

ORACLE® Aconex

EVENT TRACING @ ACONEX

INTRODUCTION

- Principal Architect @ Oracle - Construction & Engineering GBU (CEGBU)
- An Engineer who's had his coding license taken away?
- Draw boxes on whiteboards & wave my hands about?
- PCP user & fan for ~14 years



OUTLINE

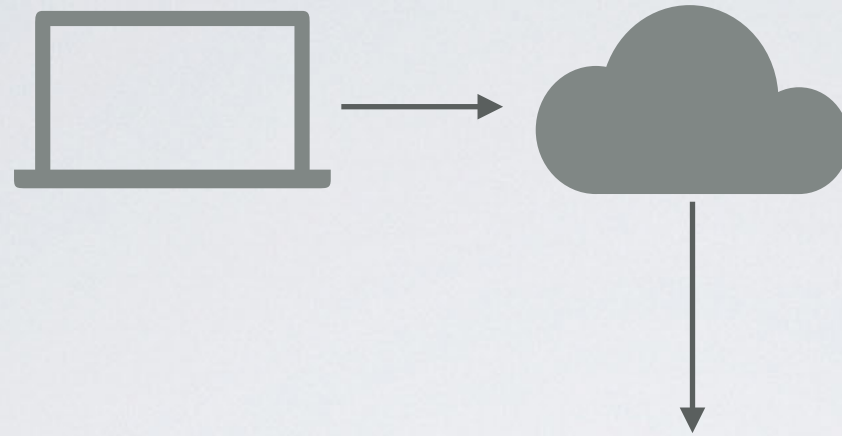
- Key Problems we faced
- How PCP & Parfait help us
- Where PCP & Parfait fit into the broader monitoring suite

ASSUMPTIONS

- Have an understanding of PCP
- Some understanding of Java

ACONEX TECH STACK





Apache
httpd

Application

Other Stuff

DB

Indexer

Other Stuff

TERMINOLOGY



Project

AECOM



Organization



User

CONTROLLER

- Servlet Main-entry point for requests
- Groups related *Actions* together
- Often linked to a particular screen within our application
- e.g. Document Search page -
/SearchDocuments



ACTION

- Specific user-initiated request from a particular screen
- e.g. `zippedDownload`
- Some actions are quick & cheap... others not so much...



