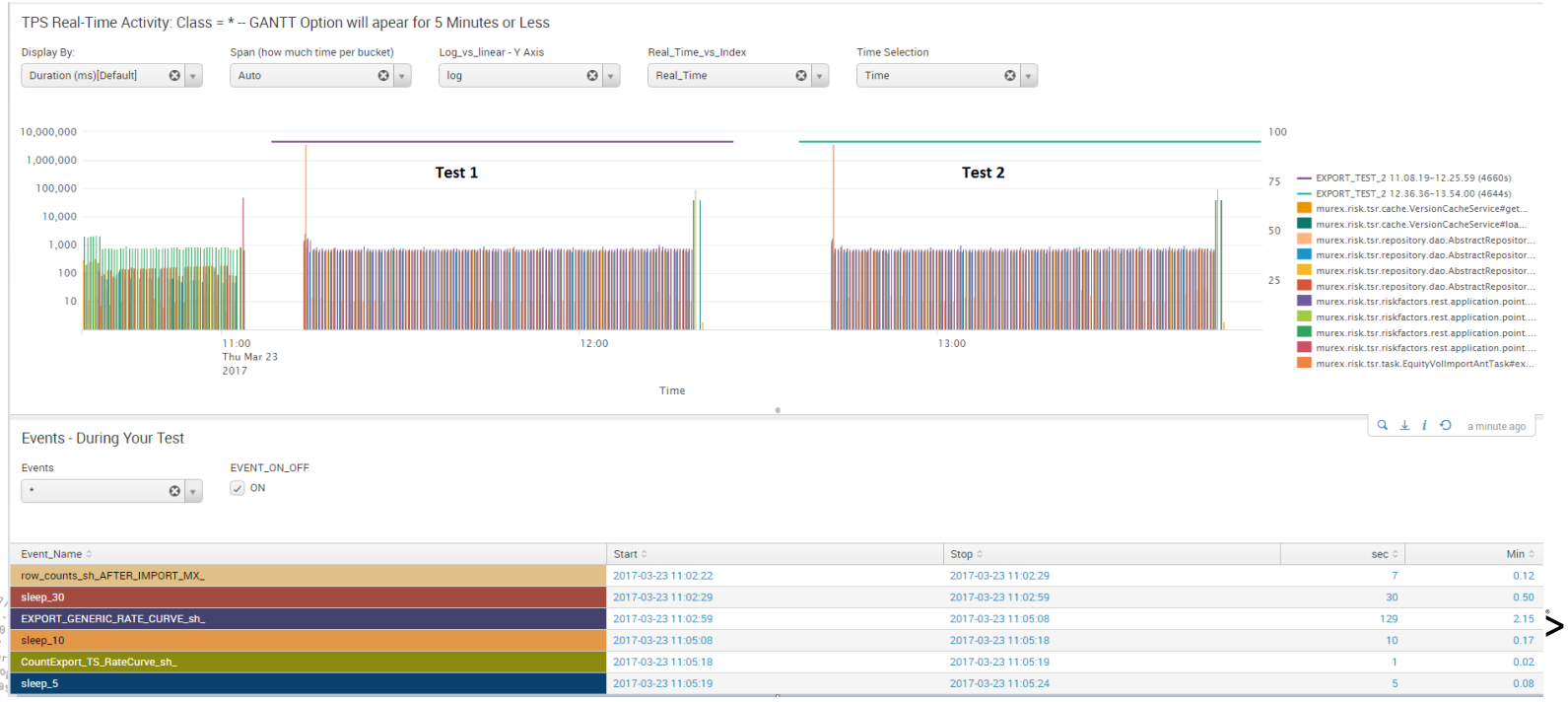
TPS Real-Time Stats TPS_Class = * [For the time-period selected in the chart above]

