CERN

Storage Latency for Oracle DBAs

Luca Canali – CERN

Marcin Blaszczyk - CERN

Outline

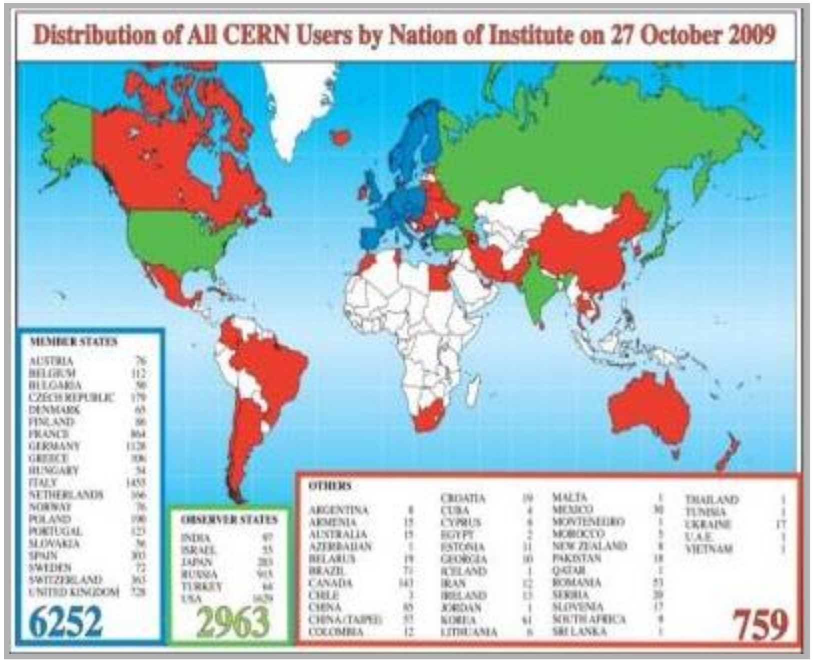
- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency in Oracle
- Examples
- Tools
- Conclusions

Outline

- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency in Oracle
- Examples
- Tools
- Conclusions

CERN

- European Organization for Nuclear Research founded in 1954
- 20 Member States, 7 Observer States + UNESCO and UE
- 60 Non-member States collaborate with CERN
- 2400 staff members work at CERN as personnel, 10 000 more researchers from institutes world-wide



LHC, Experiments, Physics

- Large Hadron Collider (LHC)
  - World's largest and most powerful particle accelerator
  - 27km ring of superconducting magnets
  - Currently undergoing upgrades, restart in 2015
- The products of particle collisions are captured by complex detectors and analyzed by software in the experiments dedicated to LHC
- **Higgs boson discovered!**

- The Nobel Prize in Physics 2013 was awarded jointly to François Englert and Peter W. Higgs *"for the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, and which recently was confirmed through the discovery of the predicted fundamental particle, by the ATLAS and CMS experiments at CERN's Large Hadron Collider"*

CERN

WLCG

- The world's largest scientific computing grid

More than 100 Petabytes
of data stored and analysed.
Increasing: 20+ Petabytes/year

Over 68 000 physical CPUs
Over 305 000 logical CPUs

157 computer centres in 36 countries

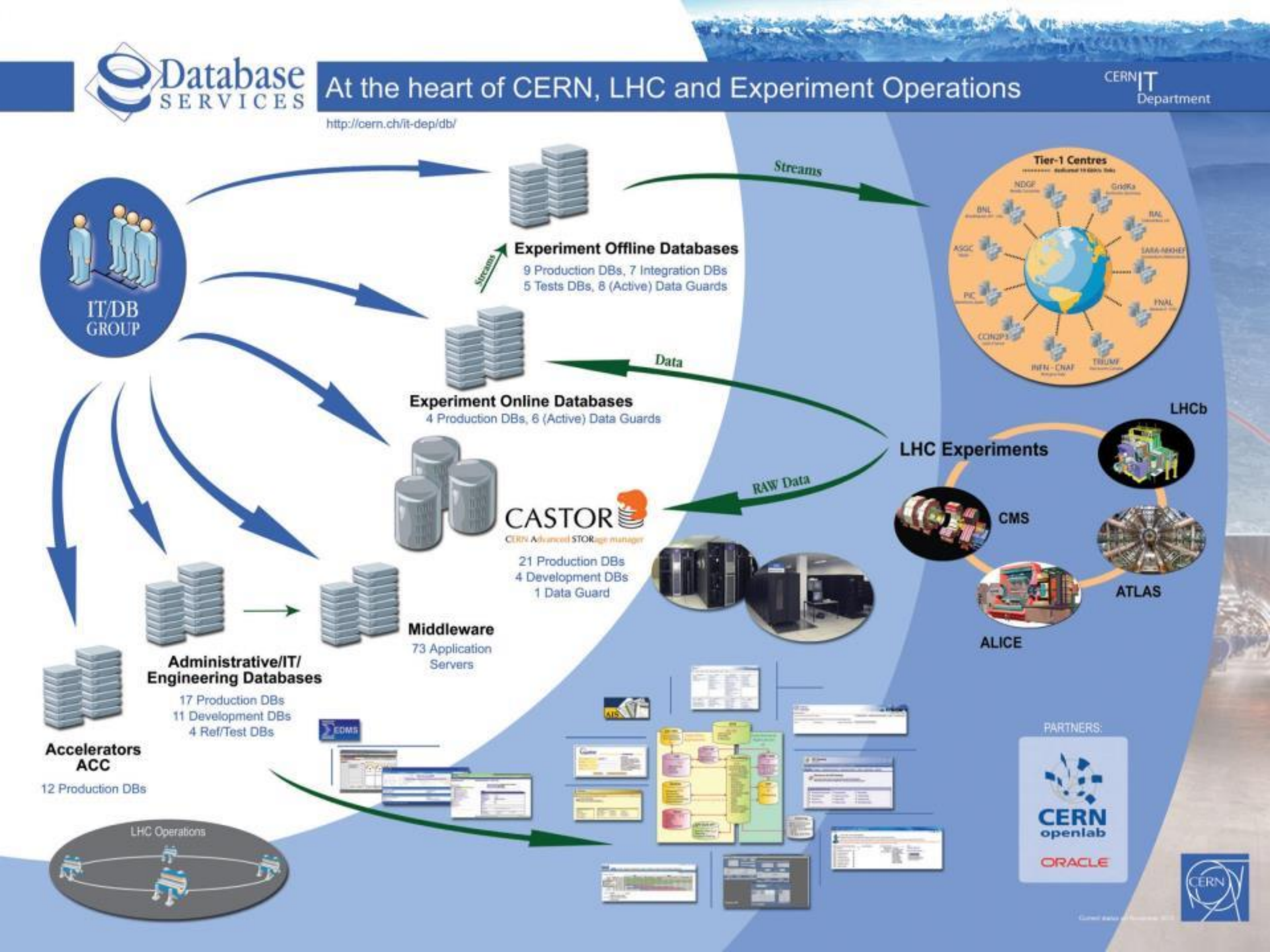
More than 8000 physicists with real-time access to LHC data

CERN's Databases

- ~100 Oracle databases, most of them RAC
  - Mostly NAS storage plus some SAN with ASM
  - ~500 TB of data files for production DBs in total

- Examples of critical production DBs:
  - LHC logging database ~170 TB, expected growth up to ~70 TB / year
  - 13 Production experiments' databases

- Relational DBs play a key role in the LHC production chains
  - Accelerator logging and monitoring systems
  - Online acquisition, offline: data (re)processing, data distribution, analysis
  - Grid infrastructure and operation services
    - Monitoring, dashboards, etc.
  - Data management services
    - File catalogues, file transfers, etc.
  - Metadata and transaction processing for tape storage system



Database SERVICES
At the heart of CERN, LHC and Experiment Operations
CERN IT Department
http://cern.ch/it-dep/db/
IT/DB GROUP
Streams
Experiment Offline Databases
9 Production DBs, 7 Integration DBs
5 Tests DBs, 8 (Active) Data Guards
Streams
Tier-1 Centres
NDGF
GridKa
BNL
RAL
ASGC
SARA-NIKHEF
PIC
FNAL
CCIN2P3
INFN - CNAF
TRIUMF
Data
Experiment Online Databases
4 Production DBs, 6 (Active) Data Guards
LHC Experiments
LHCb
CMS
ATLAS
ALICE
RAW Data
CASTOR
CERN Advanced STORage manager
21 Production DBs
4 Development DBs
1 Data Guard
Middleware
73 Application Servers
Administrative/IT/ Engineering Databases
17 Production DBs
11 Development DBs
4 Ref/Test DBs
Accelerators ACC
12 Production DBs
LHC Operations
EDMS
AIS
PARTNERS:
CERN openlab
ORACLE
CERN

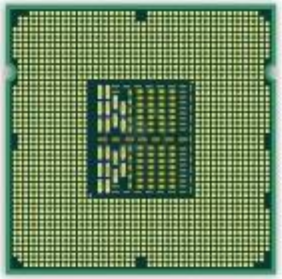
Outline

- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency data in Oracle
- Examples
- Tools
- Conclusions

Latency

- Latency, a measure of time.
    - In the context of this presentation: time to access data






Understanding Latency

- How long I should wait for baby elephant?
  - Elephant gestation period ~22 month

  - Latency: 22 months

Understanding Throughput

- What if I want 2 baby elephants?

- Throughput has doubled:
  - 2 elephants in 22 months
- Latency: still 22 months

I/O Operations Per Second

- IOPS is a measure of throughput
- IOPS depends also on latency
- Latency differs for
  - *'random'* reads
  - *'sequential'* reads

- How can we get more IOPS without increasing the latency?
  - Use Many HDDs!

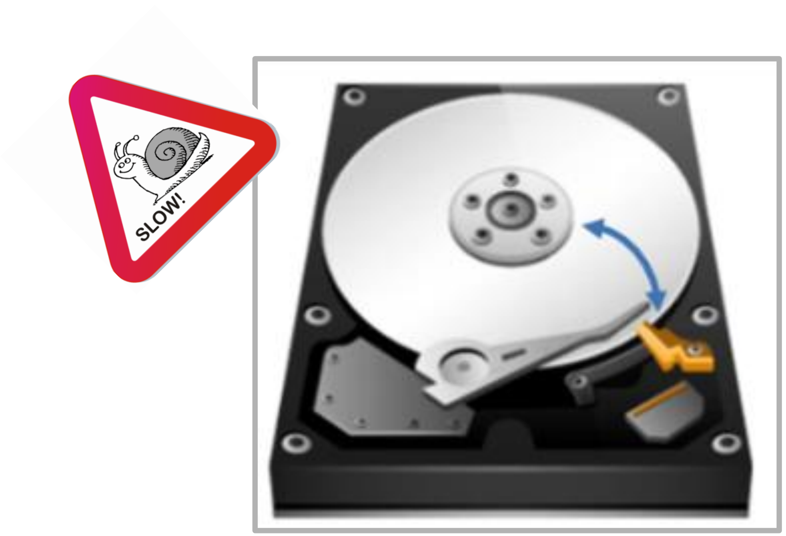
Why We Care About Storage Latency

- Performance analysis and tuning:
  - Where is the time spent during a DB call?
  - What response time do the users see from the DB?

- OLTP-like workloads:
  - Response time can be dominated by I/O latency
  - Index-based access, nested loops joins

Physical Sources of Latency

- Blocking I/O calls:
  - Think access to a large table via an index
  - Random access
  - HDD: head movement and disk spinning latency



What can we do: Hardware

- Current trends for HW
  - Large SSD cache in storage
  - Tiered storage
  - Servers with large amounts of memory
    - Reduce (random) reads
    - Caching large amounts of data
    - Trends towards in-memory DBs

- A balance act performance vs. cost

What can we do: Software

- Big gains in application/SQL optimization
  - SW optimisation beats HW optimisation most of the times

- Oracle tuning:
  - Understand when single-block access is not optimal
  - Full scan vs. index-based access
  - Hash join vs. nested loop
  - In general: get a good execution plan

So Where is the Problem?

DB Admin:

- Storage is slow!

Storage Admin:

- The problem is with the DB!

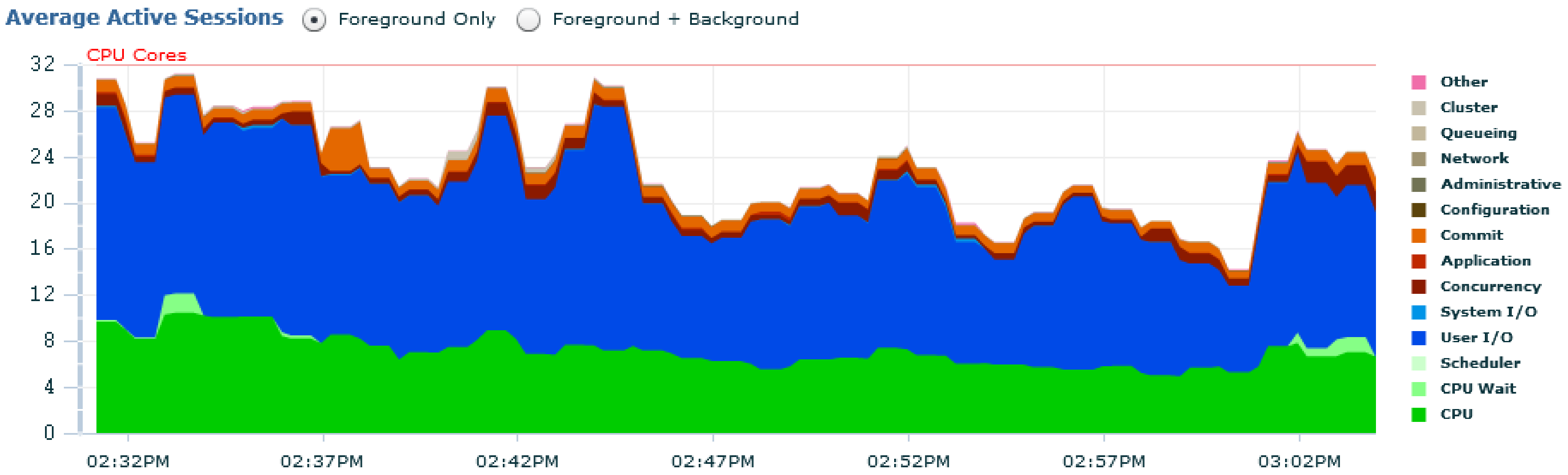
- Reality check:
  - Lack of clear storage performance data.
  - Changing database workloads.