PROBLEMS

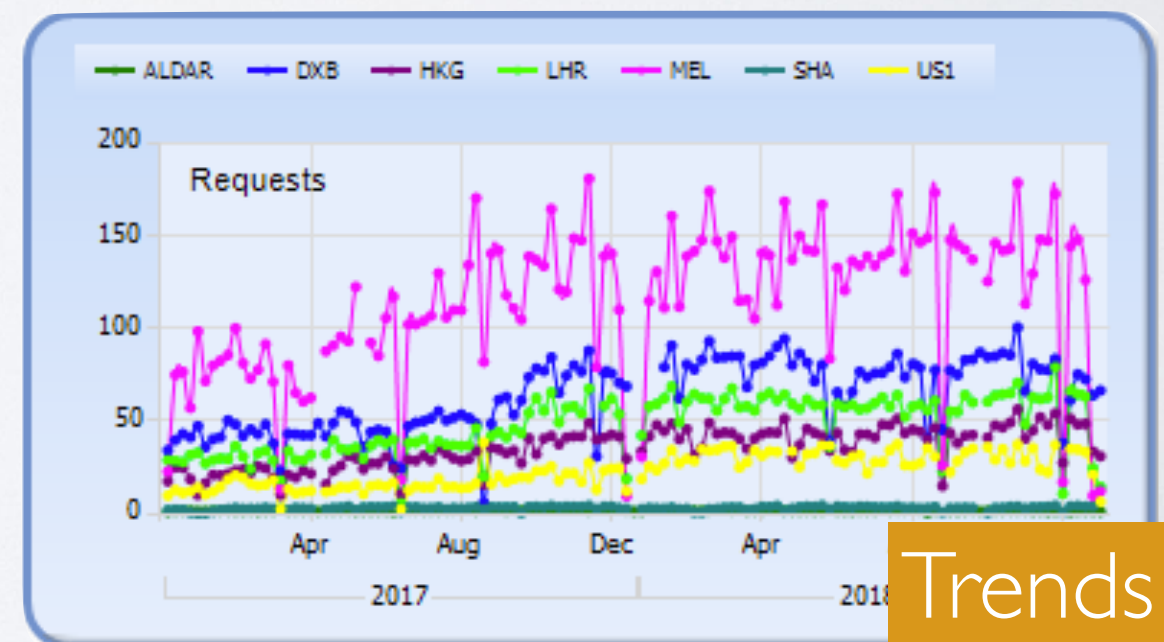
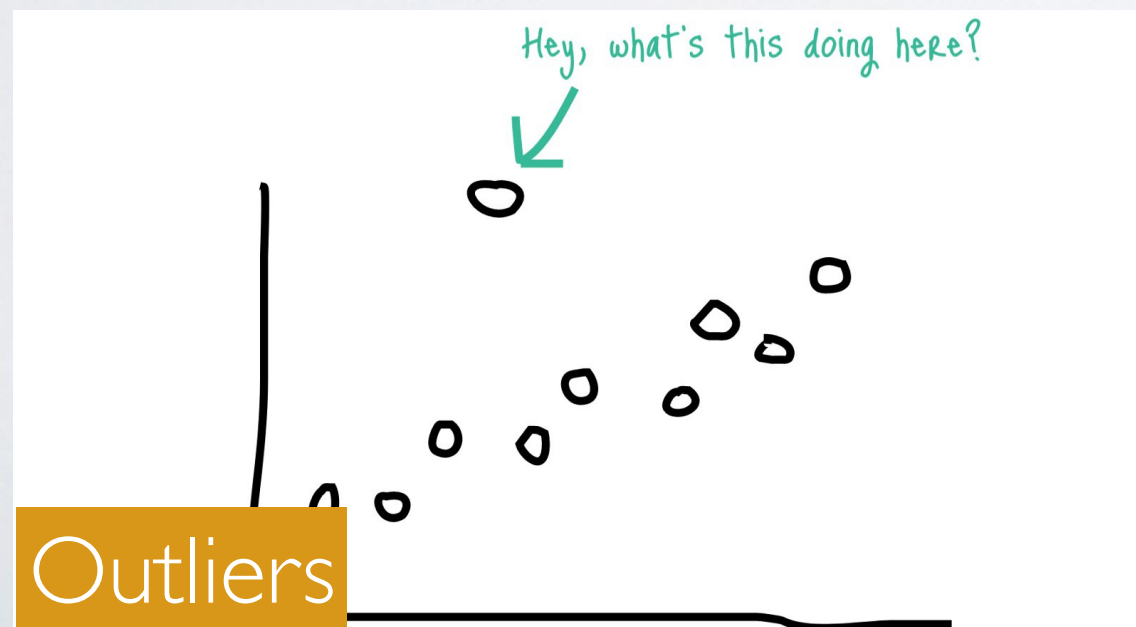


Averages



Outliers

WHAT DID WE TRY TO ACHIEVE?



SO HOW DO WE DO IT?

Code

PER-CONTROLLER METRICS

- Each Controller is an Instance Domain string
- Resource-specific Counters
 - Elapsed Time
 - CPU
 - Database
 - Heap
 - Service Calls


```
aconex.controllers.utime["mel/SearchDocuments"]  
aconex.controllers.db.count["mel/SearchDocuments"]
```

...



Users

& other Segmentations

POPULATION SEGMENTATION

- Users
- Projects
- Threads & Thread Pools
- IP Addresses

LOG4J

- Thread name
- MDC - Mapped Diagnostic Context
- Each Request - UserId, ProjectId is populated


```
[2019-02-27 11:49:34,790 INFO ][it.timing.LoggerSink]  
[tid=http-general-246936,  
rid=XHXenlcxTsaBgQao63SSVwAAA-0, rgi=,  
ip=120.22.220.1,uid=1476736294, pid=26905] |
```

REQUESTS

- Each Request is mapped to a Controller & Action
- Executed by a Thread within the JVM
- Each Thread can have a name
- Threads belong to Thread Pools which have a common Prefix - e.g. `http-extapi-123`