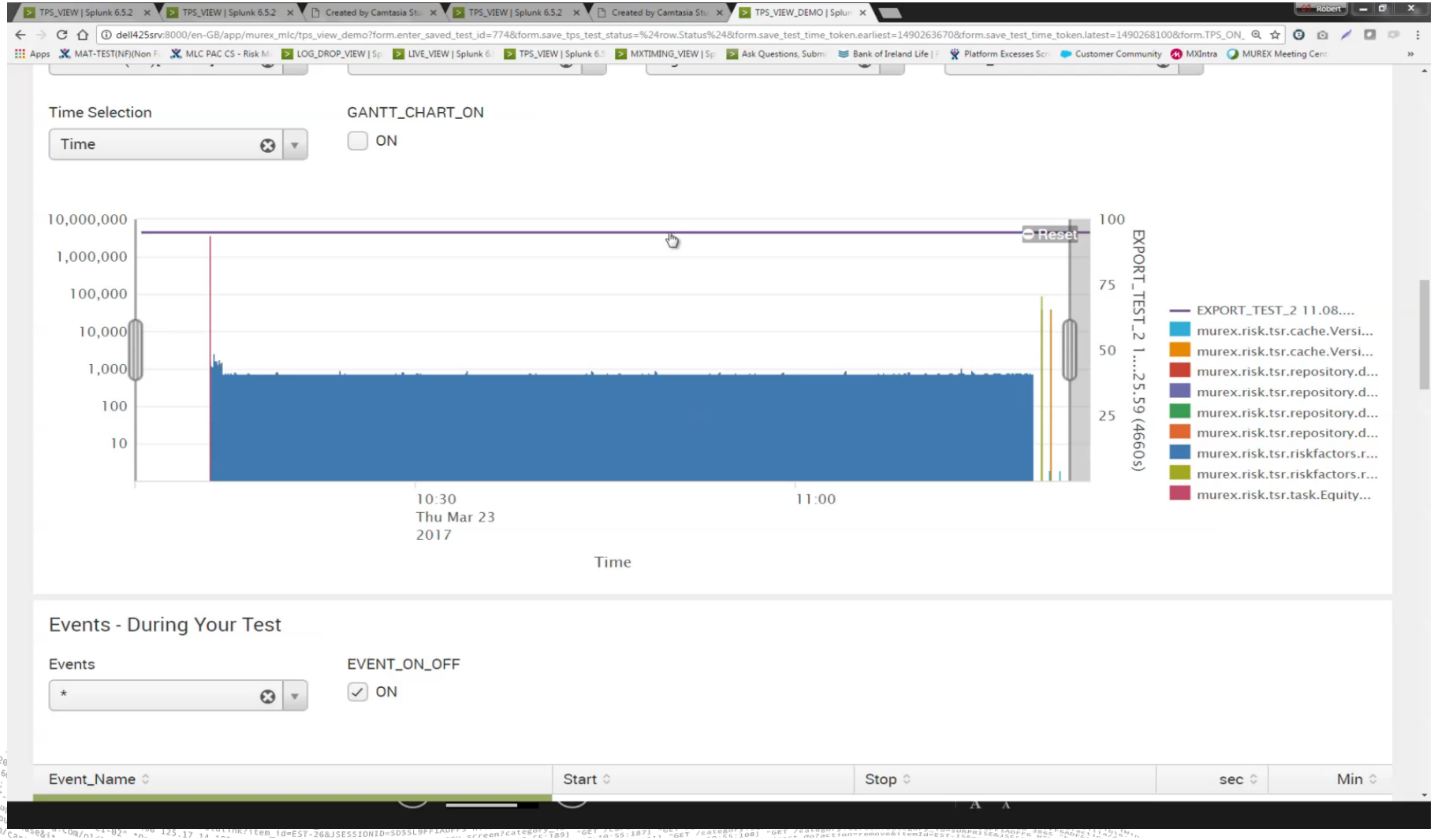
fullyQualifiedMethod	average	count	stdev	median	perc95
murex.risk.tsr.repository.dao.AbstractRepositoryService#executeSeriesQuery	4.90	15	1.8	6	8
murex.risk.tsr.repository.dao.AbstractRepositoryService#storeExisting	1,236.90	8	19.5	1243	1261
murex.risk.tsr.repository.dao.AbstractRepositoryService#storeInstant	1,036.70	15	234.6	1210	1269
murex.risk.tsr.repository.dao.AbstractRepositoryService#storeNew	797.30	7	76.9	821	853
murex.risk.tsr.riskfactors.rest.application.point.EquityVolPointResource#addPoints	2,063.60	9	19.2	2061	2098

splunk> .conf2017

Use Case 3: So Where Is My Test?

- Over a large timeline we need to be able to identify different tests quickly
- Overnight a user might run 10 different tests
- By developing an overlap it becomes obvious where each test is (Test 1 or Test 2)