Outline

- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency in Oracle
- Examples
- Tools
- Conclusions

Transactional Workload

- Example from OEM



  - DB time dominated by ‘db file sequential read’
    - CPU is not a bottleneck
    - Ignore locks and other serialization events

Oracle Wait Events

- Can we troubleshoot a storage issue from the DB engine?
  - Not in the general case

- What can we do?
  - Oracle wait event instrumentation is great
  - Wait event histograms is a key source of data

Wait Event Analysis

- We want to drill down
  - ‘db file sequential read’ wait event
  - Also useful for ‘log file sync’ event

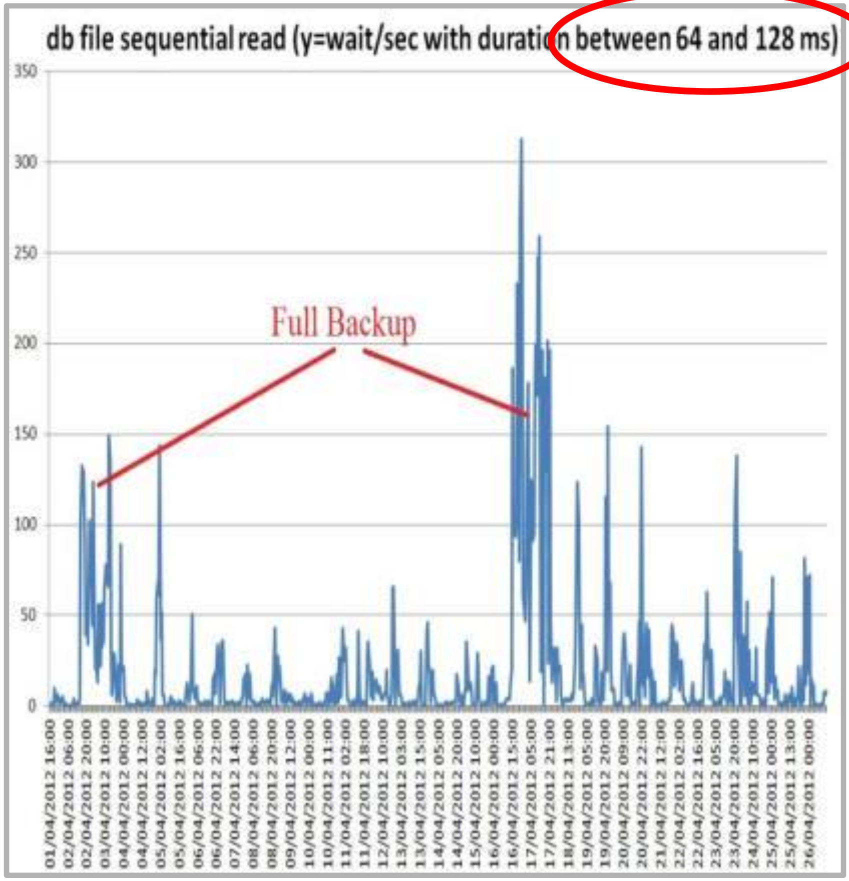
- What can we gain?
  - Make educated guesses of what is happening on the storage
  - Attack the root causes of the problem

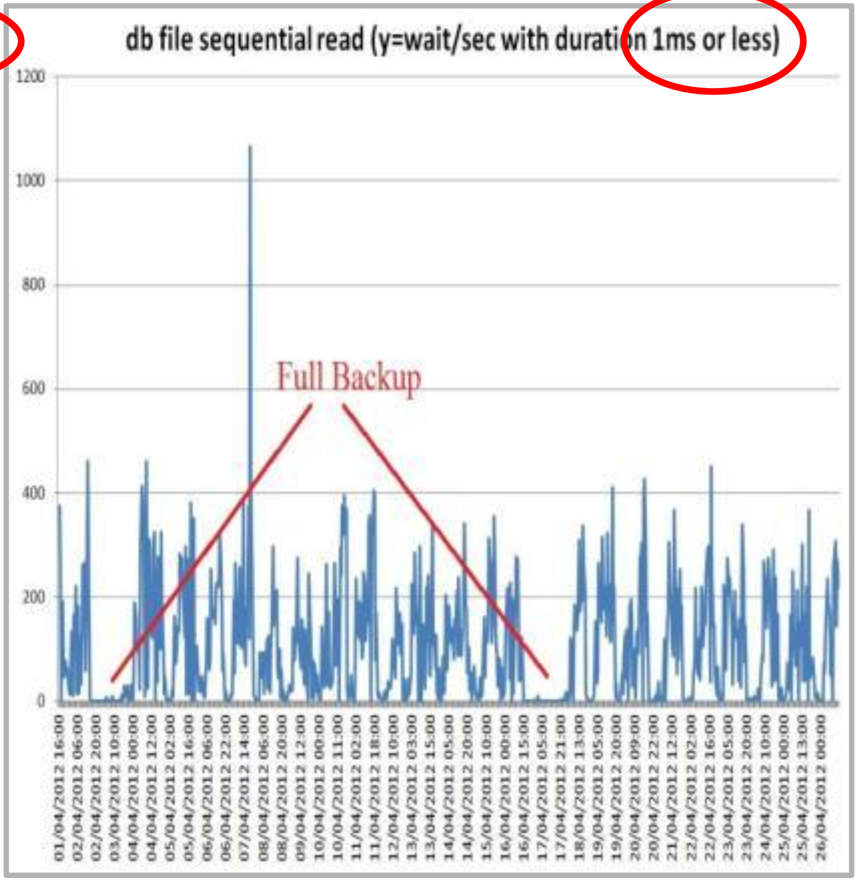
A Story From Our Production

- Migrated to a new storage system
  - NAS storage with SSD cache
  - Good performance: because of low-latency reads from SSD
- Issue:
  - From time to time production shows unacceptable performance
- Analysis:
  - The issue appears when the backup runs!

Wait Event Drill Down

Number of waits





Very slow reads appear

Reads from SSD cache go to zero

What We Found

- AWR data used to examine the issue
  - DBA_HIST_EVENT_HISTOGRAM
  - Wait event: db file sequential read
- I/O slow during backups
  - because fewer I/O requests were served from SSD

- Note: how we worked around this issue
  - Short term: moved backup to Active Data Guard replica
  - Medium term: upgraded filer model

Lesson Learned

- If response time is dominated by db_file_sequential_read
  - Drill down on wait event histogram
  - Average latency values are not good enough
  - Latency details provide info on what is happening on the storage

Real-Time Monitoring

- Problem:
  - How to perform real-time monitoring of the event latency?
- Answer: V$EVENT_HISTOGRAM
  - Cumulative counters
  - We need to compute deltas

Monitoring Latency - Snapshots

- Custom script: ehm.sql

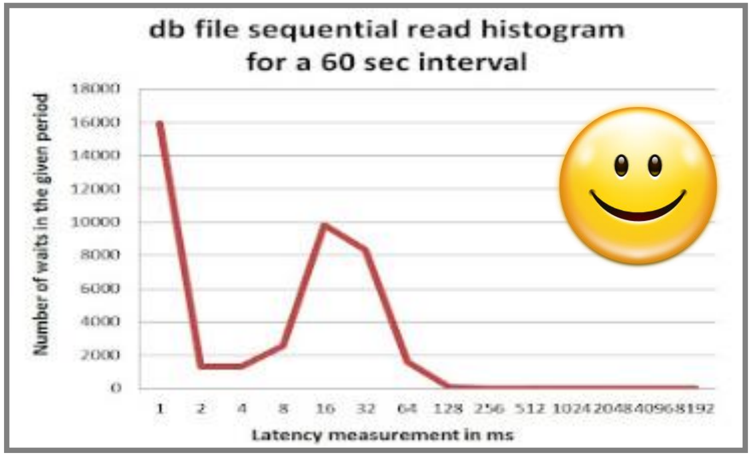
```
primary:system@orclrac1> @ehm 60 db%sequential

waiting for 60 sec (delta measurement interval = 60 sec)

Wait (ms)    N#          Event                   Last update time
----------- ----------- ----------------------- -----------------------------------
1            12588       db file sequential read 20-NOV-13 04.52.02.549024 PM +02:00
2            638         db file sequential read 20-NOV-13 04.52.02.323209 PM +02:00
4            241         db file sequential read 20-NOV-13 04.52.00.017278 PM +02:00
8            1032        db file sequential read 20-NOV-13 04.52.02.407010 PM +02:00
16           6128        db file sequential read 20-NOV-13 04.52.02.520877 PM +02:00
32           3865        db file sequential read 20-NOV-13 04.52.02.526403 PM +02:00
64           622         db file sequential read 20-NOV-13 04.52.02.475484 PM +02:00
128          48          db file sequential read 20-NOV-13 04.52.02.454875 PM +02:00
256          2           db file sequential read 20-NOV-13 04.51.35.738163 PM +02:00
512          1           db file sequential read 20-NOV-13 04.51.54.617231 PM +02:00
1024         13          db file sequential read 20-NOV-13 04.52.01.560293 PM +02:00
2048         0           db file sequential read 20-NOV-13 03.19.40.350234 PM +02:00
4096         0           db file sequential read 15-NOV-13 02.25.22.371191 AM +02:00
8192         0           db file sequential read 31-OCT-13 01.01.10.757675 AM +02:00
16384        0           db file sequential read 28-OCT-13 11.51.50.122887 PM +02:00
32768        0           db file sequential read 11-OCT-13 12.42.21.599088 PM +02:00
65536        0           db file sequential read 11-OCT-13 12.42.21.601458 PM +02:00
131072       0           db file sequential read 11-OCT-13 12.42.21.606092 PM +02:00

Avg_wait(ms) N#          Tot_wait(ms) Event
------------ ----------- ------------ -------------------
8.5          25177       214095.1     db file sequential read
```

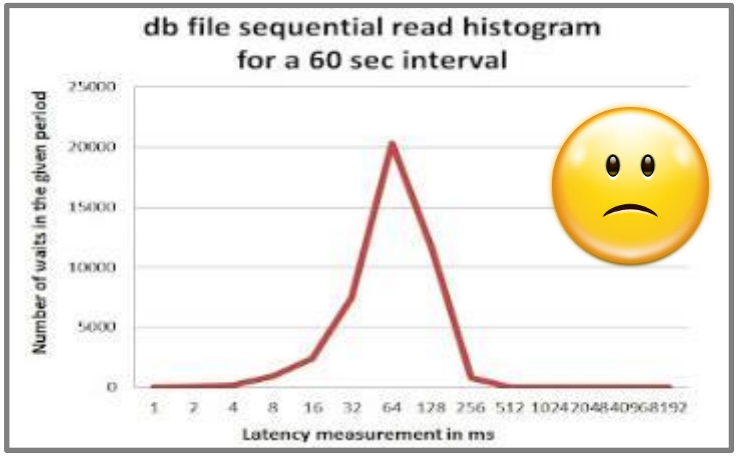
Script can be downloaded from: http://canali.web.cern.ch/canali/resources.htm

Monitoring Latency - Snapshots

```
primary:system@orclrac1> @ehm 60 db%sequential

waiting for 60 sec (delta measurement interval = 60 sec)

Wait (ms)     N#          Event
---------     -------     --------------------
1             15958       db file sequential read
2             1317        db file sequential read
4             1355        db file sequential read
8             2590        db file sequential read
16            9845        db file sequential read
32            8339        db file sequential read
64            1607        db file sequential read
128           124         db file sequential read
256           7           db file sequential read
512           1           db file sequential read
1024          15          db file sequential read
2048          0           db file sequential read
4096          0           db file sequential read
8192          0           db file sequential read

Avg_wait(ms)  N#          Tot_wait(ms) Event
------------  -------     ------------ --------------------
10.3          41159       423786       db file sequential read
```

```
primary:system@orclrac1> @ehm 60 db%sequential

waiting for 60 sec (delta measurement interval = 60 sec)

Wait (ms)     N#          Event
---------     -------     --------------------
1             16          db file sequential read
2             61          db file sequential read
4             230         db file sequential read
8             930         db file sequential read
16            2427        db file sequential read
32            7488        db file sequential read
64            20352       db file sequential read
128           11616       db file sequential read
256           814         db file sequential read
512           20          db file sequential read
1024          22          db file sequential read
2048          0           db file sequential read
4096          0           db file sequential read
8192          0           db file sequential read
16384         0           db file sequential read
32768         0           db file sequential read
65536         0           db file sequential read
131072        0           db file sequential read

Avg_wait(ms)  N#          Tot_wait(ms) Event
------------  -------     ------------ --------------------
53.4          43992       2350745.9    db file sequential read
```



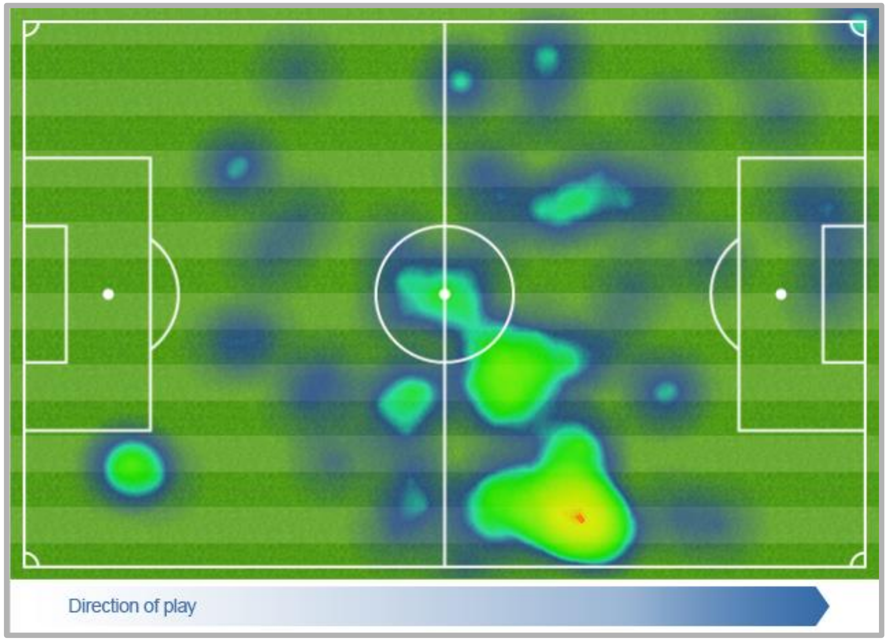


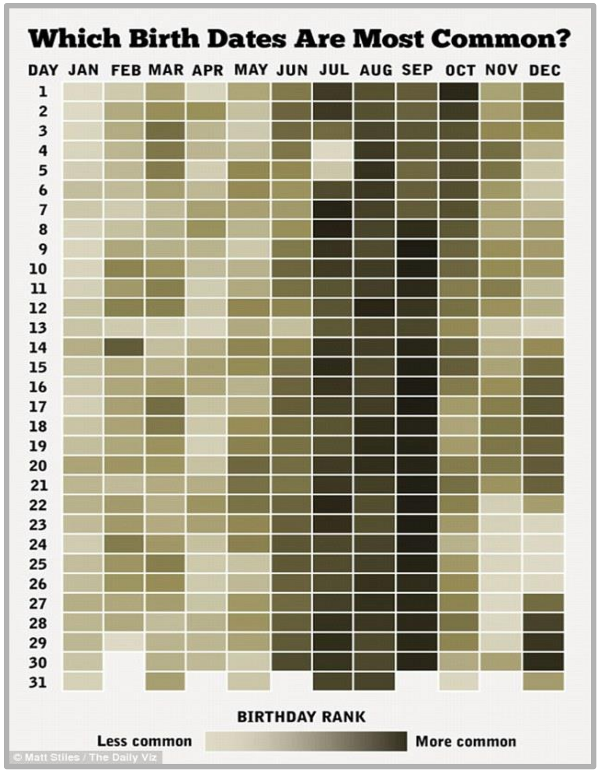
Display Latency Data over Time

- It's a three dimensional representation:
  - Latency bucket, value, time
- This problem has been solved before!
- Heat Map representation
  - Used for example in Sun ZFS Storage 7000 Analytics
  - Reference: Brendan Gregg, Visualizing system latency, Communications of the ACM, July 2010

Heat Maps

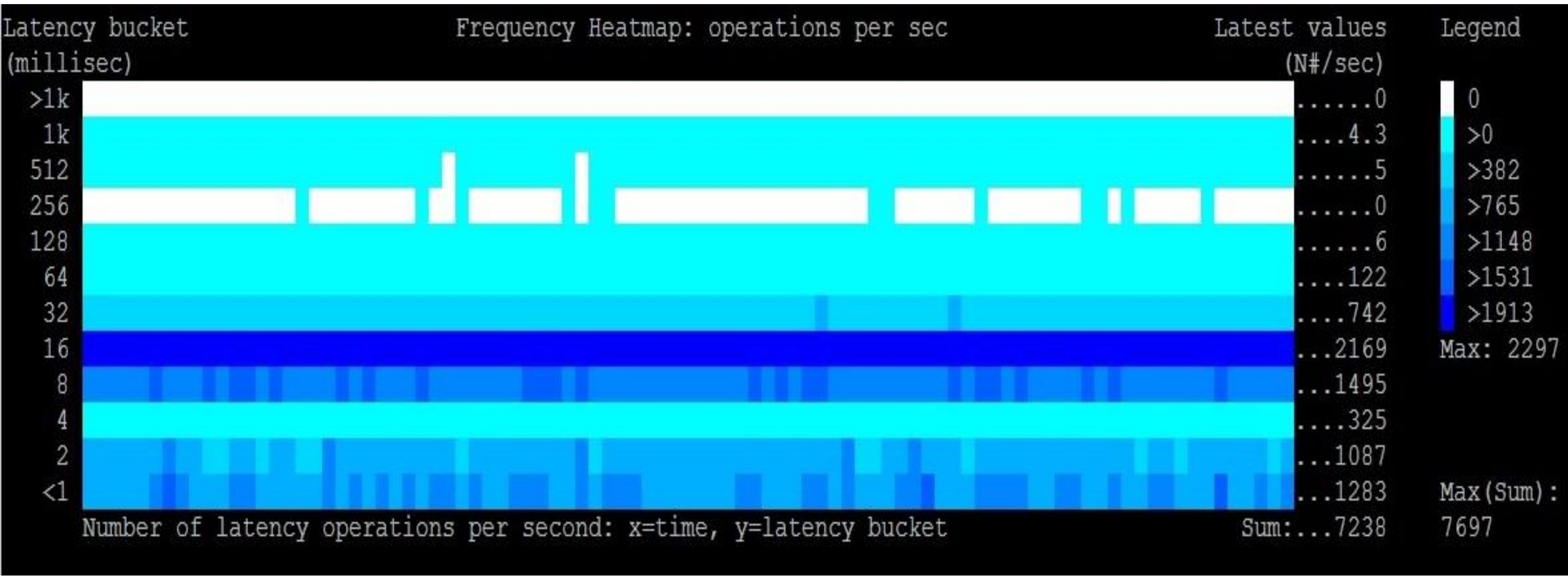
- By Definition
  - *graphical representation of data where the individual values contained in a matrix are represented as colours (wikipedia)*
- Examples:





