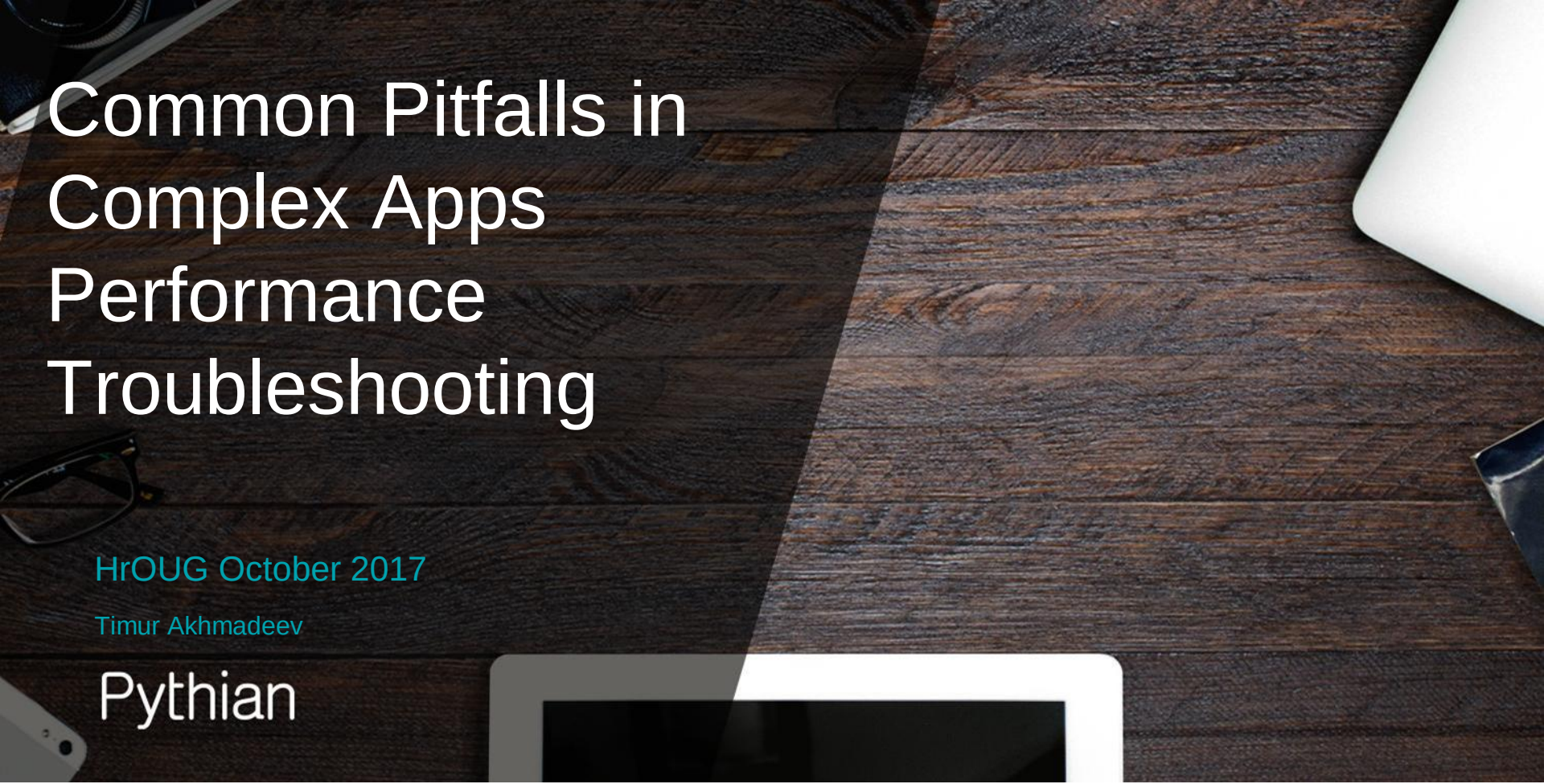


Common Pitfalls in Complex Apps Performance Troubleshooting

HrOUG October 2017

Timur Akhmadeev

Pythian

ABOUT PYTHIAN

Pythian’s 400+ IT professionals help companies adopt and manage disruptive technologies to better compete

About Me (Short Version)

DBA who was a Developer

About Me (Long version)

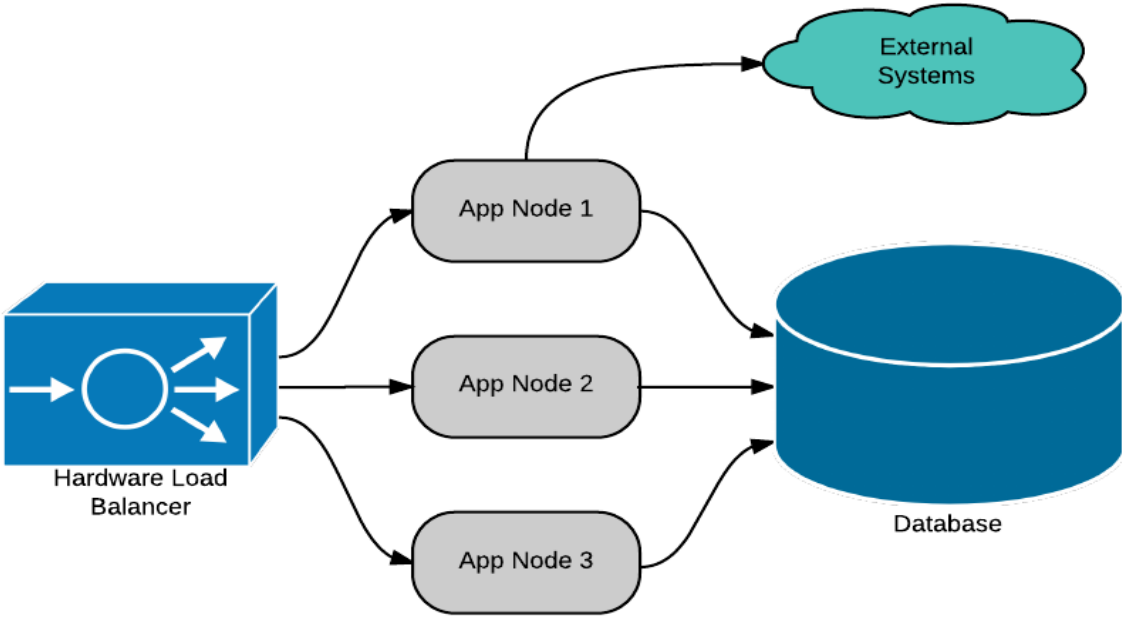
Dev	Perf	DBA

Database Consultant at Pythian
12+ years with Database and Java
Systems Performance and Architecture
OakTable member
timurakhmadeev.wordpress.com
pythian.com/blog/author/akhmadeev
twitter.com/tmmdv
timur.akhmadeev@gmail.com