TPS_VIEW | Splunk 6.5.2
TPS_VIEW | Splunk 6.5.2
Created by Camtasia Stu
TPS_VIEW | Splunk 6.5.2
Created by Camtasia Stu
TPS_VIEW_DEMO | Splun
Robert
dell425srv:8000/en-GB/app/murex_mlc/tps_view_demo?form.enter_saved_test_id=774&form.save_tps_test_status=%24row.Status%24&form.save_test_time_token.earliest=1490263670&form.save_test_time_token.latest=1490268100&form.TPS_ON_
Apps
MAT-TEST(NF)(Non F
MLC PAC CS - Risk M
LOG_DROP_VIEW | Sp
LIVE_VIEW | Splunk 6.
TPS_VIEW | Splunk 6.5
MXTIMING_VIEW | Sp
Ask Questions, Subm
Bank of Ireland Life |
Platform Excesses Scr
Customer Community
MXIntra
MUREX Meeting Cent
Time Selection
Time
GANTT_CHART_ON
ON
10,000,000
1,000,000
100,000
10,000
1,000
100
10
Reset
100
75
50
25
EXPORT_TEST_2 1....25.59 (4660s)
EXPORT_TEST_2 11.08....
murex.risk.tsr.cache.Versi...
murex.risk.tsr.cache.Versi...
murex.risk.tsr.repository.d...
murex.risk.tsr.repository.d...
murex.risk.tsr.repository.d...
murex.risk.tsr.repository.d...
murex.risk.tsr.riskfactors.r...
murex.risk.tsr.riskfactors.r...
murex.risk.tsr.task.Equity...
10:30
Thu Mar 23
2017
11:00
Time
Events - During Your Test
Events
*
EVENT_ON_OFF
ON
Event_Name
Start
Stop
sec
Min

Conclusion: The Splunk "TPS View"

- **Problem**:
  - We needed to view millions upon millions of TPS tracing quickly
- **Problem**:
  - We needed to be able to get data into Splunk quickly
- **Problem**
  - We needed to overlap different data sets to find bottle necks
- **Problem**:
  - How can I show my colleagues quickly what I am looking at?

- **Solution:**
  - How about….100 million lines of data (sliced and diced) in 10 seconds!!
  - New TPS VIEW is hyper accelerated to handle heavy volumes
- **Solution**:
  - LIVE Monitoring: 20 seconds to attach any environment to Splunk (one line command)
  - Log DROP: Upload logs from anywhere (one line command)
- **Solution**:
  - By overlapping TPS tracing with GC, CPU etc.. new bottle necks became obvious:
- **Solution:**
  - Splunk ID: Unique ID to allow you to save a point of investigation and pass it on to you colleagues

splunk> .conf2017


Thanks
MUREX™
Don't forget to rate this session in the .conf2017 mobile app
splunk>
.conf2017

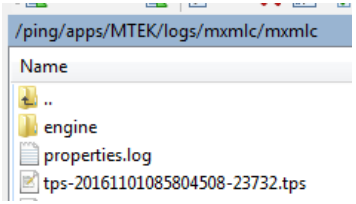
Appendix - Simple TPS Example

- Annotation for basic classes and methods for TPS tracing to produce traces into a file.

```
class Example{
   @Traced
   public void exampleMethod() {
      doWork();
   }
}
```

Output: in “json”

```
{
  "endTime":1473177098093,                    (milliseconds)
  "startTime":1473177098031,
  "operationIdentity":“exampleMethod",
  "name":"Murex.Exmaple“
}
```



_time	FULL_TPS_TRACE
2017-07-18 08:50:11.375	{"endTime":1500360611376,"startTime":1500360611375,"operationIdentity":"publishCacheStatistics","name":"murex.limits.utilities.cache.statistics.CacheStatisticsTimerTask","context":{"parentContext":{"id":-1,"parentContext":null},"data":[{"value":"RTLIM-12472@2015-06-09--Sanity_MaxMaturity_PCE-47--PillarNettingUTable-177891","key":"name"},{"value":"0","key":"hits"},{"value":"0","key":"misses"},{"value":"0","key":"count"},{"value":"4096","key":"maxElements"},{"value":"0","key":"evictions"},{"value":"Default","key":"policy"}],"id":8590023}}
2017-07-18 08:50:11.382	{"endTime":1500360611382,"startTime":1500360611382,"operationIdentity":"publishCacheStatistics","name":"murex.limits.utilities.cache.statistics.CacheStatisticsTimerTask","context":{"parentContext":{"id":-1,"parentContext":null},"data":[{"value":"GlobalRuleCache:Sanity_MaxTenor_ISSp:ValidationRuleRepository","key":"name"},{"value":"0","key":"hits"},{"value":"0","key":"misses"},{"value":"0","key":"count"},{"value":"0","key":"maxElements"},{"value":"0","key":"evictions"},{"value":"Default","key":"policy"}],"id":8590168}}

Appendix – SPLUNK ID

- A SPLUNK ID is stored in a Lookup Table, there are 4 main parts
  1. ID = Unique ID numeric value
  2. Host = What data set are you looking at
  3. Start_Time = Start time of the time slice
  4. Stop_Time = Stop time of the time slice
- Additional Information can be stored and displayed if needed

New Search

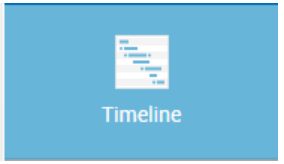
```
| inputlookup Saved_Tests.csv | where Host="EW_COMP" | where 1=1 | search Dev_Optimization="*" | search Functional_Optimization="*" | eval
   Start_epoc=Start | eval Stop_epoc=Stop | convert ctime(Start) | convert ctime(Stop) | table ID, Comment, Dev_Optimization,
   Functional_Optimization, Status, Start, Stop, Version, MX_Build_ID , Start_epoc , Stop_epoc | head 1001 | sort - by ID
```