PER-THREAD ACCOUNTING

- JVM
 - `ManagementFactory.getThreadMXBean()`
- Parfait
 - `ThreadMetric`
 - `EventTimer`
 - `EventMetricCollector.startTiming(...)`

PARFAIT

- `StandardThreadMetrics` – set of standard per-thread JVM/OS Metrics (Elapsed, CPU, Heap)
- `LoggerSink` – `log4j` output at the end of the Request cycle


```
[2019-02-27 11:49:34,790 INFO ][it.timing.LoggerSink]
[tid=http-
general-246936,rid=XHXenlcxTsaBgQao63SSVwAAA-0,rgi=,ip
=120.22.220.1,uid=1476736294,pid=26905] || Top
mailThreadBffController:getMailThread
mailThreadBffController:getMailThread Elapsed time:
own 323.622595 ms, total 323.623111 ms Total CPU: own
2.688965 ms, total 2.692129 ms User CPU: own 0.0 ms,
total 0.0 ms System CPU: own 2.774263 ms, total
2.854661 ms Heap Bytes: own 136184 By,....
```

Users

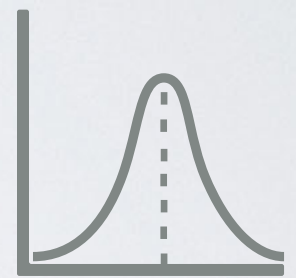
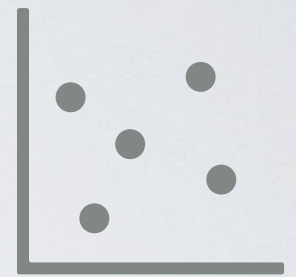


Projects

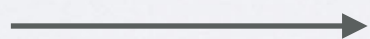
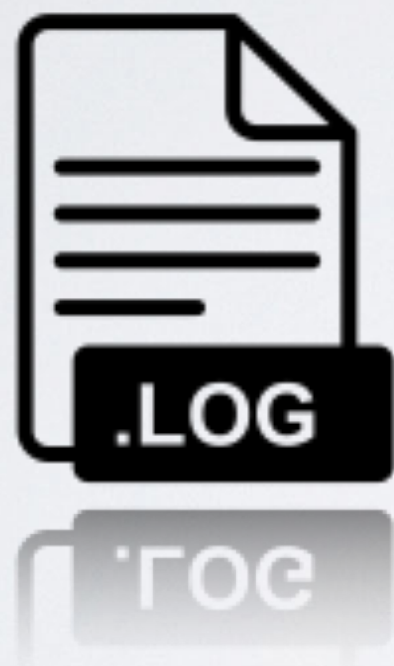


AGGREGATIONS

- We have the raw data & taxonomy
- Aggregate / Summaries
- Original/OldSchool: `sed/grep/awk`
- Modern: Sumologic



SHORT TERM ANALYSIS



▼

`_sourceCategory=au1/babylon/metrics | where !isempty(projectId) | sum(heapbytesown) by projectId | top 10 projectId by _sum`