Agenda

- Complex apps architecture
- What usually happens when a performance issue is reported
- Capacity vs Performance
- What should happen

Architecture



Architecture

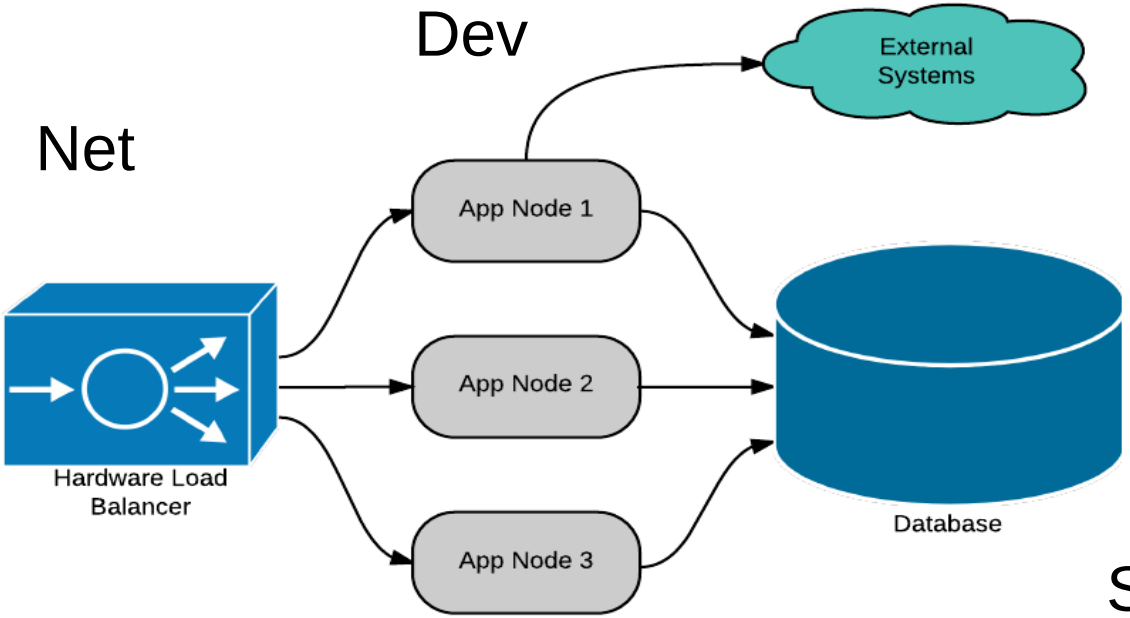
NOC

Dev

Net

DBA

Storage



Response #1

I've checked application & database, load is low, under 5-7%

Response #2

IOPS is good

Response #3

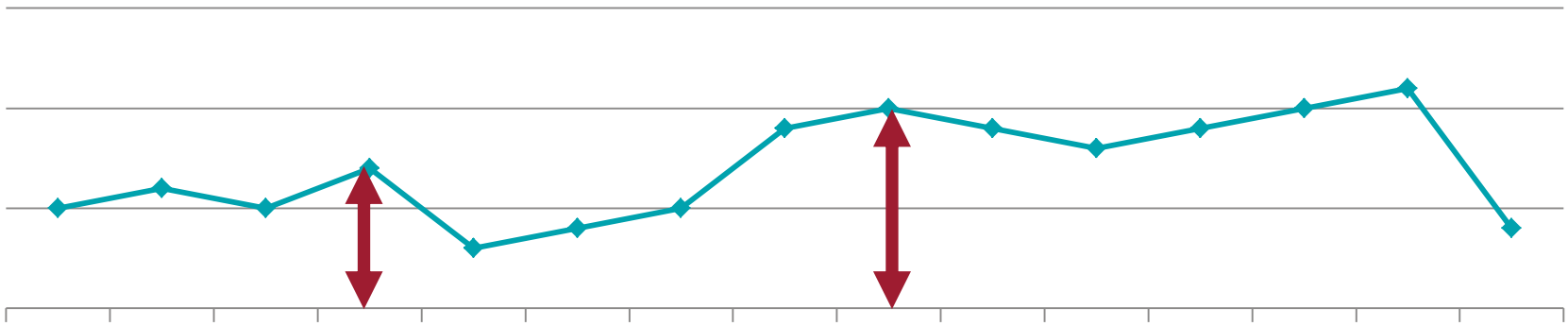
Application server was unresponsive so I've restarted it & all is good now