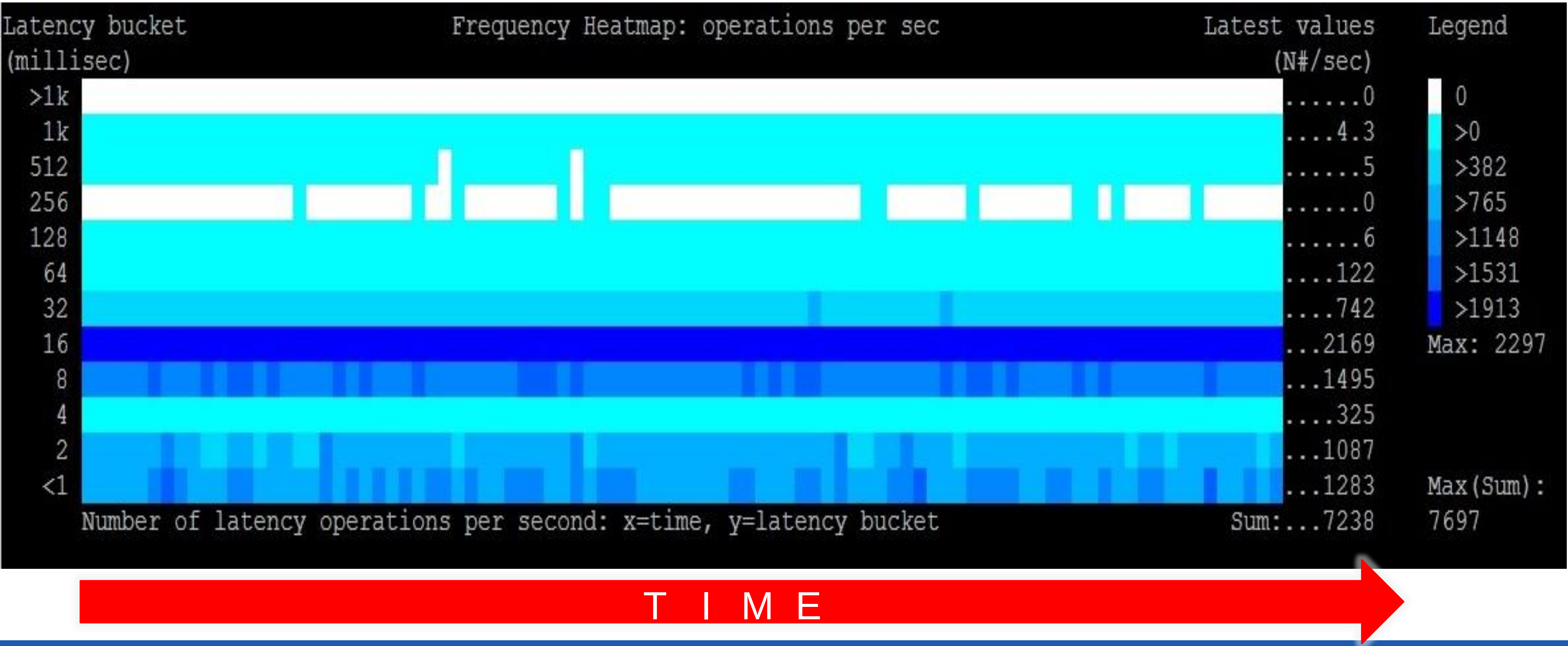
Latency Heat Maps - Frequency

- X=time, Y=latency bucket
- Colour=events per second (e.g. IOPS)



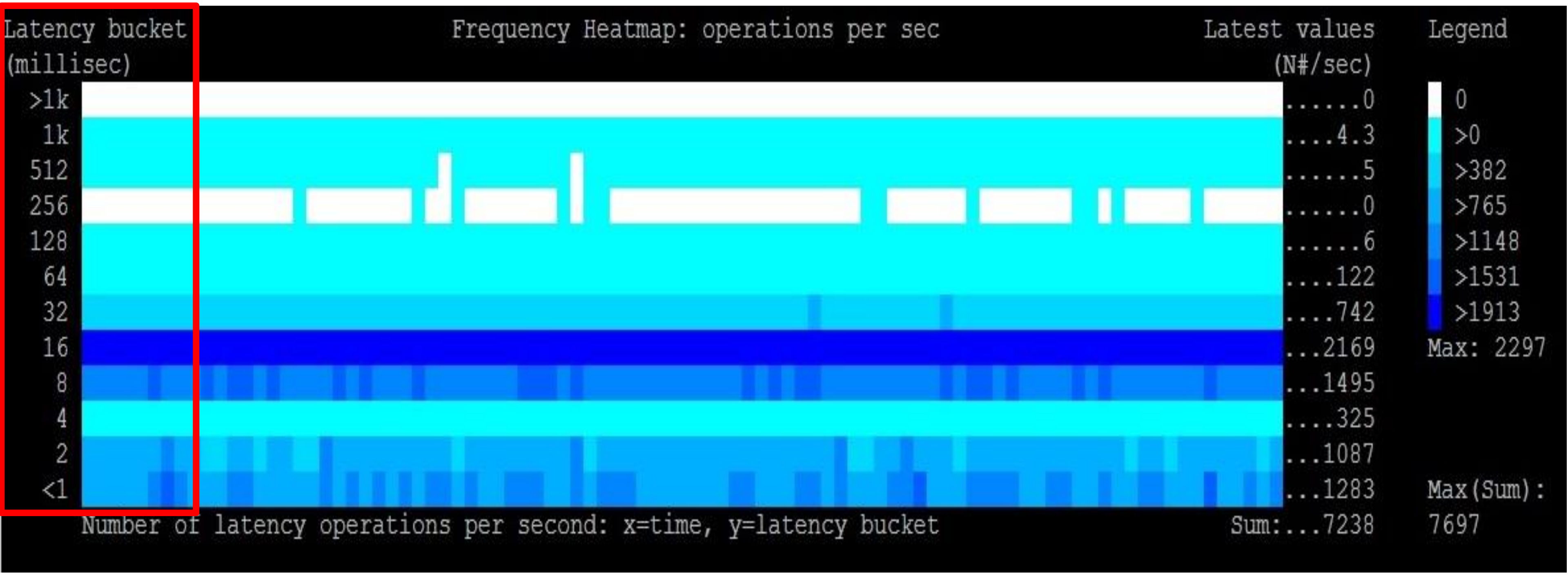
Latency Heat Maps - Frequency

- X=time, Y=latency bucket
- Colour=events per second (e.g. IOPS)



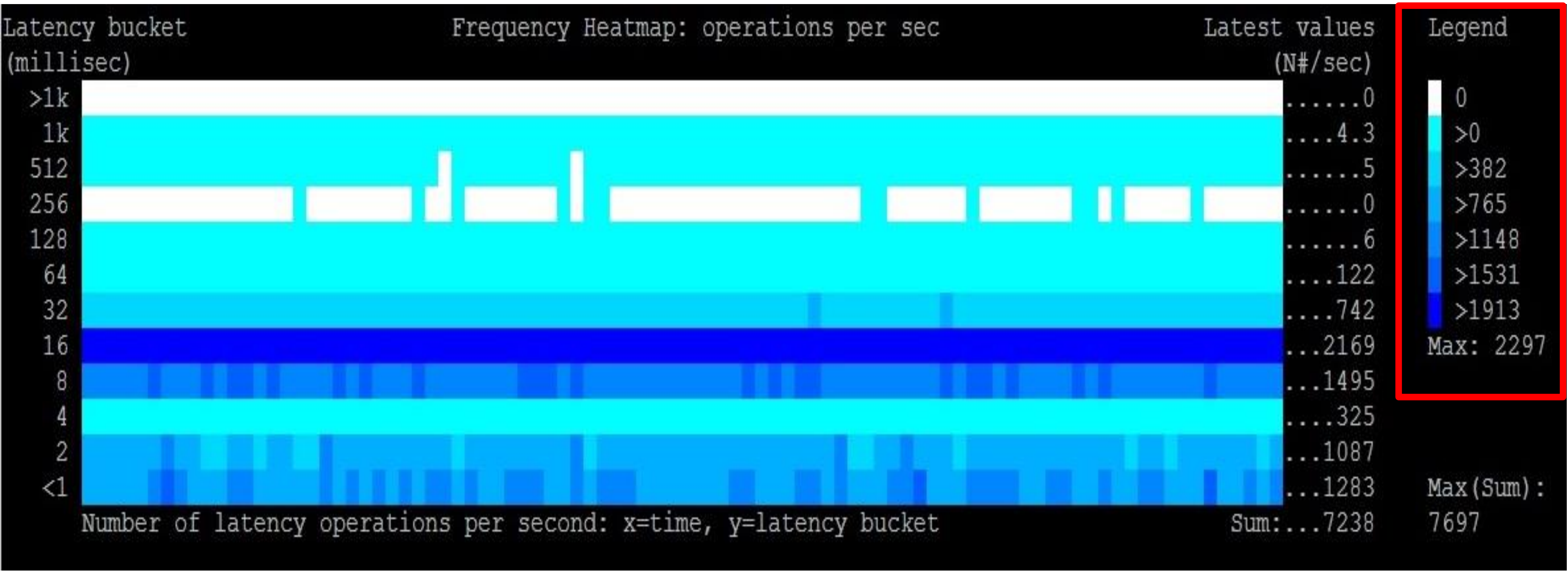
Latency Heat Maps - Frequency

- X=time, Y=latency bucket
- Colour=events per second (e.g. IOPS)



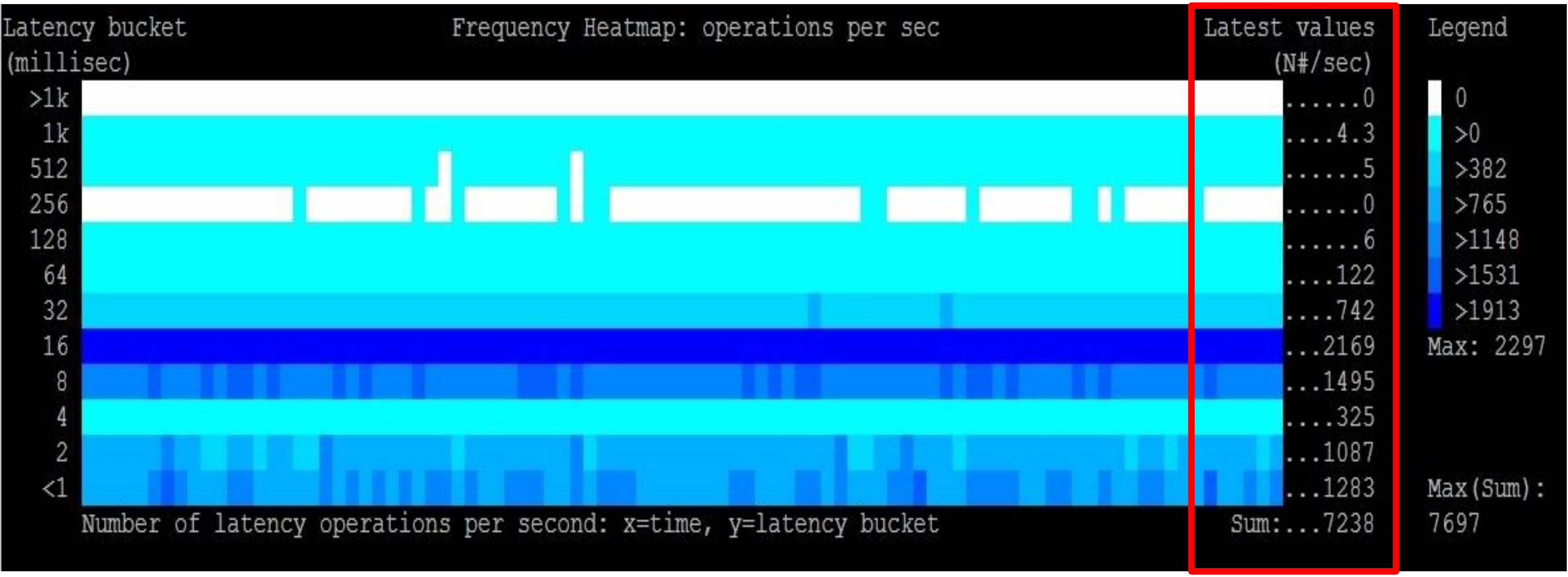
Latency Heat Maps - Frequency

- X=time, Y=latency bucket
- Colour=events per second (e.g. IOPS)



Latency Heat Maps - Frequency

- X=time, Y=latency bucket
- Colour=events per second (e.g. IOPS)



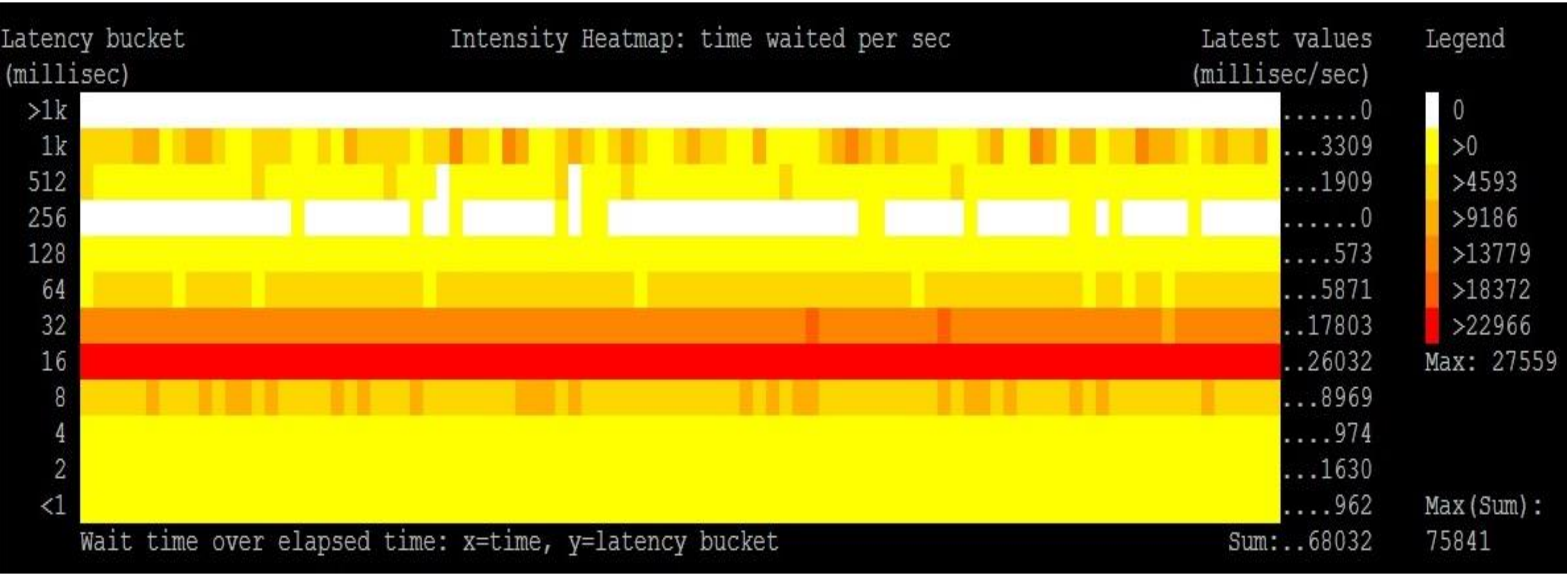
Another Metric of Interest

- How much time do we wait in a given bucket?
  - Not directly available in v$event_histogram
- How to estimate it? Example:
  - 100 waits in the bucket 8ms means
  - Wait time between 100*4 ms and 100*8 ms
  - Approximate: 100 * 6 ms [that is 100 * ¾ * 8 ms]