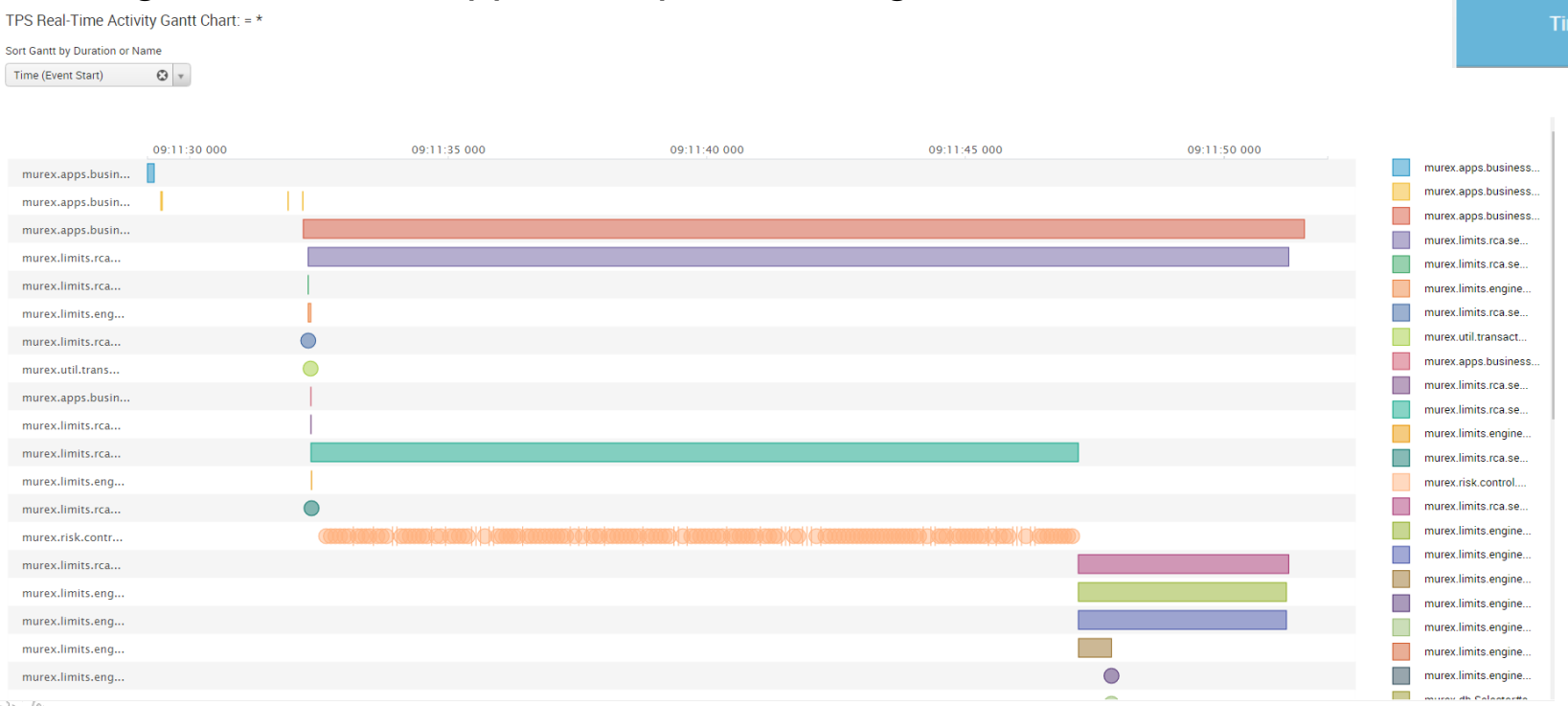
Currently viewing : SPLUNK ID = " 488 " : LINK to Test = http://dell425srv:8000/en-GB/app/murex_mlc/tps_view?&form.enter_saved_test_id=488

ID	Comment	Dev_Optimization	Functional_Optimization	Status	Start	Stop
777	SOFT_TEST_V5	-	-	GOLD	02/14/2017 15:11:01	02/14/2017 15:11:14
776	SOFT_TEST4	-	INDEX_REMOVED	GOLD	02/14/2017 15:11:02	02/14/2017 15:11:12
775	SOFT_TEST_3	-	NO_INDEX	$row.Status$	02/14/2017 15:10:58	02/14/2017 15:11:14

Appendix - Charts

- Using the TimeLine app it was possible to get the below visualizations





splunk> .conf2017

Disclaimer

splunk> .conf2017