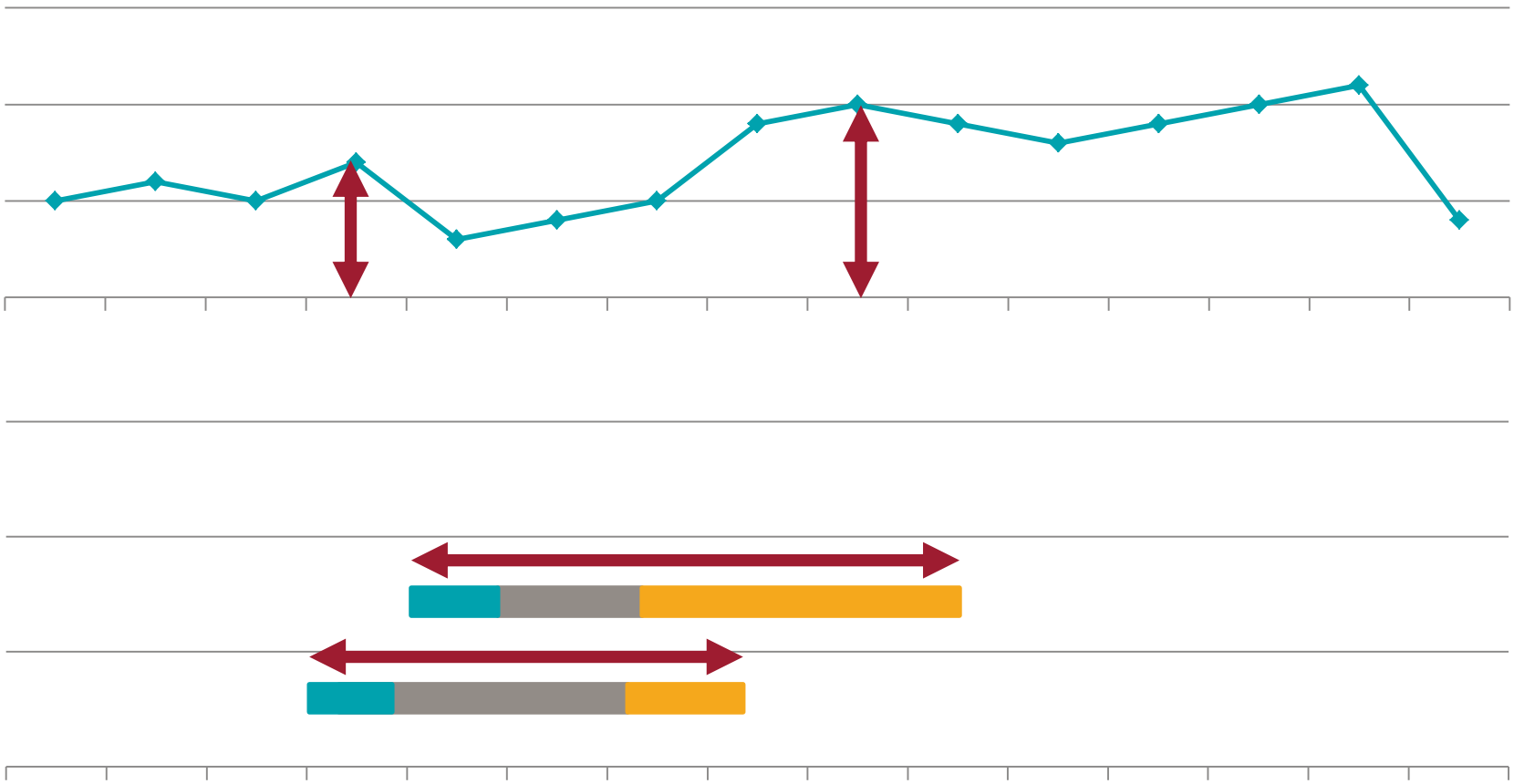
Response #4

We need to increase the number of app nodes

What’s Wrong

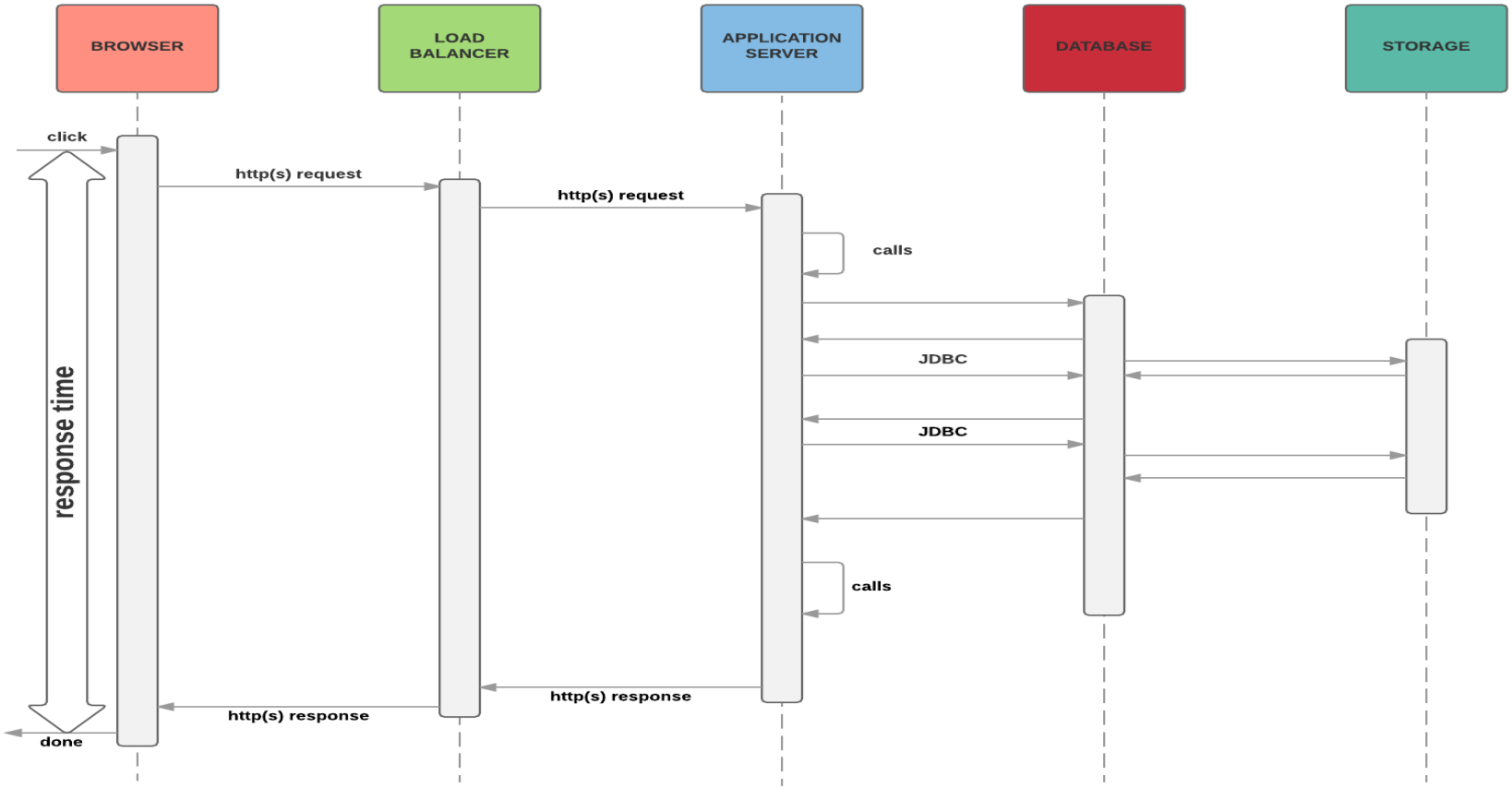
data

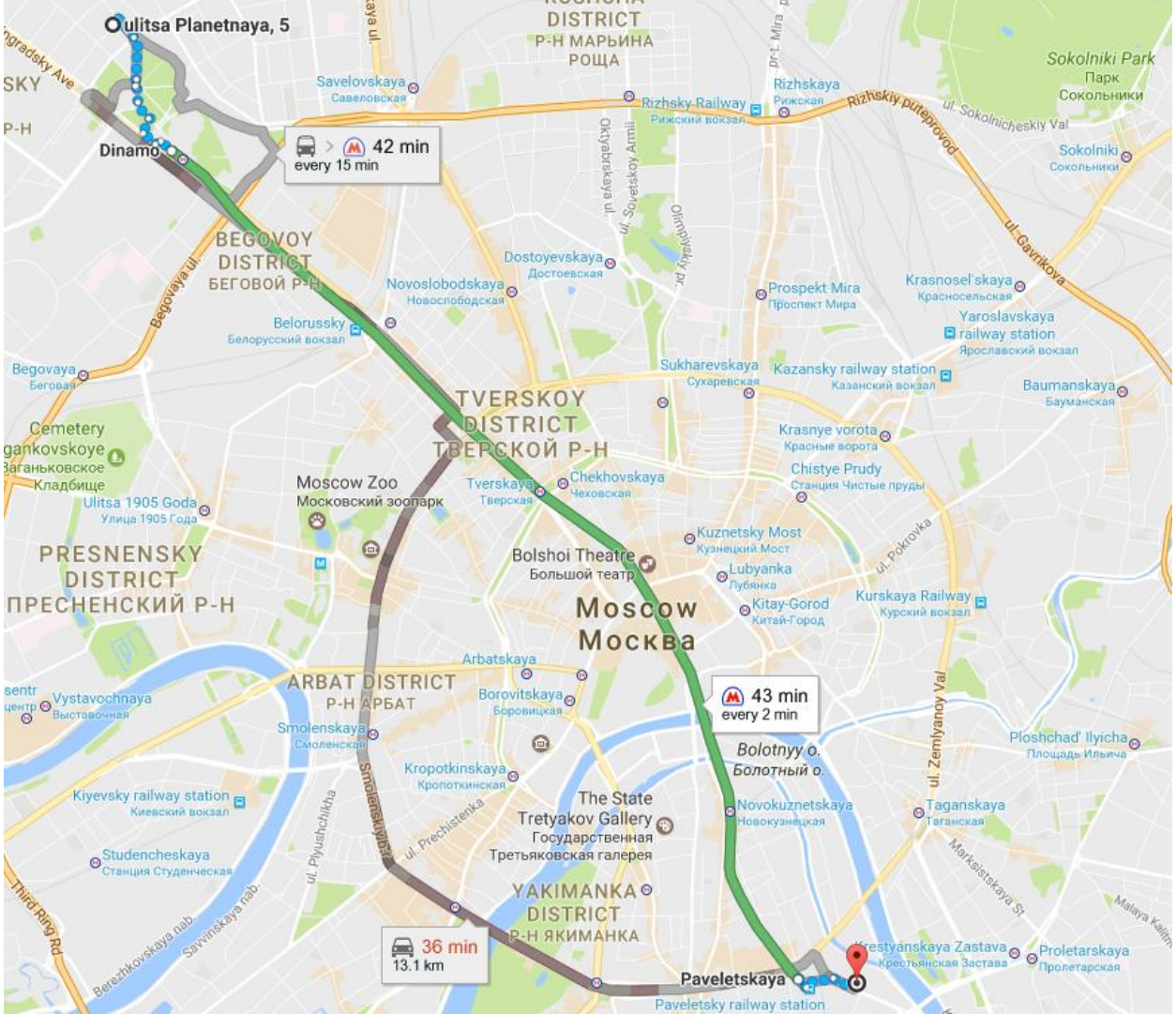


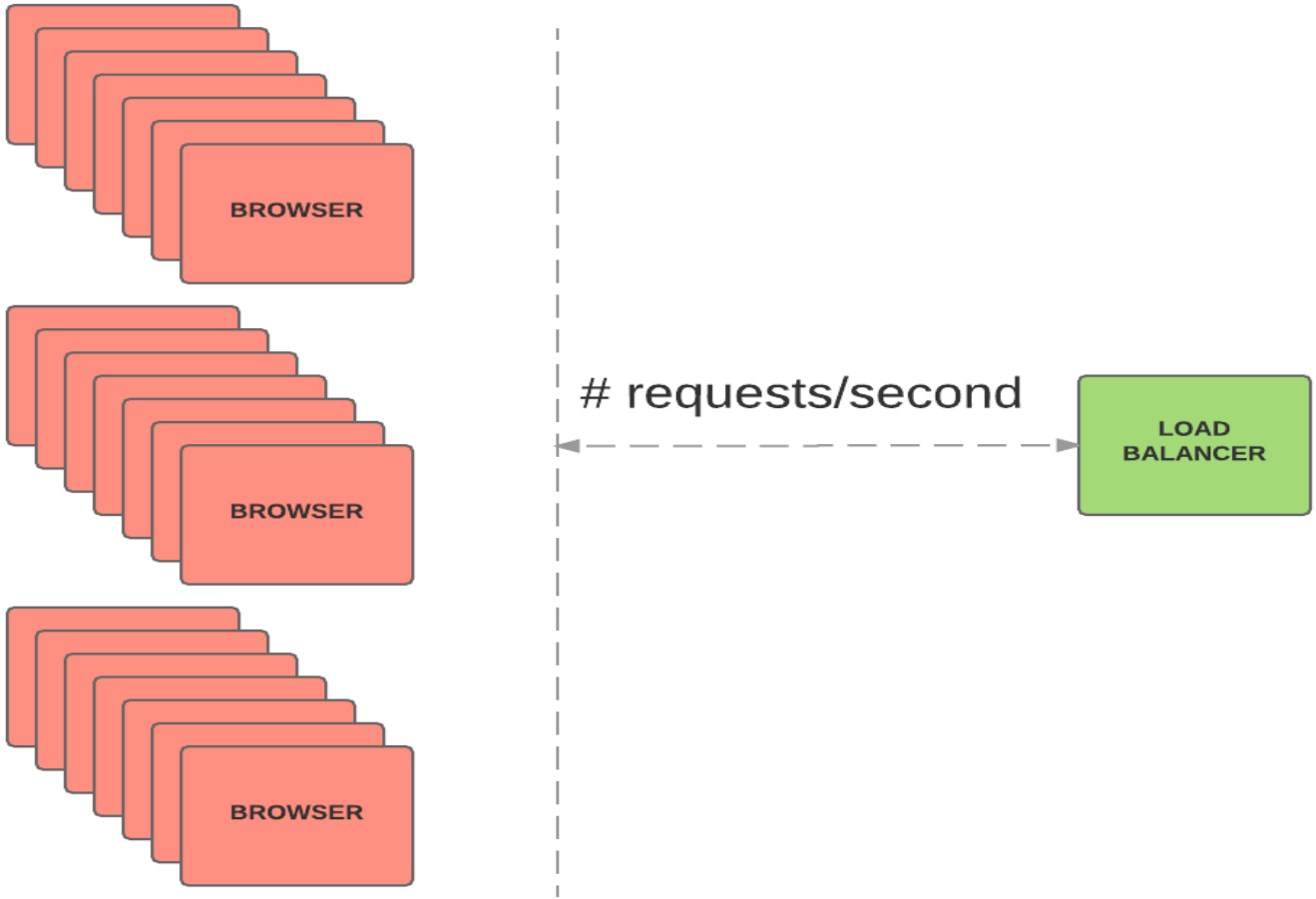


	Good Performance	Bad Performance
Metric is low	✓	✓
Metric is high	✓	✓

	Good Performance	Bad Performance
5 seconds	✗	✓
45 seconds	✓	✗


BROWSER
LOAD BALANCER
APPLICATION SERVER
DATABASE
STORAGE
click
http(s) request
http(s) request
calls
JDBC
JDBC
calls
http(s) response
http(s) response
done
response time


ulitsa Planetnaya, 5
Dinamo
42 min
every 15 min
Savelovskaya
Савеловская
Rizhskaya
Рижская
Rizhsky Railway
Рижский вокзал
Rizhskiy puteprovod
ul. Sokolnicheskiy Val
Sokolniki Park