- Definition:
  - Intensity = 0.75 * bucket_value * frequency_count

Latency Heat Maps - Intensity

- X=time, Y=latency bucket
- Colour= intesity [time waited per bucket]



Outline

- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency in Oracle
- Examples
- Tools
- Conclusions

Stress Testing

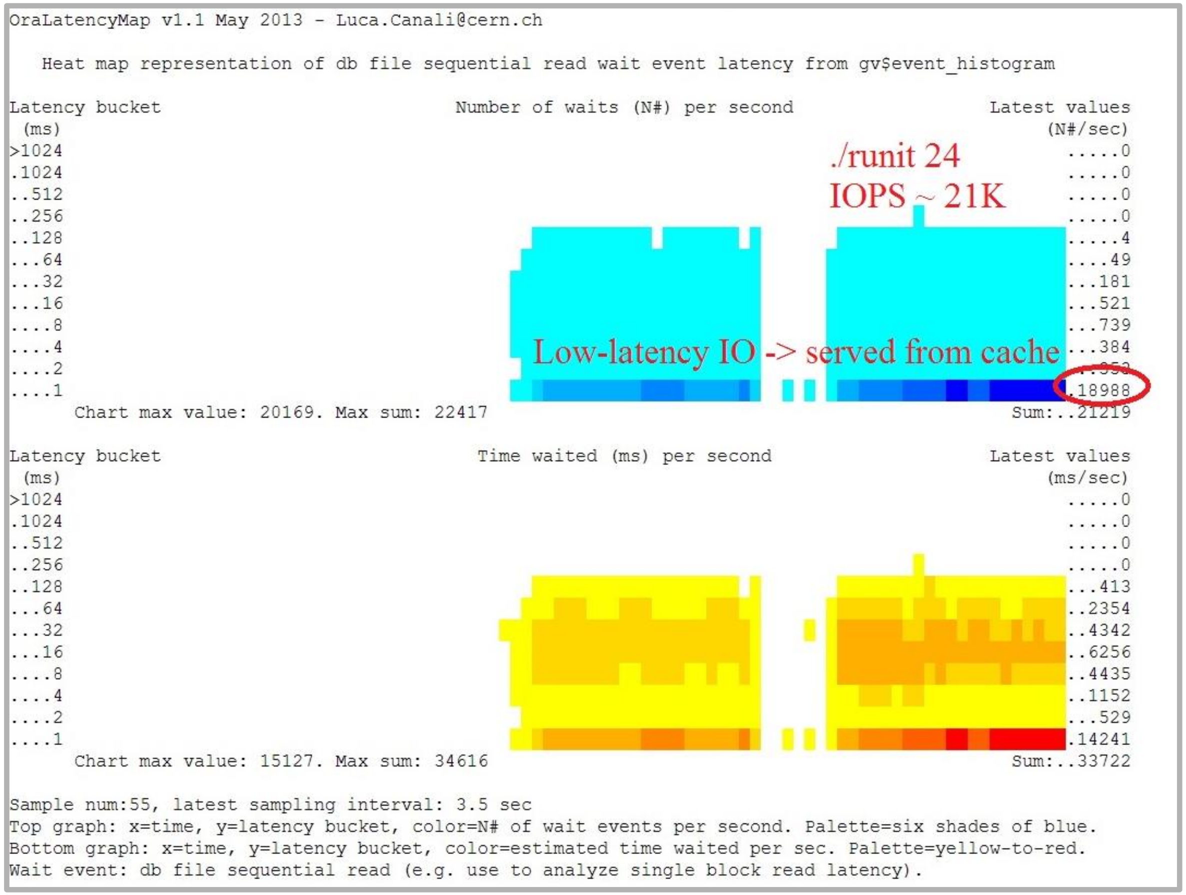
- Scenarios
  - Investigate HW performance
  - Compare different systems
  - Example: compare current storage with a new system
- It’s hard:
  - Choose test workloads that make sense
  - Understand effects of caching

SLOB 2

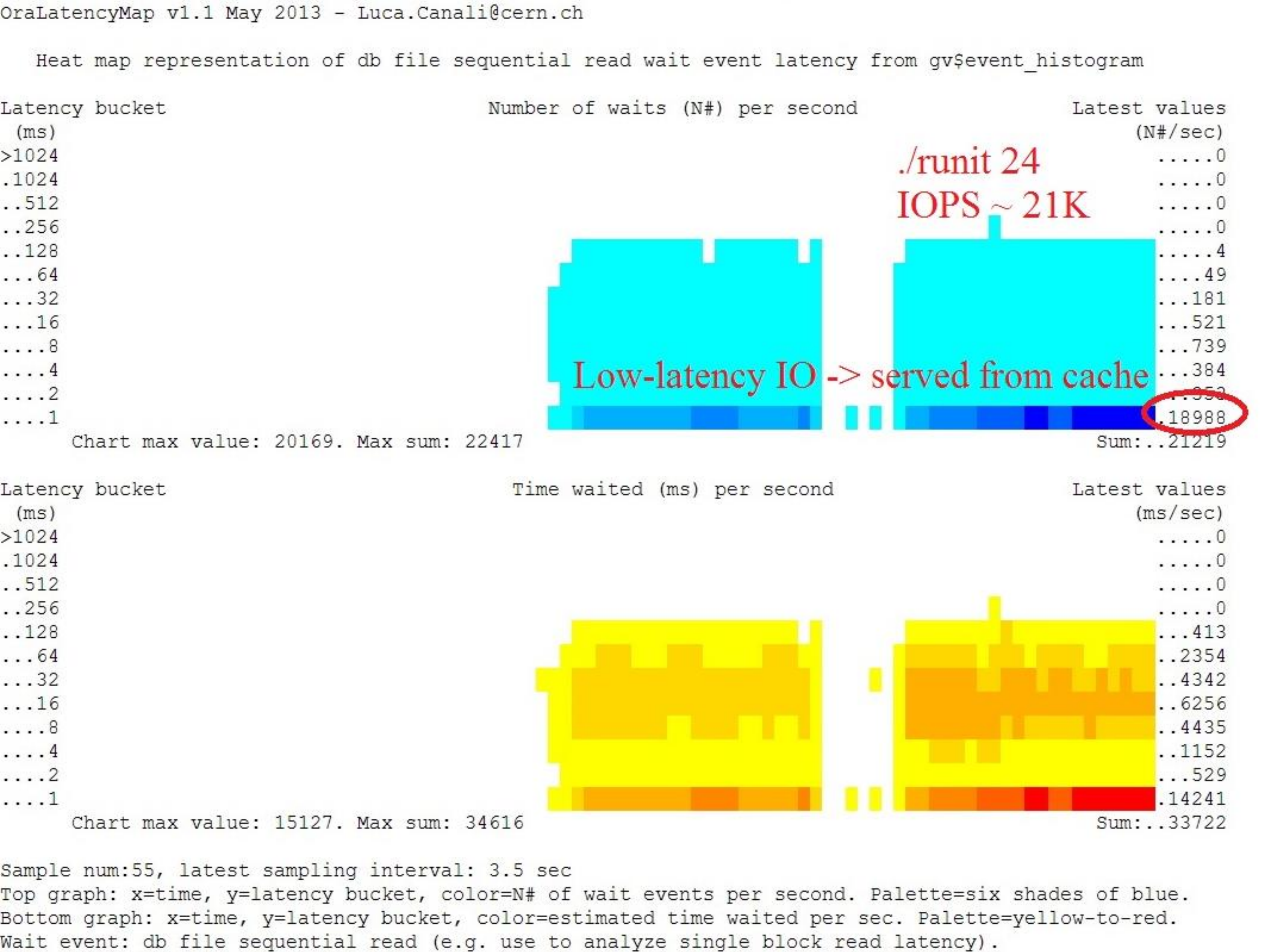
- An Oracle-based stress testing tool
  - Search: “SLOB 2 by Kevin Closson”
- Great tool generate lots of random IO
  - Directly from Oracle processes
  - Physical reads from storage
    - Become Oracle’s wait events for db file sequential read
- Size of test data is configurable
- Concurrency is configurable

Example: “Too Good To Be True”

- 23 SAS disks delivering 20K IOPS?
- It doesn’t make sense
- Latency is the clue
- Reads served by controller cache!

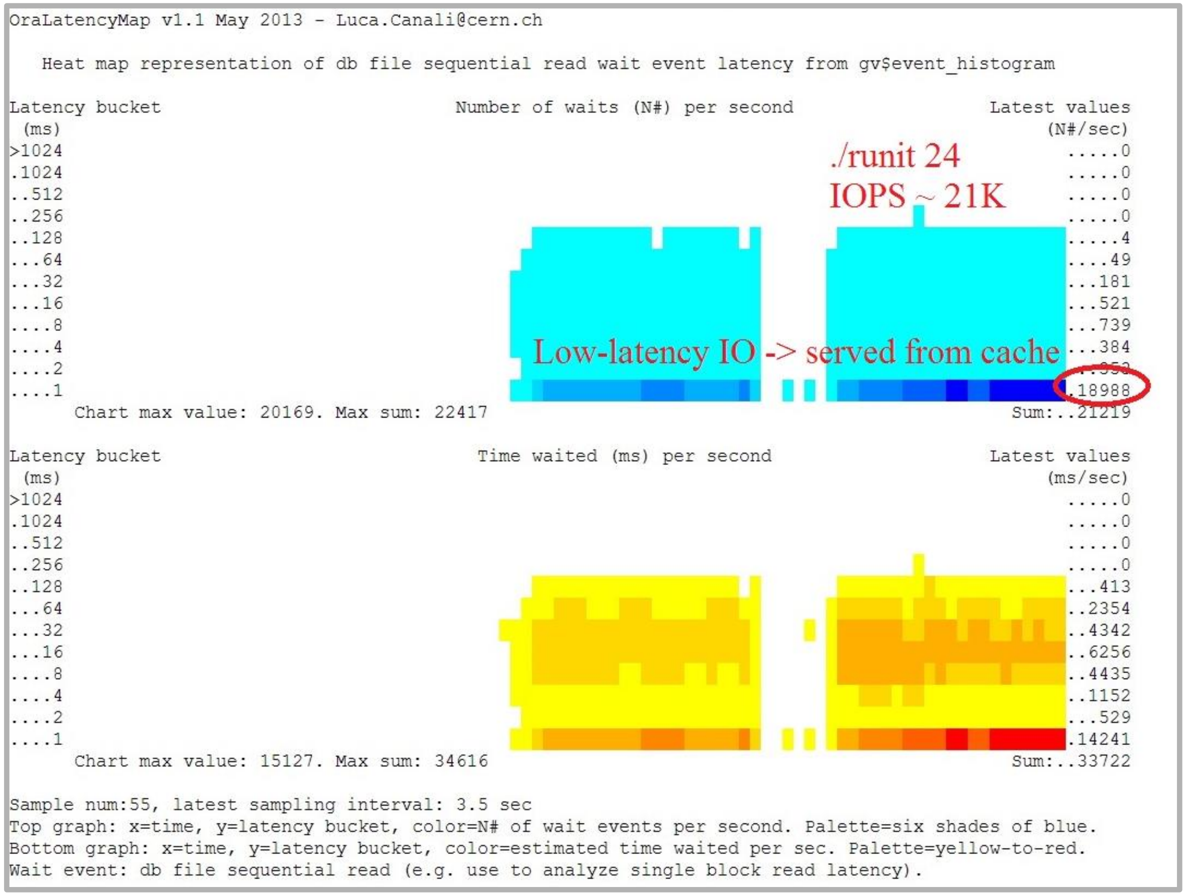


- Lesson learned: test data size was too small


OraLatencyMap v1.1 May 2013 - Luca.Canali@cern.ch
Heat map representation of db file sequential read wait event latency from gv$event_histogram
Latency bucket (ms)
Number of waits (N#) per second
Latest values (N#/sec)
>1024 0
.1024 0
..512 0
..256 0
..128 4
...64 49
...32 ...181
...16 ...521
....8 ...739
....4 ...384
....2
....1 .18988
./runit 24
IOPS ~ 21K
Low-latency IO -> served from cache
Chart max value: 20169. Max sum: 22417
Sum:..21219
Latency bucket (ms)
Time waited (ms) per second
Latest values (ms/sec)
>1024 0
.1024 0
..512 0
..256 0
..128 ...413
...64 ..2354
...32 ..4342
...16 ..6256
....8 ..4435
....4 ..1152
....2 ...529
....1 .14241
Chart max value: 15127. Max sum: 34616
Sum:..33722
Sample num:55, latest sampling interval: 3.5 sec
Top graph: x=time, y=latency bucket, color=N# of wait events per second. Palette=six shades of blue.
Bottom graph: x=time, y=latency bucket, color=estimated time waited per sec. Palette=yellow-to-red.
Wait event: db file sequential read (e.g. use to analyze single block read latency).

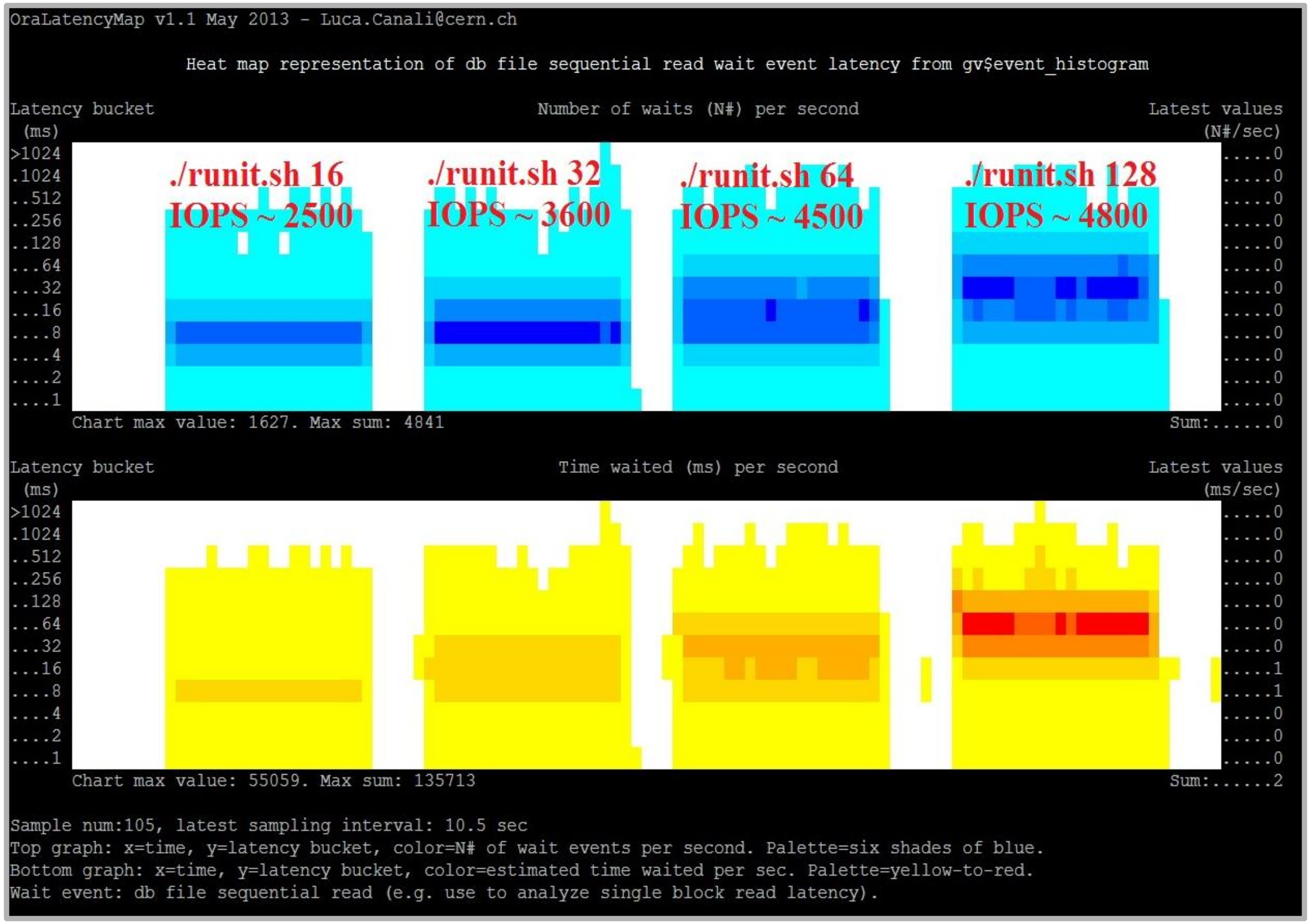
Example: “Too Good To Be True”

- 23 SAS disks delivering 20K IOPS?
- It doesn’t make sense
- Latency is the clue
- Reads served by controller cache!