Last 15 Minutes

Start

☆

Save As

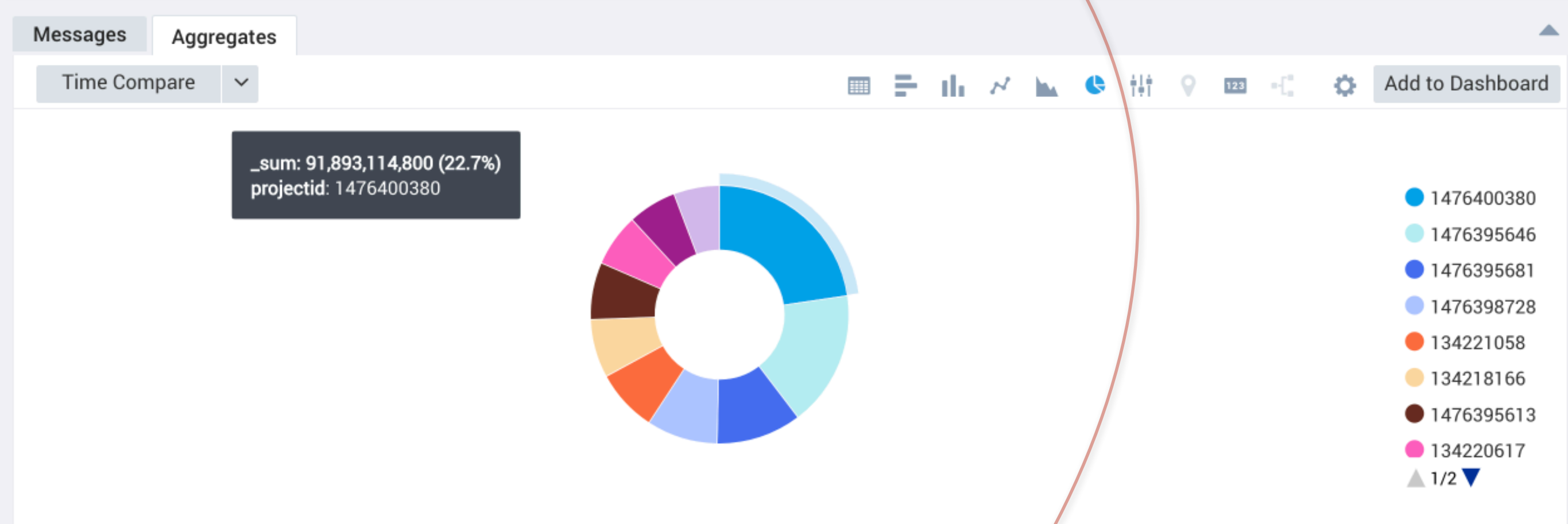
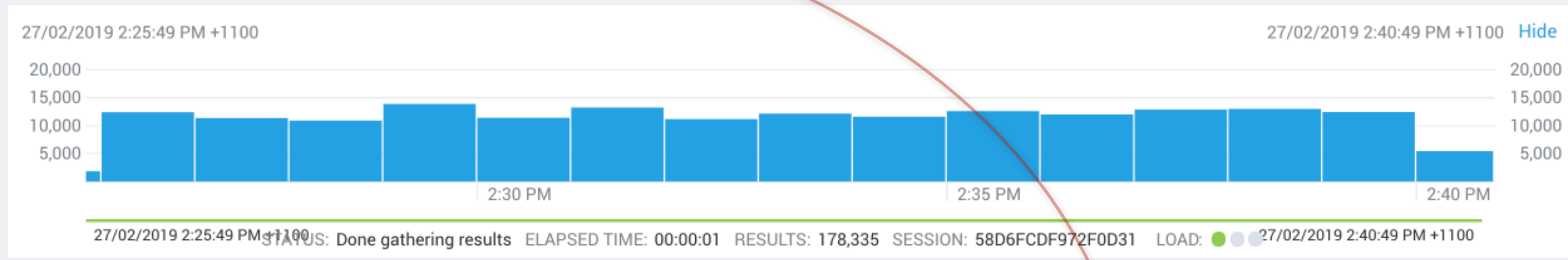
Info

Share

Pin

Live Tail

☐ Use Receipt Time



```
_sourceCategory=au1/babylon/metrics | where !isempty(projectId) | sum(heapbytesown) by  
projectId | top 10 projectId by _sum
```


▼

`_sourceCategory=au1/babylon/metrics | where !isempty(projectId) | sum(heapbytesown) by projectId | top 10 projectId by _sum`