Парк Сокольники
Sokolniki
Сокольники
ul. Gavrikova
BEGOVOY DISTRICT
БЕГОВОЙ Р-Н
Begovaya ul.
Dostoyevskaya
Достоевская
Novoslobodskaya
Новослободская
Belorussky
Белорусский вокзал
Prospekt Mira
Проспект Мира
Krasnosel'skaya
Красносельская
Yaroslavskaya railway station
Ярославский вокзал
Begovaya
Беговая
Sukharevskaya
Сухаревская
Kazansky railway station
Казанский вокзал
Baumanskaya
Бауманская
TVERSKOY DISTRICT
ТВЕРСКОЙ Р-Н
Cemetery
Ваганьковское Кладбище
Krasnye vorota
Красные ворота
Chistye Prudy
Станция Чистые пруды
Moscow Zoo
Московский зоопарк
Tverskaya
Тверская
Chekhovskaya
Чеховская
Ulitsa 1905 Goda
Улица 1905 года
Kuznetsky Most
Кузнецкий Мост
PRESNENSKY DISTRICT
ПРЕСНЕНСКИЙ Р-Н
Bolshoi Theatre
Большой театр
Lubyanka
Лубянка
ul. Pokrovka
Kitay-Gorod
Китай-Город
Kurskaya Railway
Курский вокзал
Moscow
Москва
Arbatskaya
ARBAT DISTRICT
Р-Н АРБАТ
Borovitskaya
Боровицкая
43 min
every 2 min
Vystavochnaya
Выставочная
Smolenskaya
Смоленская
Bolotnyy o.
Болотный о.
Ploshchad' Ilyicha
Площадь Ильича
Kropotkinskaya
Кропоткинская
Kiyevsky railway station
Киевский вокзал
The State Tretyakov Gallery
Государственная Третьяковская галерея
Novokuznetskaya
Новокузнецкая
Taganskaya
Таганская
ul. Zemlyanoy Val
Studencheskaya
Станция Студенческая
ul. Prechistenka
YAKIMANKA DISTRICT
Р-Н ЯКИМАНКА
Third Ring Rd
Berezhkovskaya nab.
Savvinskaya nab.
36 min
13.1 km
Paveletskaya
Paveletsky railway station
Krestyanskaya Zastava
Крестьянская Застава
Proletarskaya
Пролетарская