- Lesson learned: test data size was too small

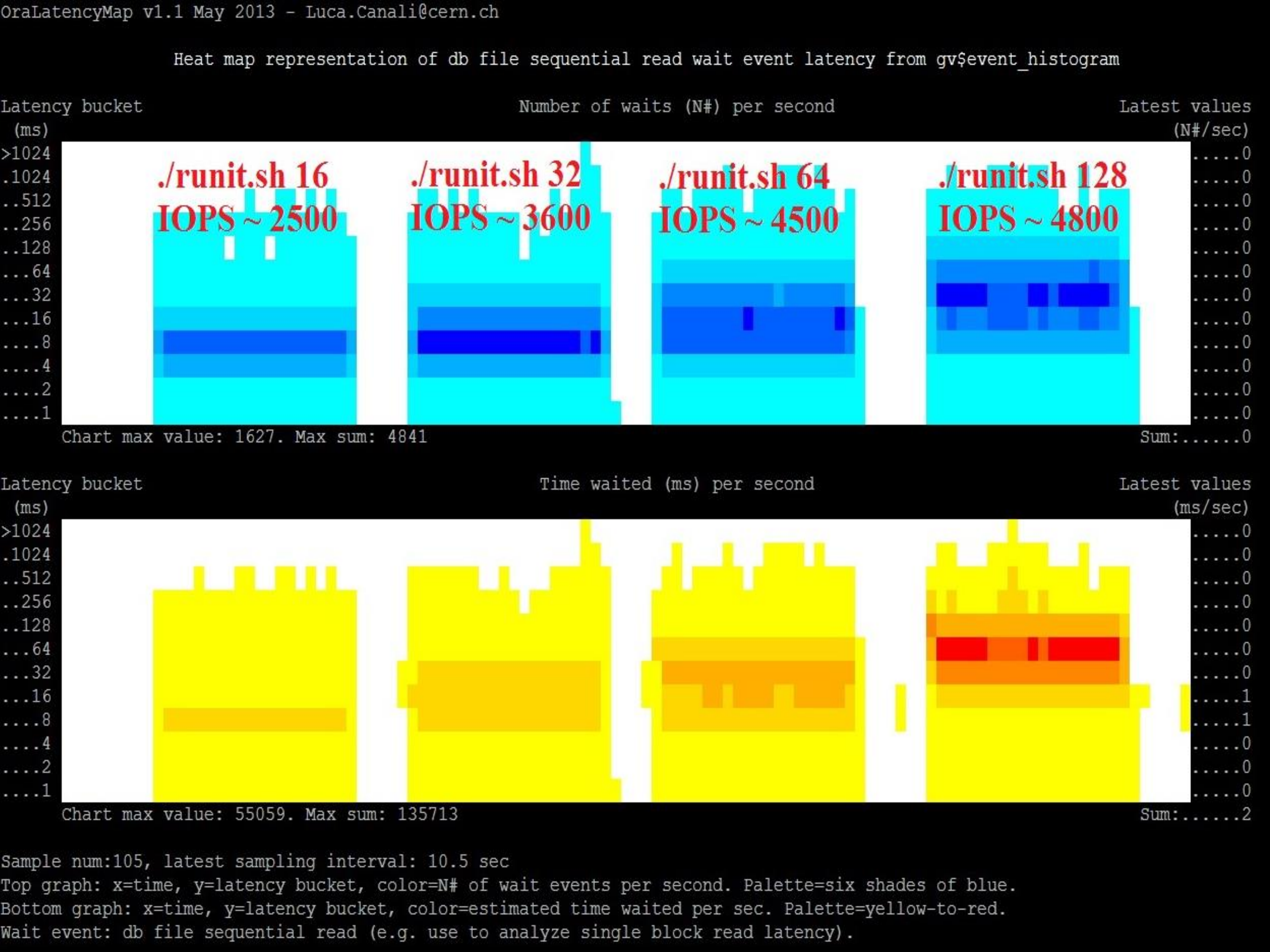
Example: “Load Till Saturation”



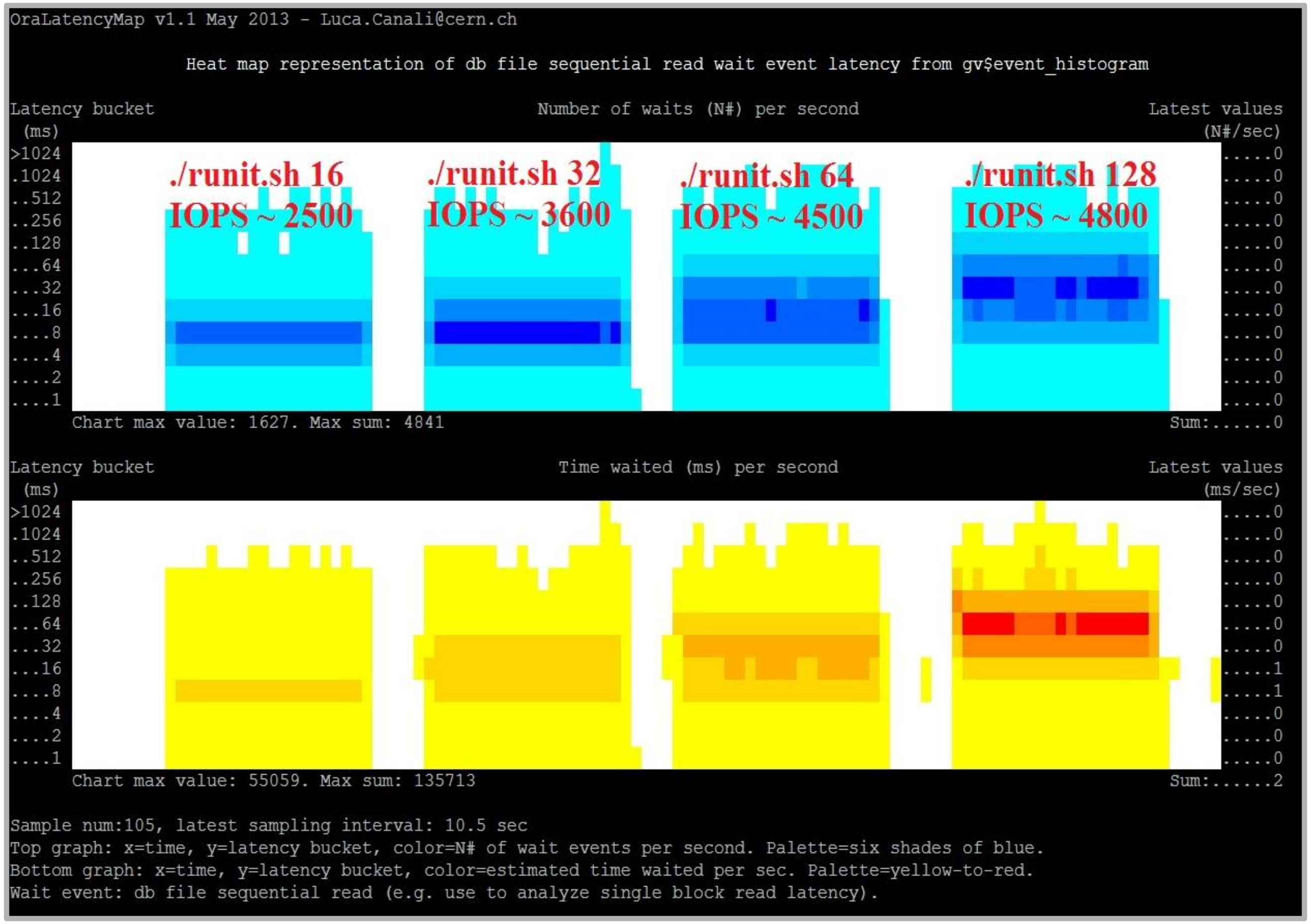
23 SAS disks
JBOD & ASM

4 consecutive tests with increasing load

Lesson learned: don’t accept IOPS numbers without latency values


OraLatencyMap v1.1 May 2013 - Luca.Canali@cern.ch
Heat map representation of db file sequential read wait event latency from gv$event_histogram
Latency bucket (ms)
Number of waits (N#) per second
Latest values (N#/sec)
>1024
.1024
..512
..256
..128
...64
...32
...16
....8
....4
....2
....1
./runit.sh 16
IOPS ~ 2500
./runit.sh 32
IOPS ~ 3600
./runit.sh 64
IOPS ~ 4500
./runit.sh 128
IOPS ~ 4800
Chart max value: 1627. Max sum: 4841
Sum:......0
Latency bucket (ms)
Time waited (ms) per second
Latest values (ms/sec)
Chart max value: 55059. Max sum: 135713
Sum:......2
Sample num:105, latest sampling interval: 10.5 sec
Top graph: x=time, y=latency bucket, color=N# of wait events per second. Palette=six shades of blue.
Bottom graph: x=time, y=latency bucket, color=estimated time waited per sec. Palette=yellow-to-red.
Wait event: db file sequential read (e.g. use to analyze single block read latency).

Example: “Load Till Saturation”



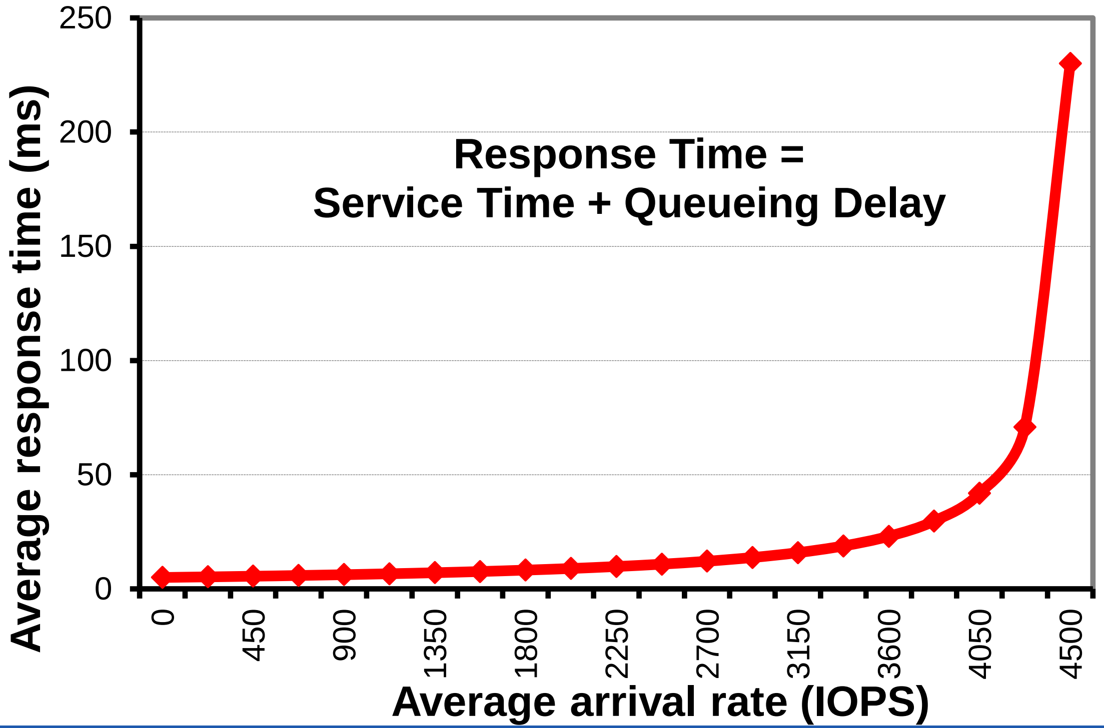
23 SAS disks
JBOD & ASM

4 consecutive tests with increasing load

Lesson learned: don’t accept IOPS numbers without latency values

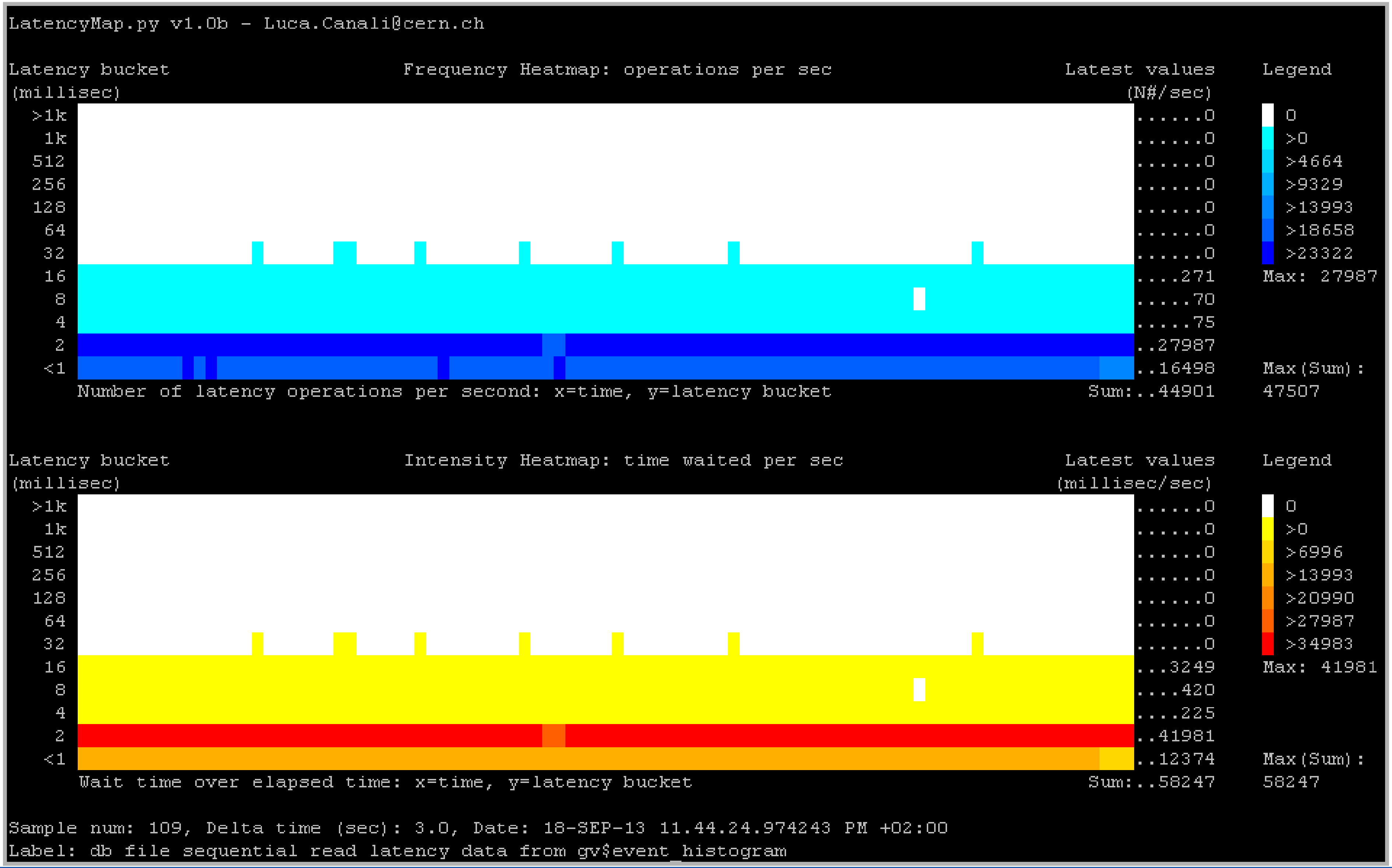
IOPS and Latency are Related

- Observation: as IOPS approach saturation latency increases fast
- Confirmed: a simple model from queueing theory:

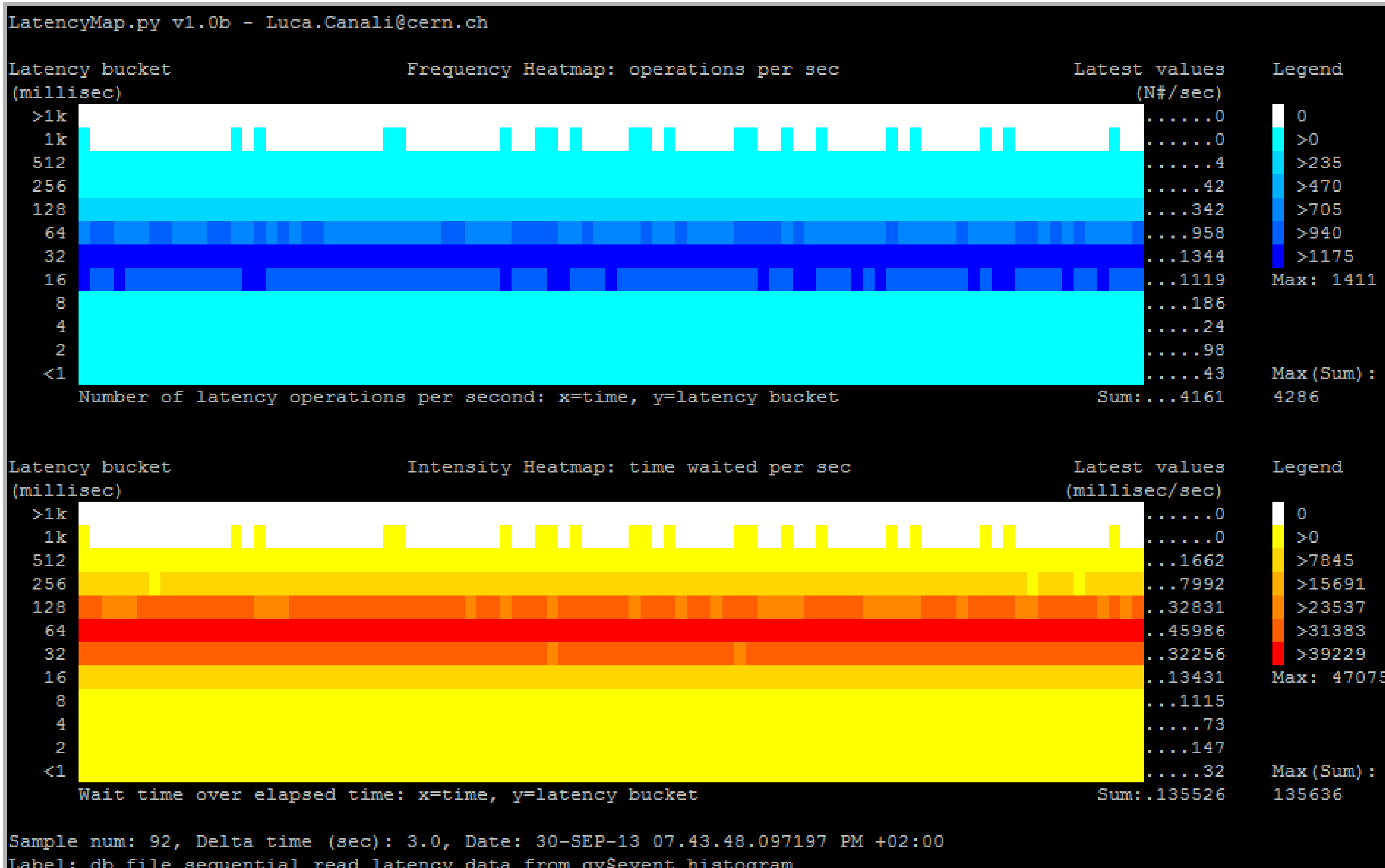


Calculation performed using MMm Multiserver model By Cary Millsap, 2003

0.5 TB dataset, 100% in SSD, 56 sessions, random reads - NAS system



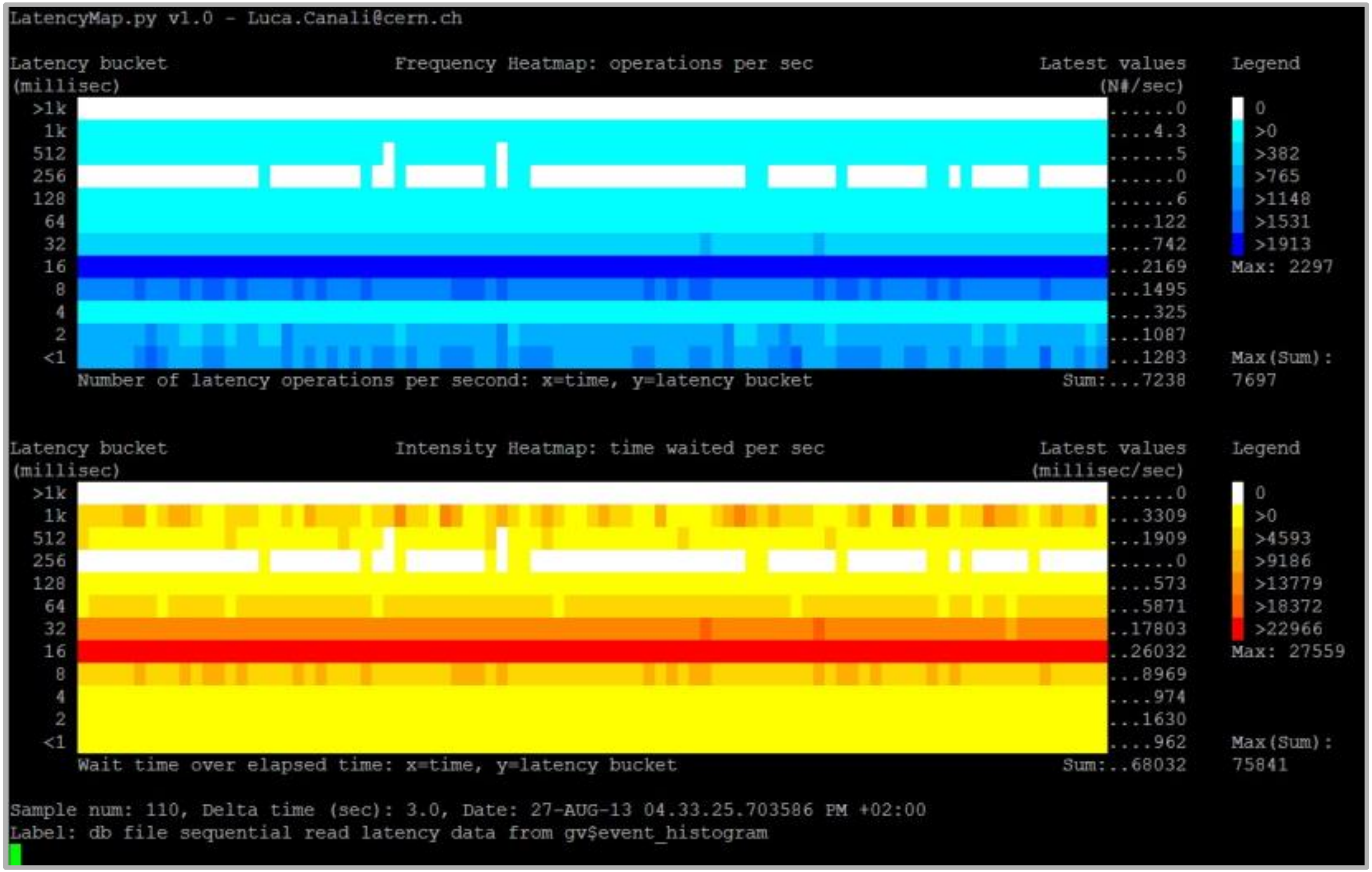
10TB dataset, 128 sessions, random reads, disk saturation - NAS system



Monitoring Production Systems

- Understand I/O response time
  - Help for tuning and capacity planning
  - Attack questions like: is the storage slow?
- Drill down on three areas:
  - I/O served by SSD/controller cache
  - I/O served by physical disk ‘spindles’
  - I/O with very large latency: outliers

An Example of a Busy System

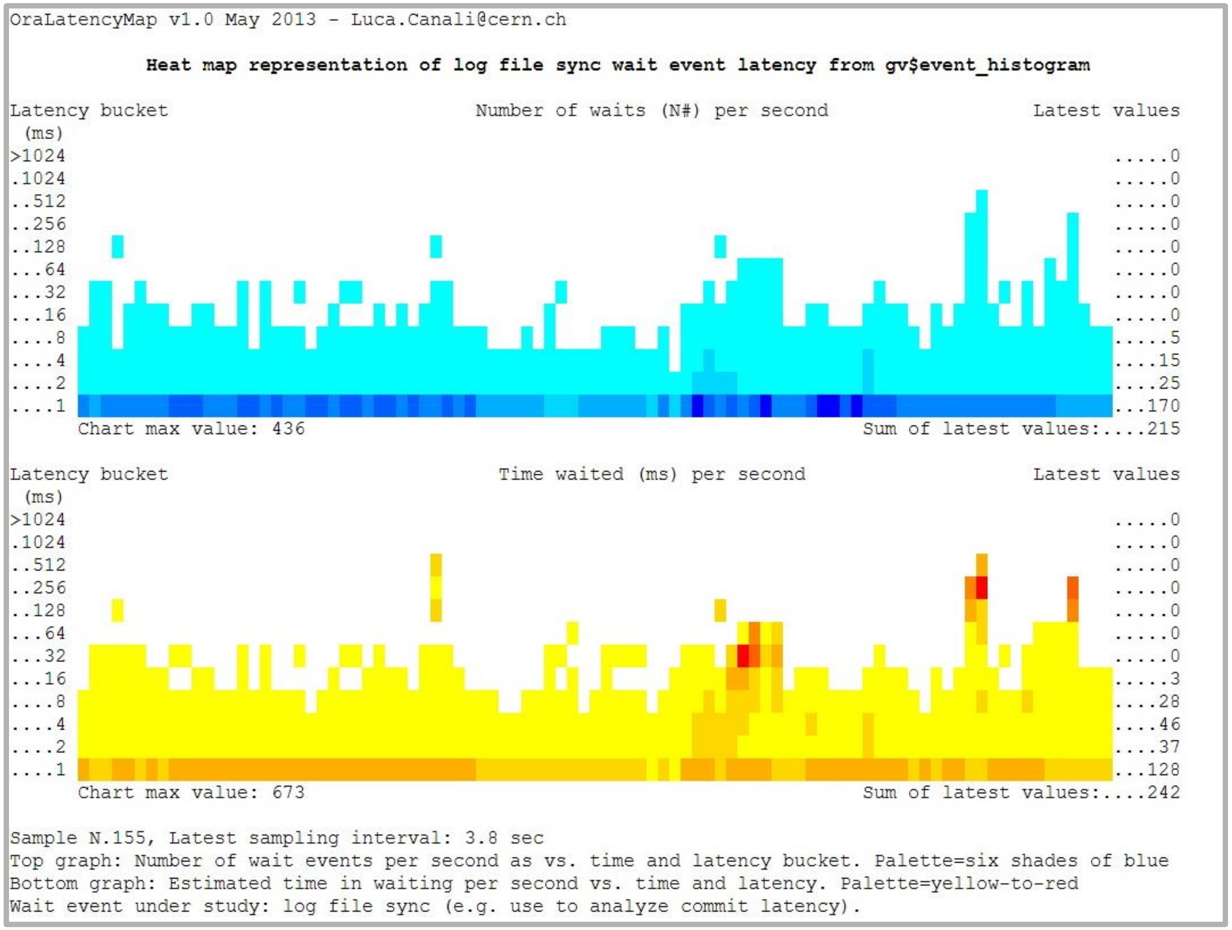


What Can We Learn?

- Example of analysis
  - i.e. drill down ‘db file sequential read’
- Are disks close to saturation?
  - NO, but latency high (SATA disks)
- I/O outliers?
  - YES, Further investigation on controller needed
- Do we have SSD/cache?
  - YES, ~30% reads with low latency
  - We could profit from a larger SSD cache maybe?

Log File Sync

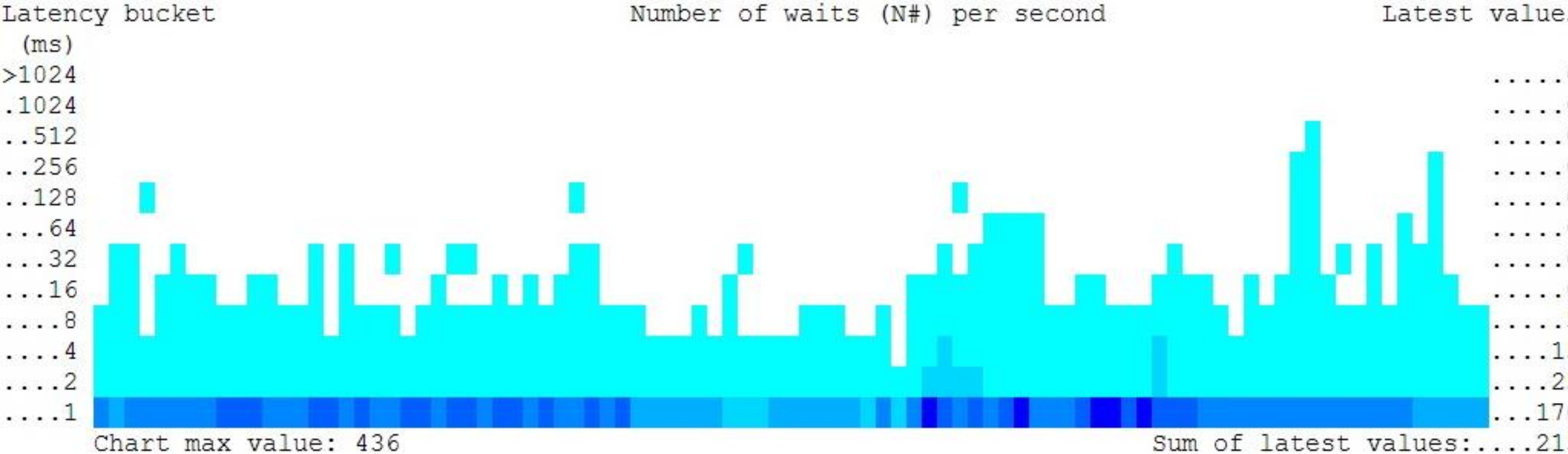
- Example from a production system

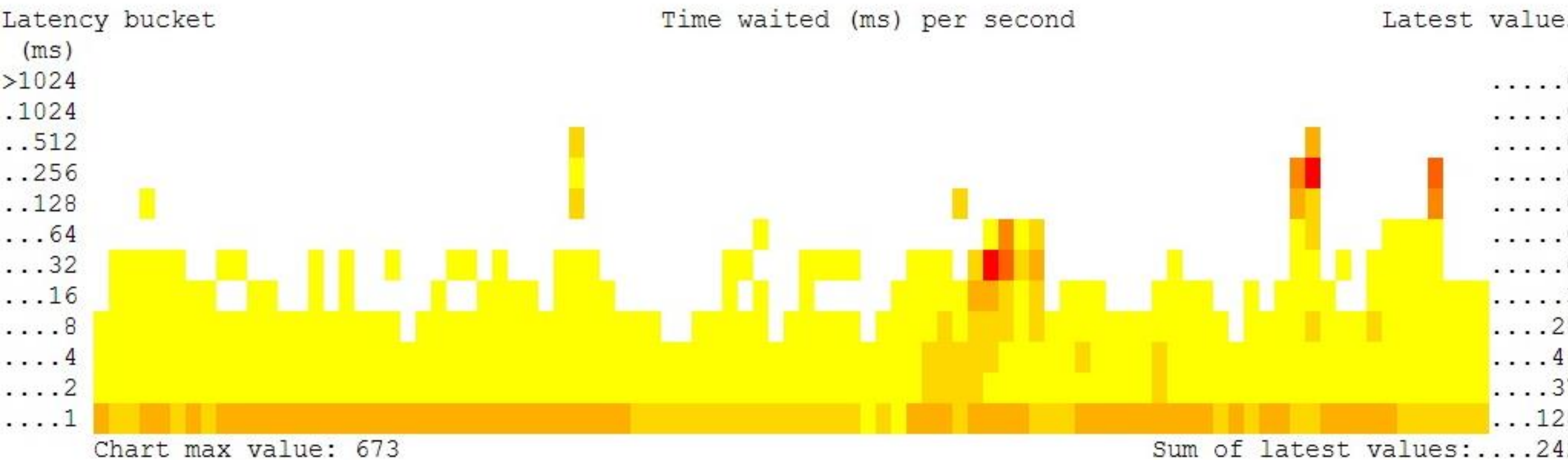


Low latency from writes because of storage cache

OraLatencyMap v1.0 May 2013 - Luca.Canali@cern.ch

Heat map representation of log file sync wait event latency from gv$event_histogram