Last 15 Minutes

Start

☆

Save As

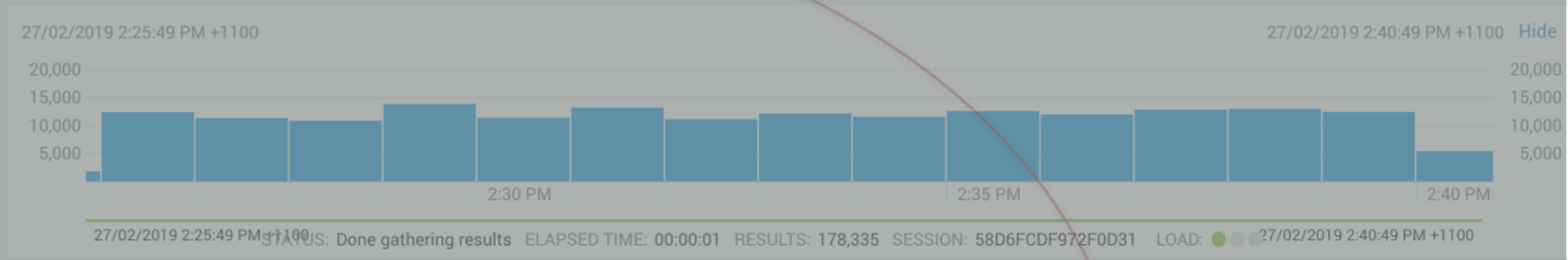
Info

Share

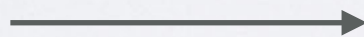
Pin

Live Tail

☐ Use Receipt Time



TRENDING



Microsoft®
SQL Server®

Analysis Services

TRENDING

- Daily sync of PCP archives to central Repo
- Extract, Transform, Load (ETL) into SQL Server
- Metric values mapped into STAR schema
- SQL Server Analysis Reporting

Year2017, 2018, 2019

InstancecodeALDAR, DXB, HKG, LHR, MEL, SI

MonthAll

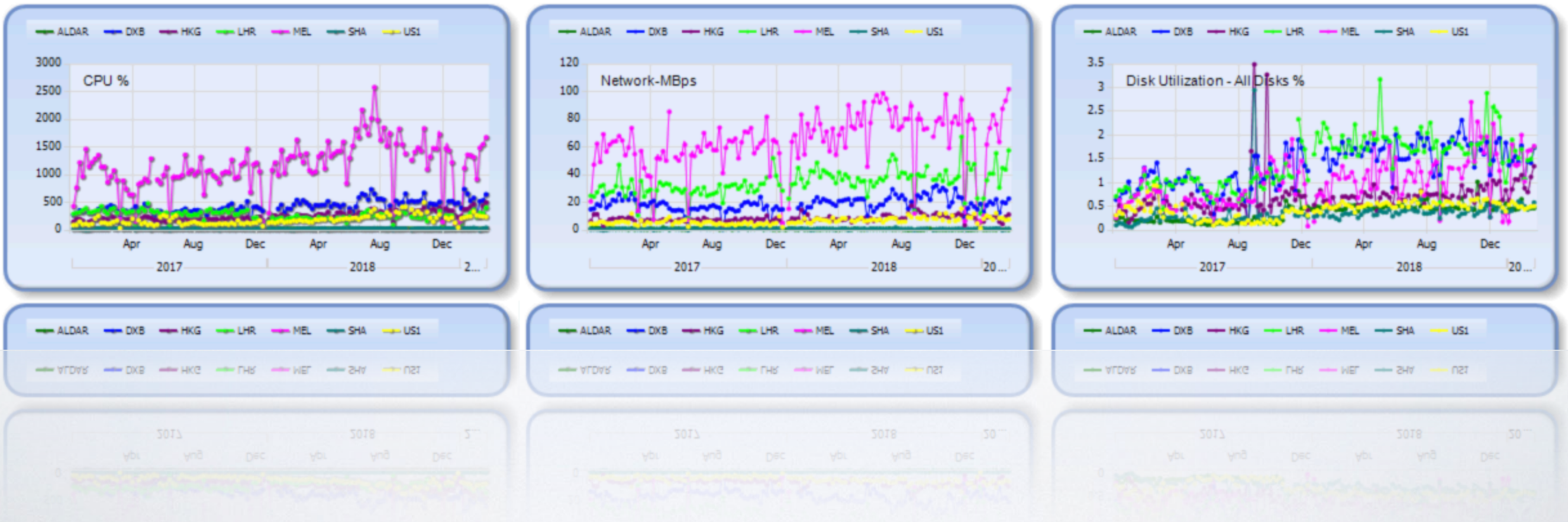
Measure NameSLA

View Report

1 of 2 ?

Find | Next

Application Server



SUMMARY



Problems



PCP & Parfait



Bigger Picture



THANK YOU