BROWSER
BROWSER
BROWSER
requests/second
LOAD BALANCER

Tools to “measure” Performance

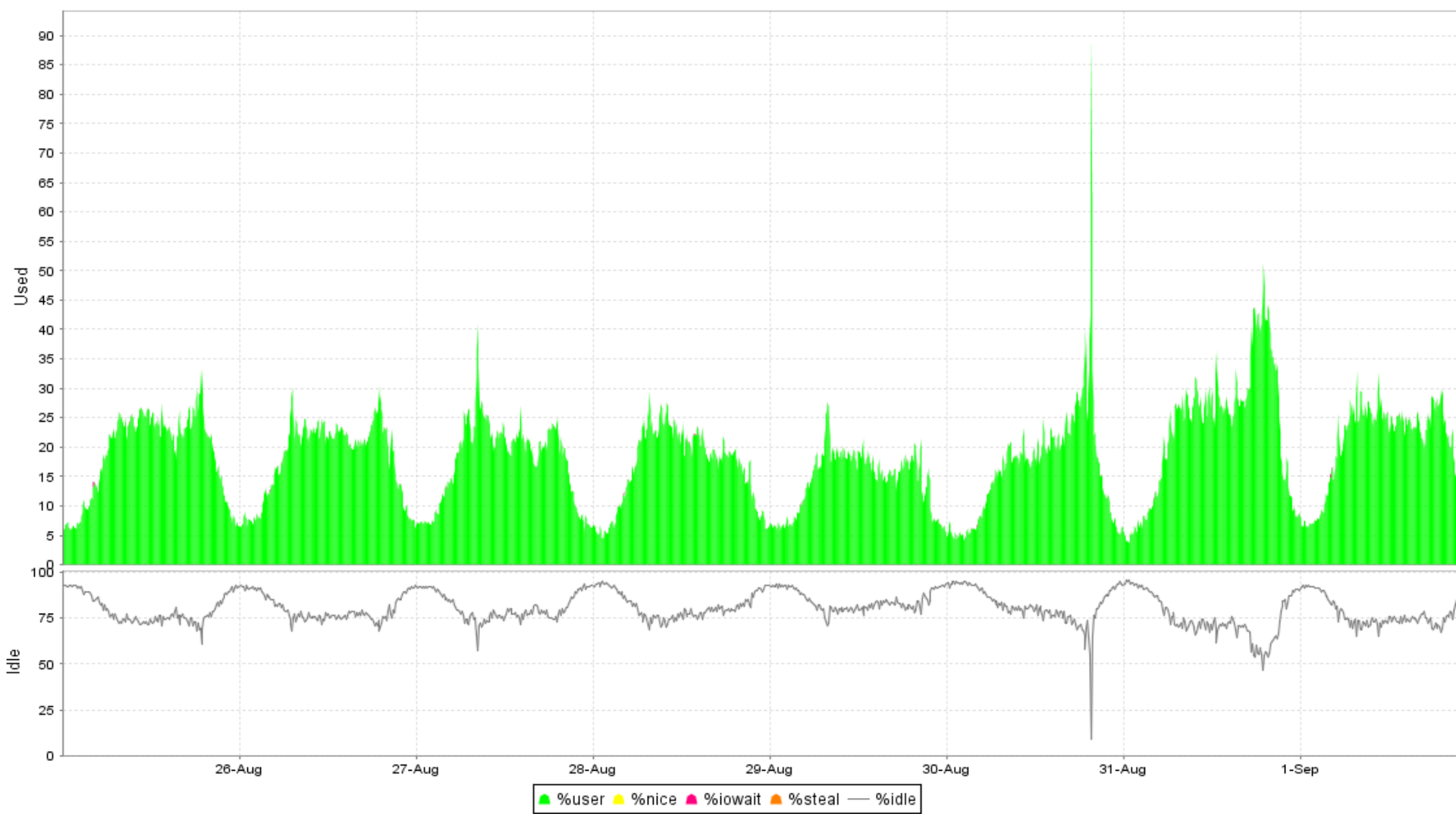
```
top
htop
nmon
```

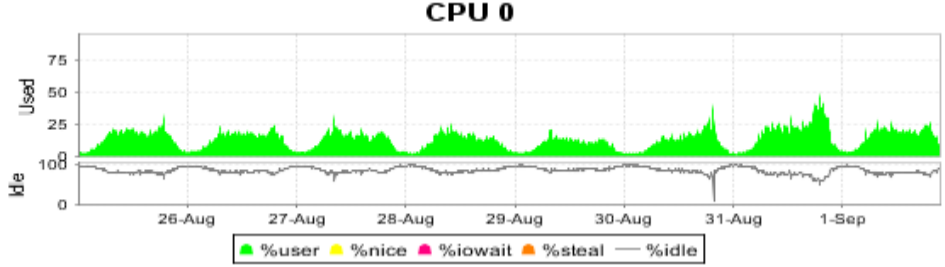
```
iostat
vmstat
sar
collectl
```

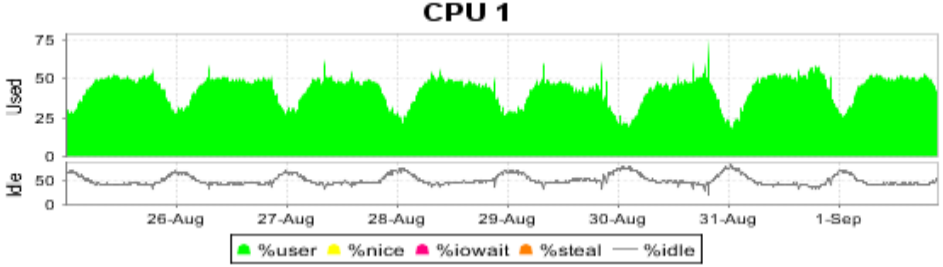
Tools to “measure” Performance

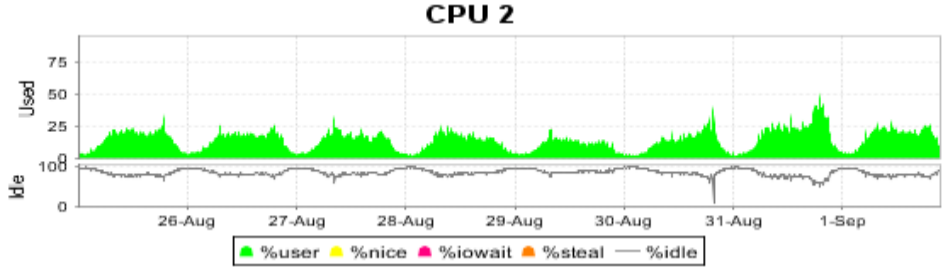
```
top
htop =
nmon
```

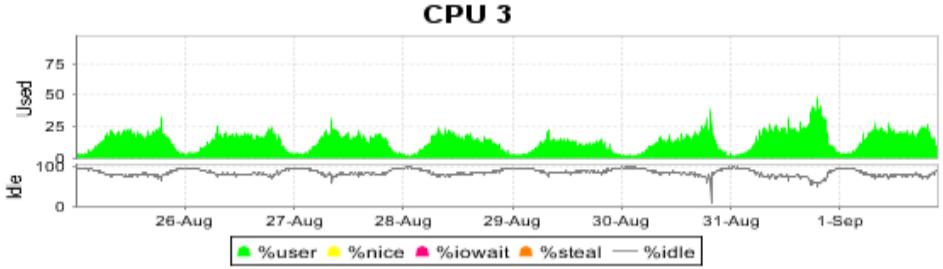
how many / fast /
big / loud / much
fuel/km / people
inside of cars
passing by a corner

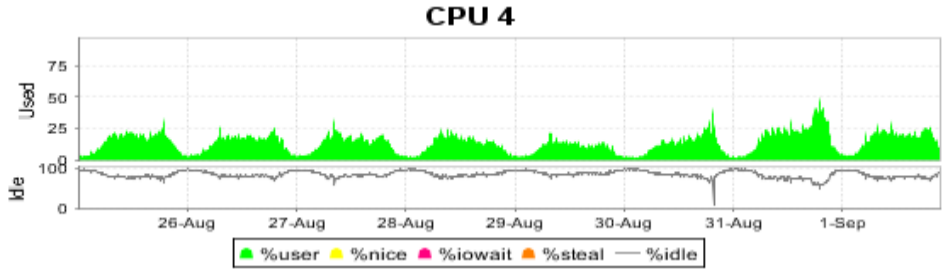

Used
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Idle
100
75
50
25
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