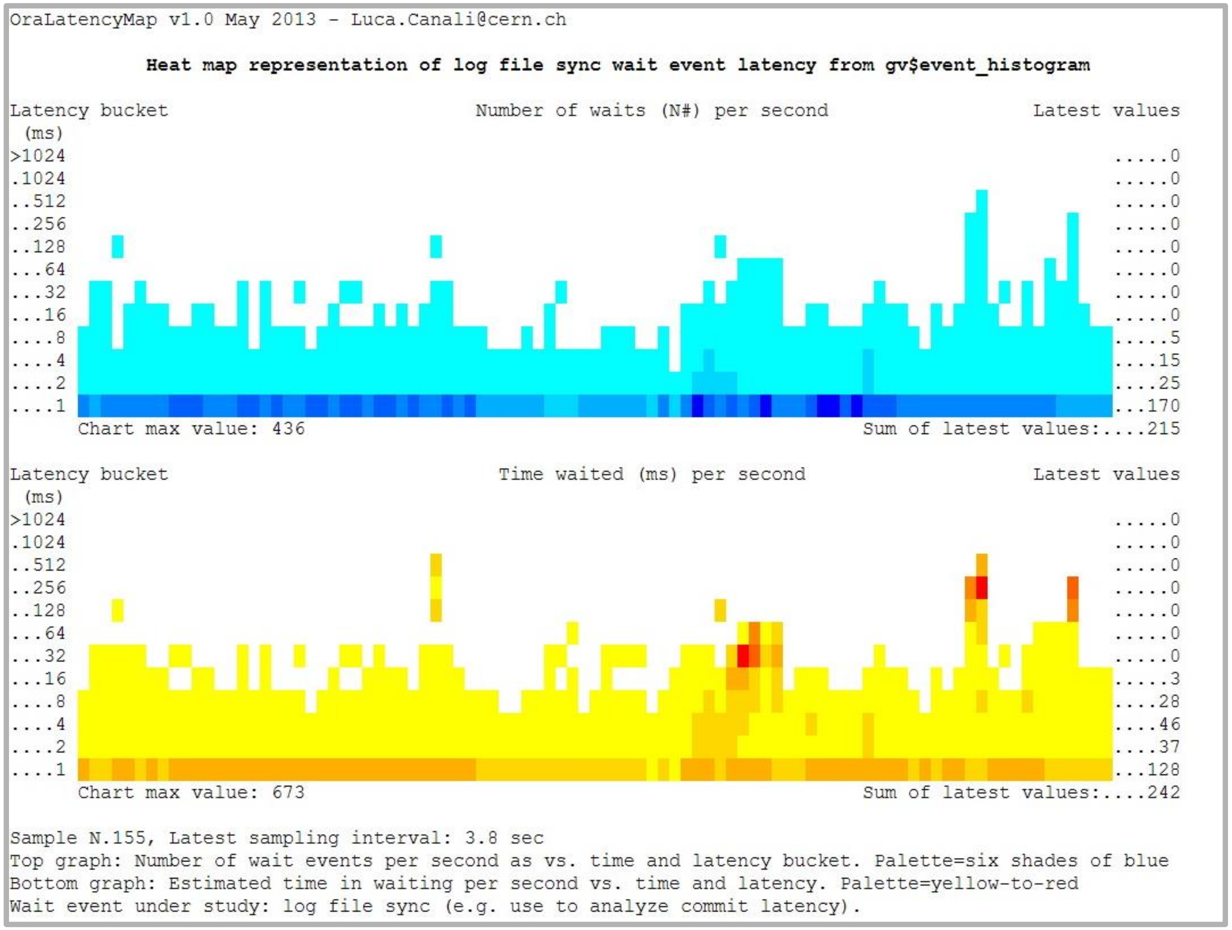




Sample N.155, Latest sampling interval: 3.8 sec
Top graph: Number of wait events per second as vs. time and latency bucket. Palette=six shades of blue
Bottom graph: Estimated time in waiting per second vs. time and latency. Palette=yellow-to-red
Wait event under study: log file sync (e.g. use to analyze commit latency).

Log File Sync

- Example from a production system

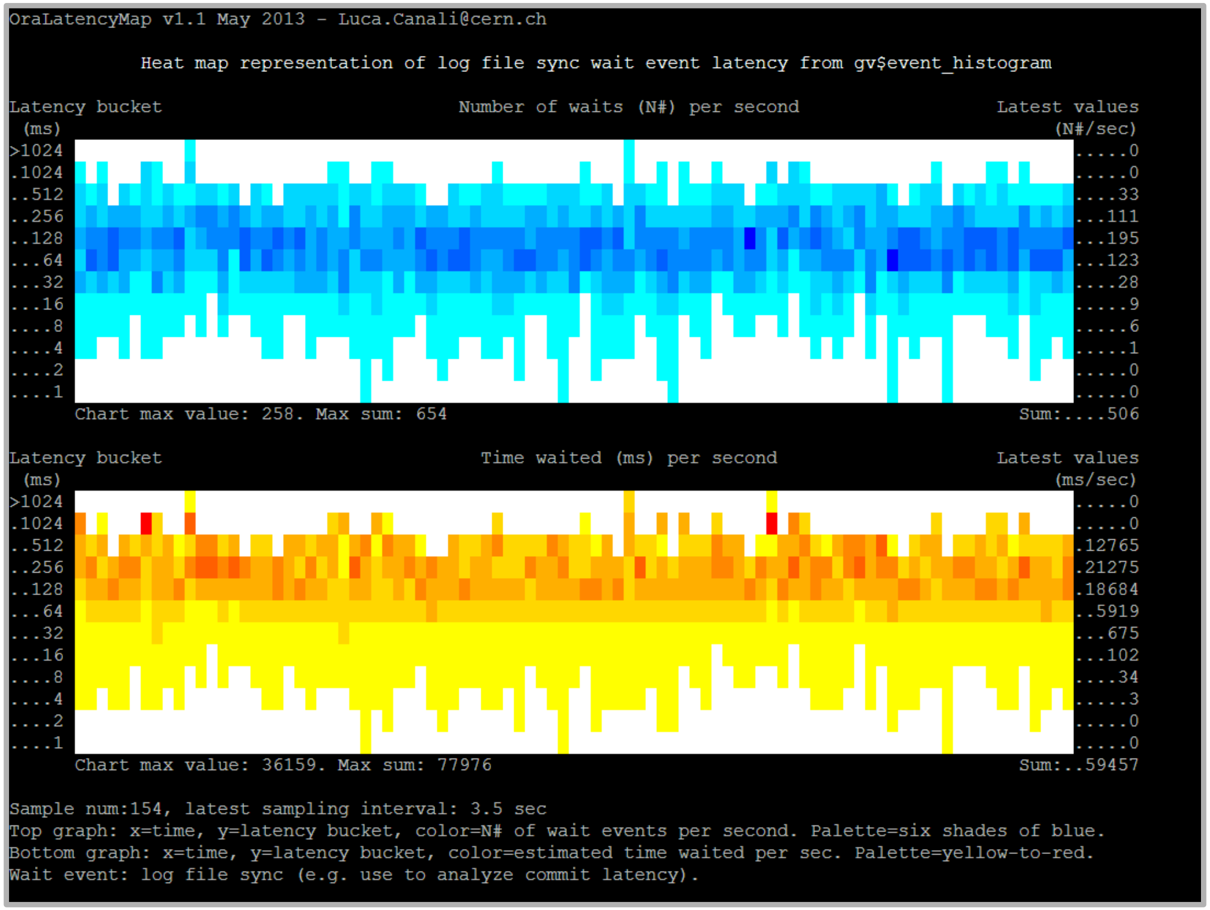


Low latency from writes because of storage cache

Log File Sync

- Anomaly, on a test system

High latency caused by HW Issues and high load from Oracle



OraLatencyMap v1.1 May 2013 - Luca.Canali@cern.ch

Heat map representation of log file sync wait event latency from gv$event_histogram

Latency bucket (ms) — Number of waits (N#) per second — Latest values (N#/sec)

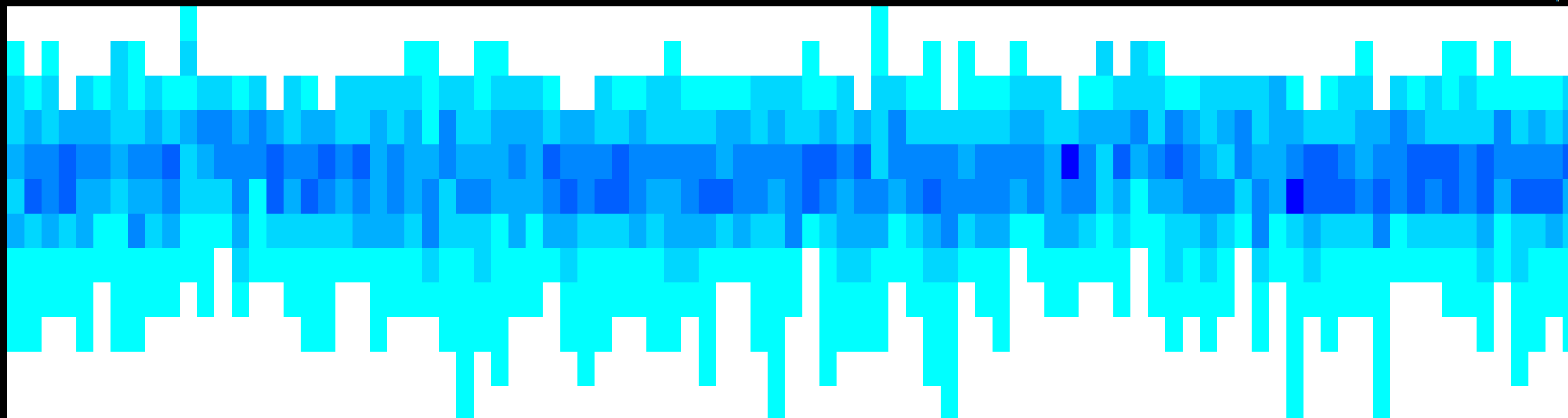


Chart max value: 258. Max sum: 654 Sum:....506

Latency bucket (ms) — Time waited (ms) per second — Latest values (ms/sec)

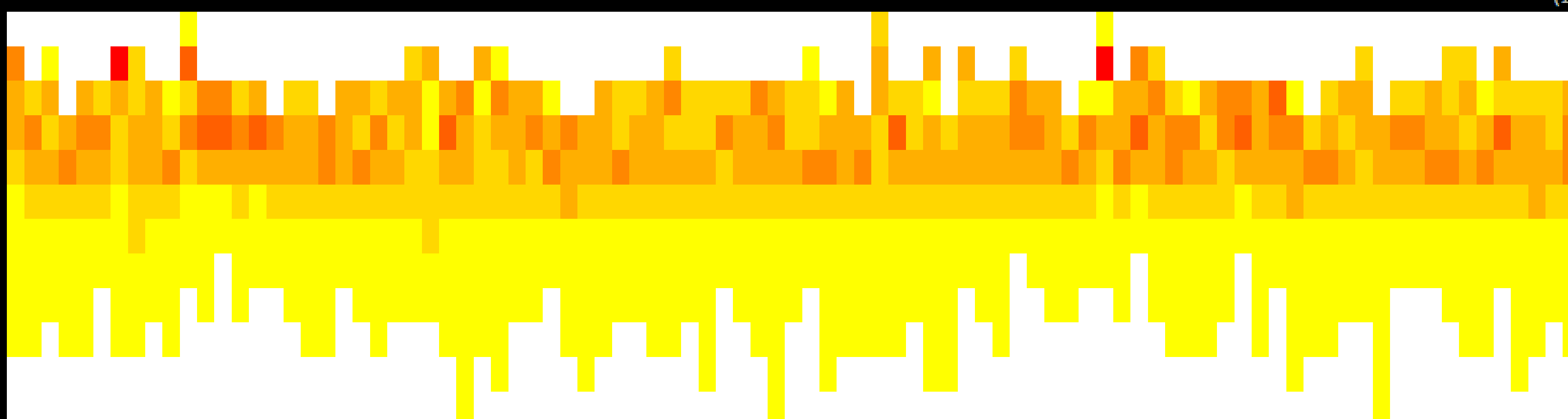


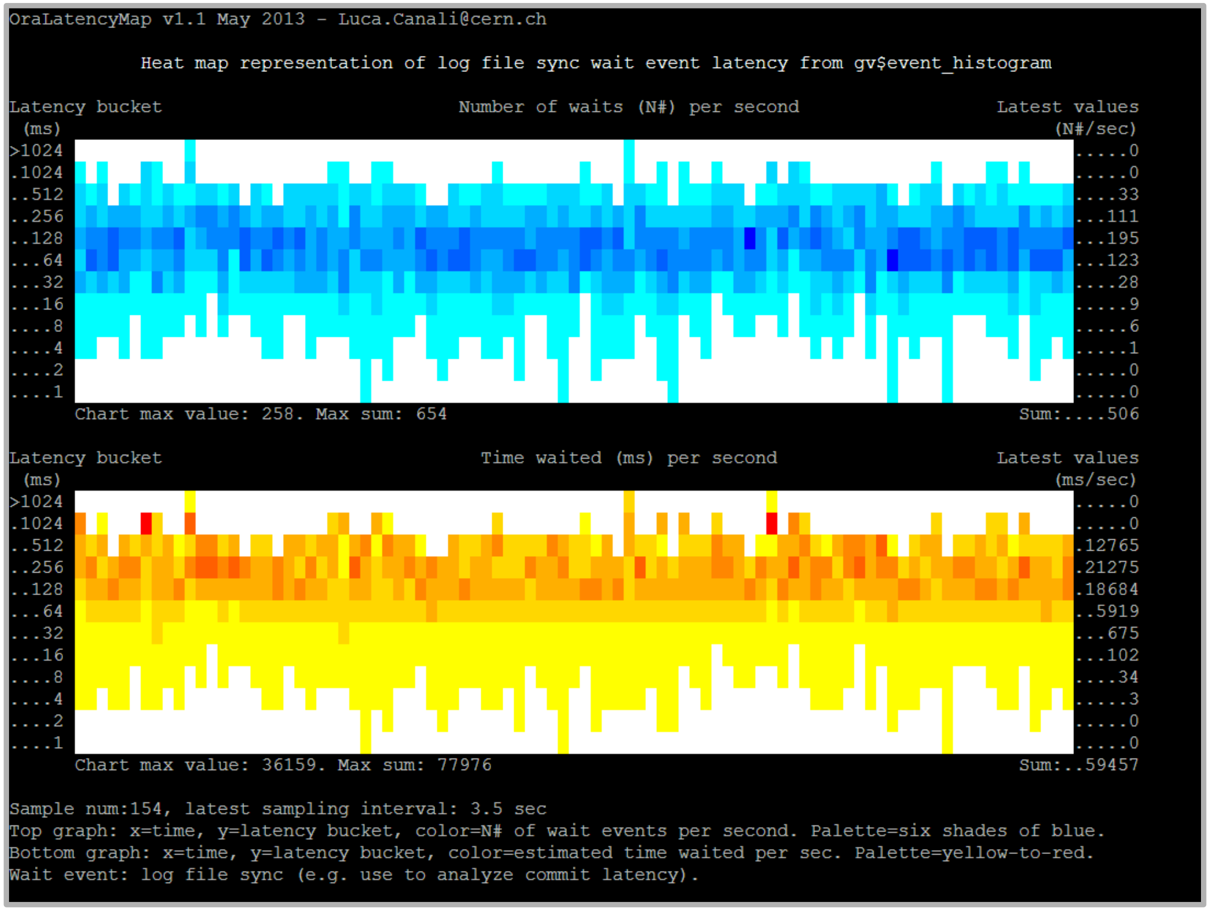
Chart max value: 36159. Max sum: 77976 Sum:..59457

Sample num:154, latest sampling interval: 3.5 sec
Top graph: x=time, y=latency bucket, color=N# of wait events per second. Palette=six shades of blue.
Bottom graph: x=time, y=latency bucket, color=estimated time waited per sec. Palette=yellow-to-red.
Wait event: log file sync (e.g. use to analyze commit latency).

Log File Sync

- Anomaly, on a test system

High latency caused by HW Issues and high load from Oracle



Limitations

- Wait event timing is done by Oracle code
  - May not reflect actual I/O service time
  - CPU time may also be accounted as I/O wait
  - Server high load can distort the timing values
  - V$event_histogram has only milli sec precision

- Asynchronous I/O
  - Wait events of this family are very hard or impossible to utilize in the context of latency

Blocking Calls

- Db file sequential read
  - Is the easiest event to relate to I/O service time
  - Instruments single-block reads, blocking I/O
    - *Note: in some cases Oracle can use async I/O for random reads, e.g. for prefetching and batching. Wait event used in that case is 'db file parallel read'*

- Log file sync
  - Big part of the commit-time wait
  - Complex: it's not a pure I/O event
  - The root causes of high latency here can also be CPU starvation and LGWR behaviour (e.g. bugs)

Latency and Trace Files

- Latency data is available in 10046 trace files
  - With micro second precision
  - Allows drill down to session level
    - As opposed to using global GV$ views

```
SQL> exec dbms_monitor.session_trace_enable(sid,serial#)
```

```
nam='db file sequential read' ela= 977 file#=7
block#=29618220 blocks=1 obj#=82015 tim=1377520823036634
```

More Latency Sources

- DTrace
  - Great performance tool, coming to Linux too
  - Can be used to gather I/O latency histograms
    - Use of the quantize operator

```
dtrace -n '
syscall::pread64:entry { self->s = timestamp; }
syscall::pread64:return /self->s/ { @pread["ns"] =
quantize(timestamp -self->s); self->s = 0; }

tick-10s {
printa(@pread);
trunc(@pread);
}'
```

Outline

- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency in Oracle
- Examples
- Tools
- Conclusions

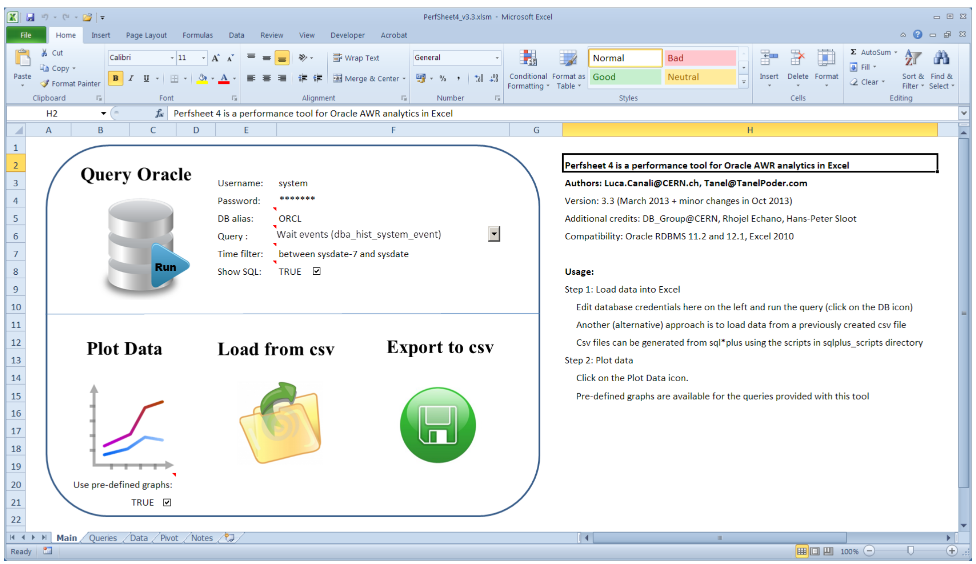
CERN

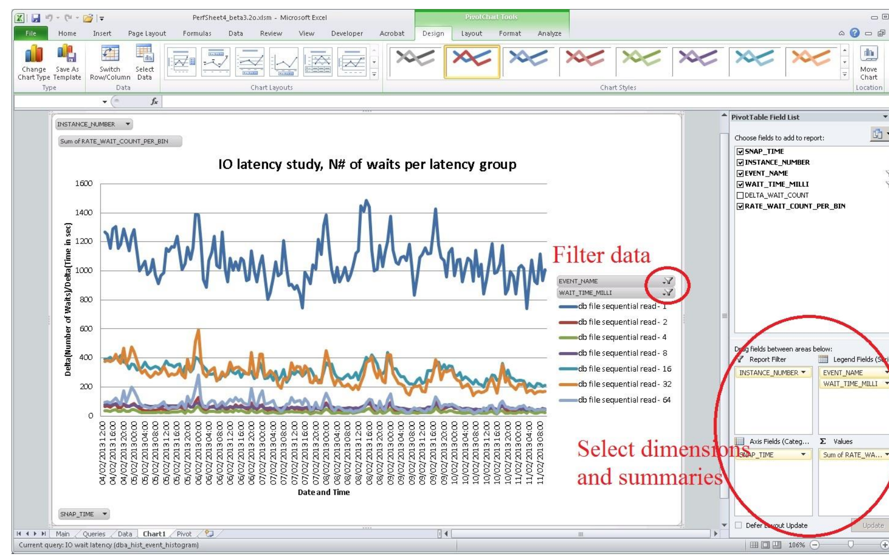
Tools

- Automate tedious tasks
  - Data collection
  - Visualisation
- Provide data and help answer questions
  - Drill down on I/O wait events
- Find trends and evolution
  - How does performance change over time
  - Is it Oracle workload changing or is it the storage that has become slow?

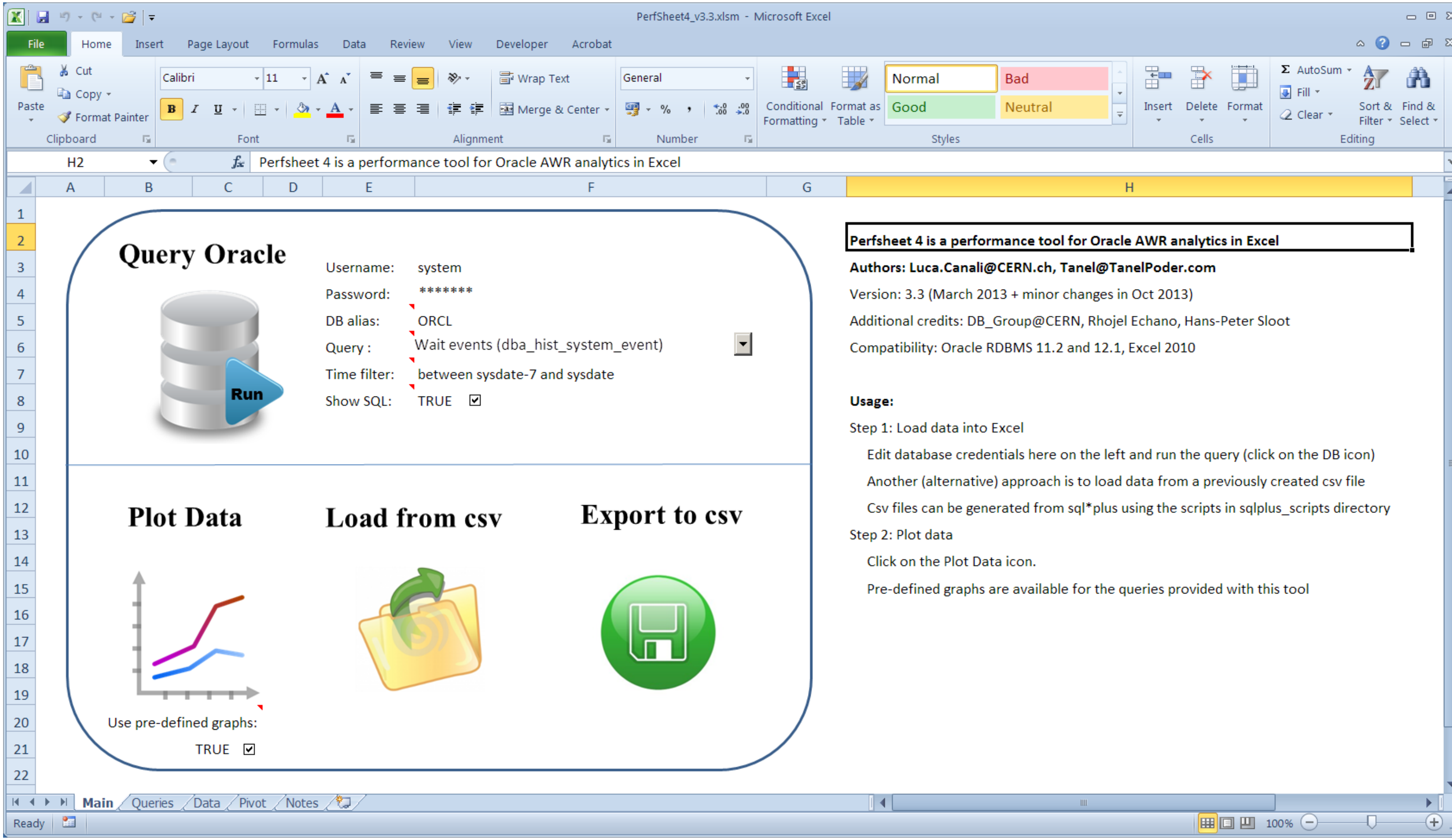
Tools: PerfSheet 4

- Simple Analytic platform for AWR data
- Predefined queries and graphs
- Power of Pivot Charts

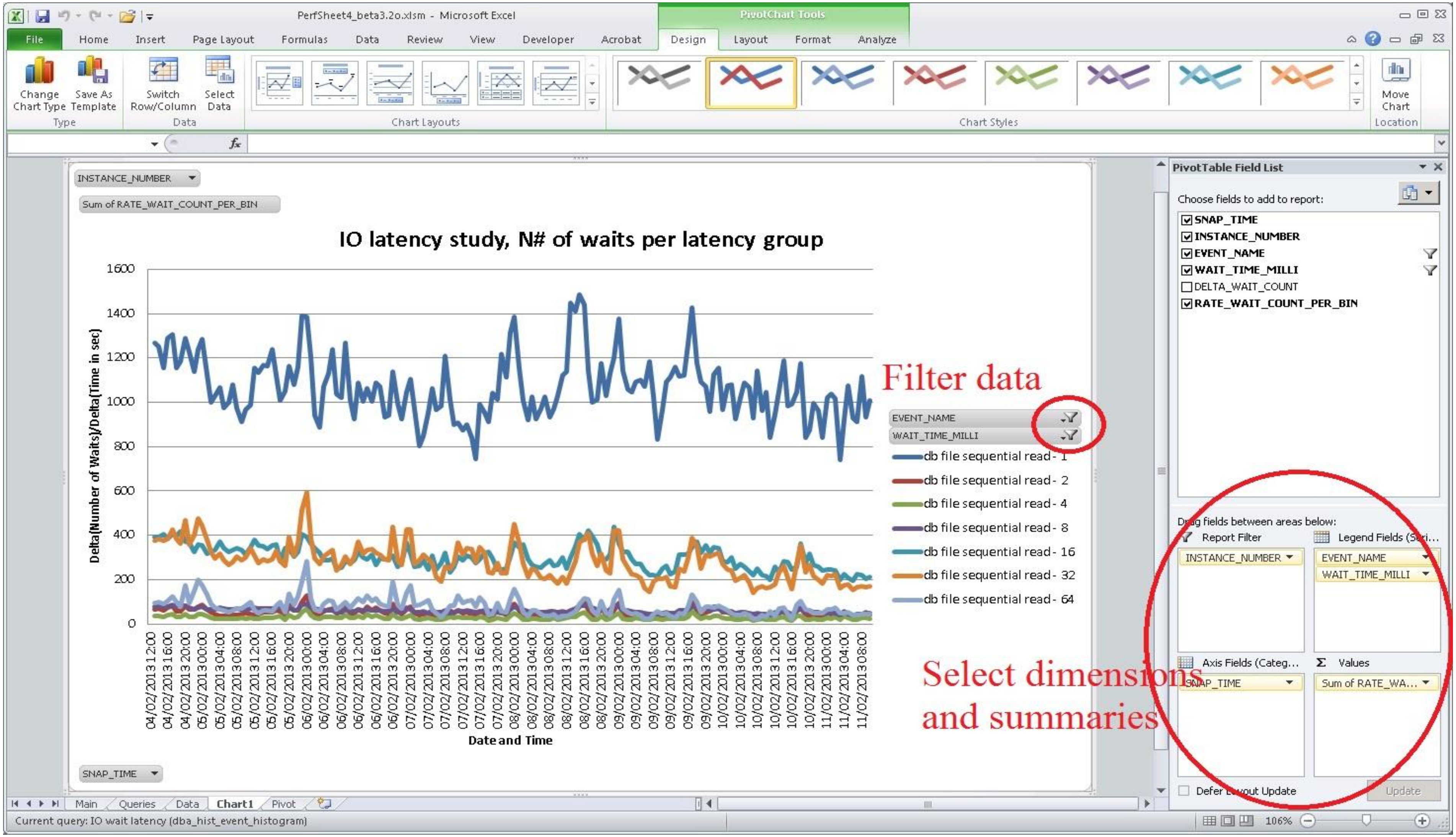




Tools: PerfSheet 4



Tools: PerfSheet 4



Tools: OraLatencyMap

- It's a SQL*Plus script based on PL/SQL
  - Lightweight, does not require any installation
  - Command line interface
  - Heat Maps generated using ANSI escape codes

- Get started:

```
SQL> @OraLatencyMap
```

Tools: PyLatencyMap

- It's written in Python + SQL*Plus scripts
  - No installations required, CLI, lightweight
  - More advanced than OraLatencyMap
- Can be used for generic latency sources
  - Oracle v$, trace files, AWR, DTrace data, etc
  - Pre-built examples available
- Feature: record and replay

Getting Started with PyLatencyMap

- Modular architecture
  - Source | <optional filter> | visualization engine

- Get started

```
./Example1_oracle_random_read.sh
```

- Video, getting started with PyLatencyMap
- http://www.youtube.com/watch?v=-YuShn6ro1g

Outline

- CERN and Oracle
- Latency: what is the problem we are trying to solve
- Storage latency in Oracle
- Examples
- Tools
- Conclusions

Conclusions

- Analysis of I/O latency
  - A powerful technique in performance tuning
  - Latency average is not enough, need histograms
  - Oracle wait interface has histogram details
- PylatencyMap for data collection and visualisation
  - for Oracle and generic data sources
  - http://cern.ch/canali/resources.htm
- Latency heat maps are great!

Acknowledgements

- Our colleagues in the CERN Database Group
  - In particular: Ruben Gaspar
- Many ideas borrowed from blogs and articles:
  - Brendan Gregg, Tanel Poder, Kevin Closson, Frits Hoogland, Marcin Przepiorowski, James Morles, Kyle Hailey, Cary Millsap

Thank you!

Luca.Canali@cern.ch

Marcin.Blaszczyk@cern.ch

www.cern.ch