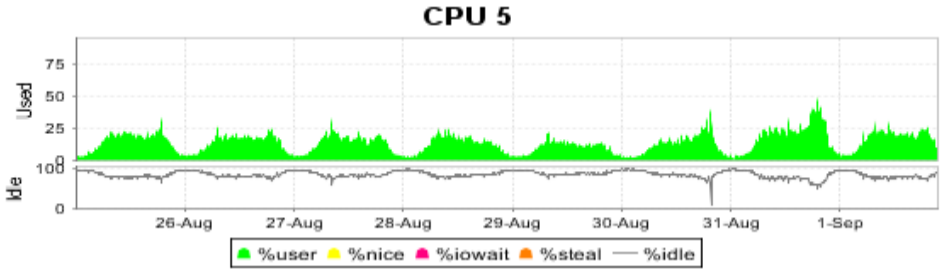

CPU 0
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

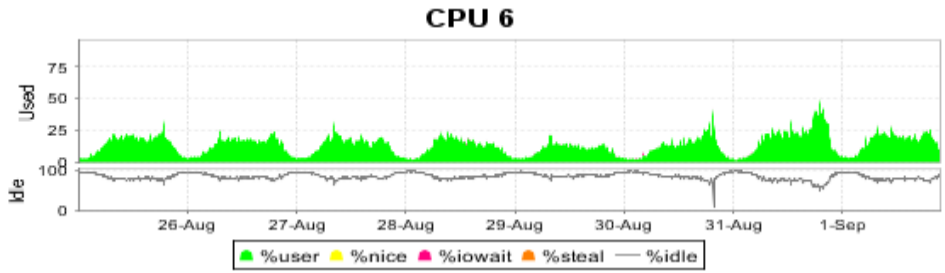

CPU 1
Used
Idle
75
50
25
0
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

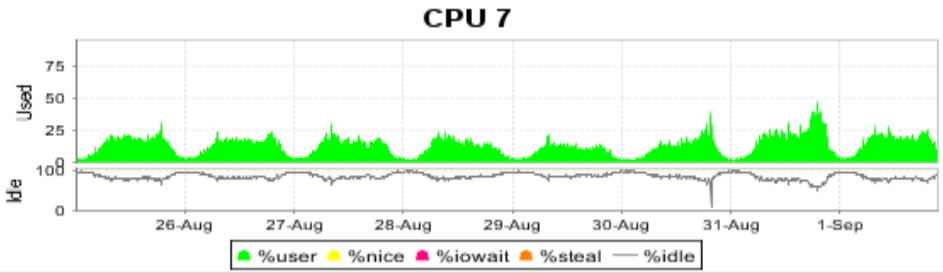

CPU 2
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

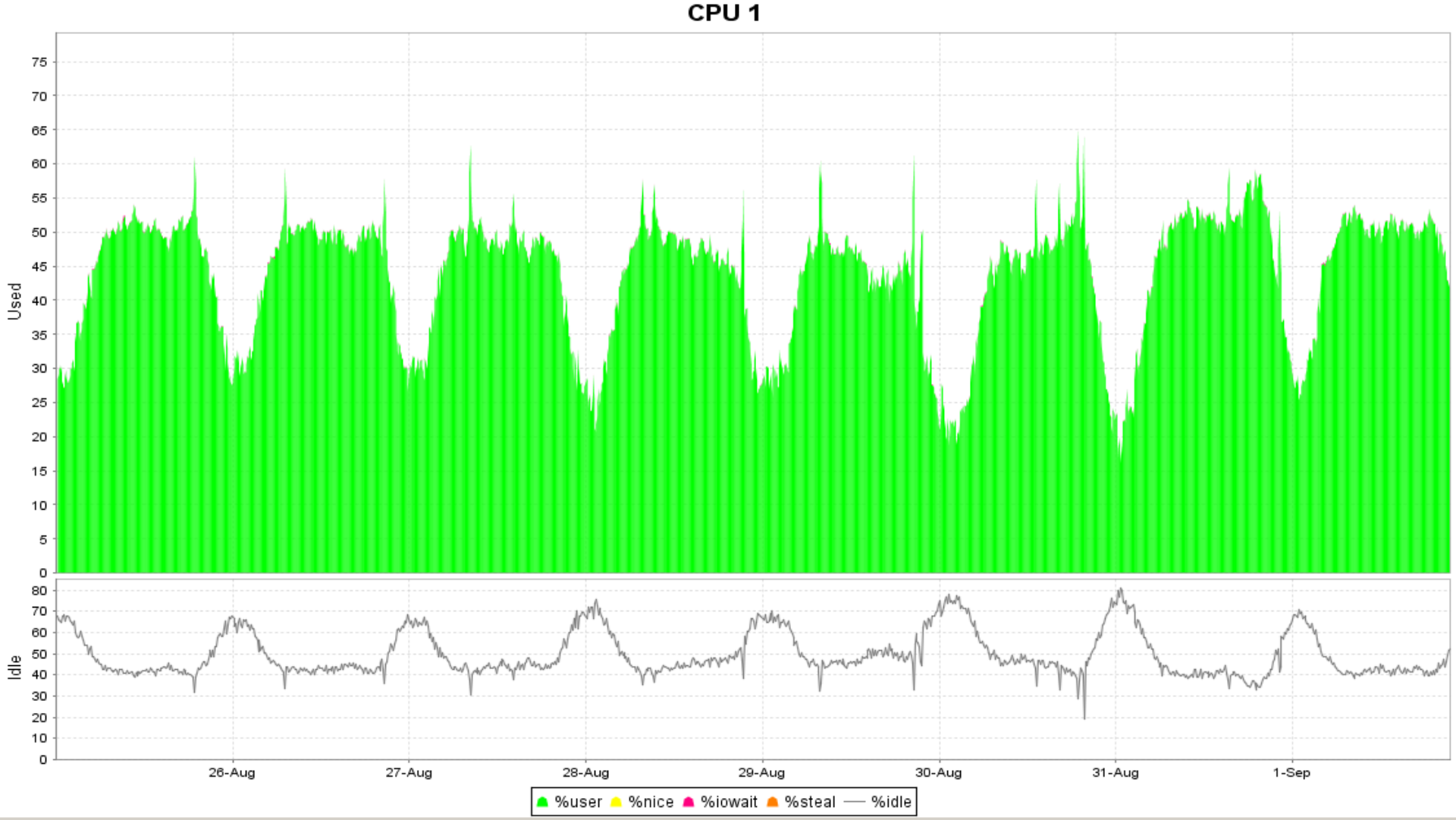

CPU 3
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle


CPU 4
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

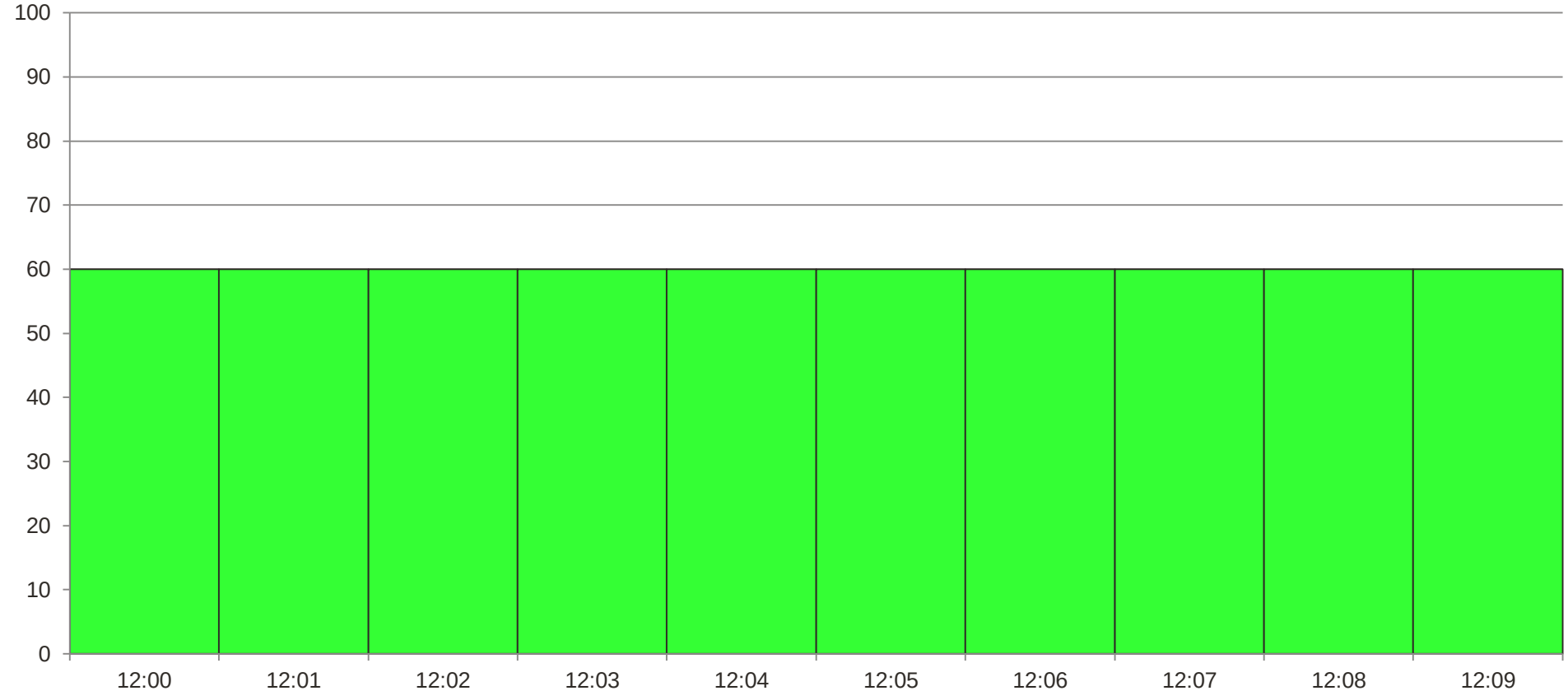

CPU 5
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

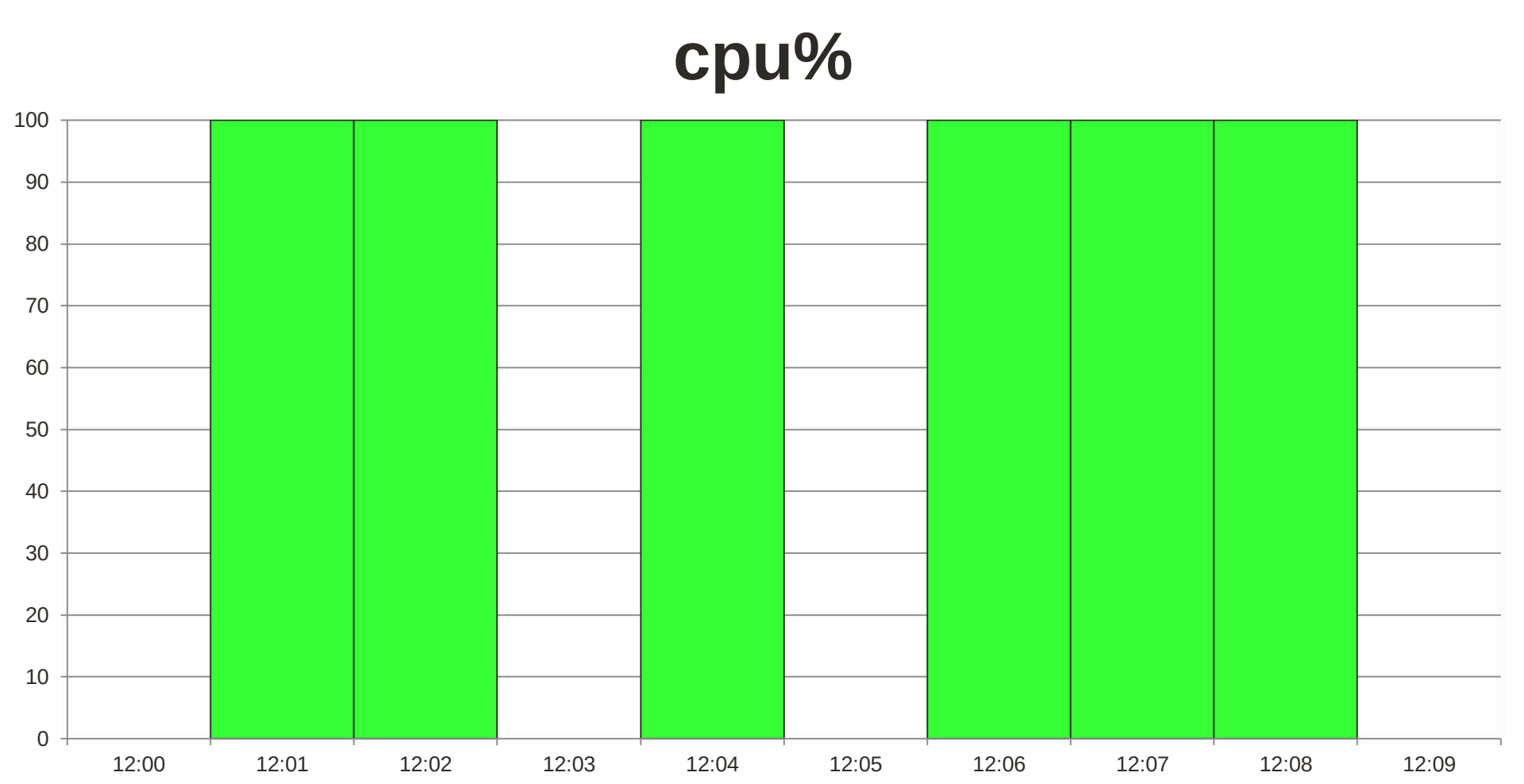

CPU 6
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle


CPU 7
Used
Idle
75
50
25
0
100
50
0
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle


CPU 1
Used
Idle
26-Aug
27-Aug
28-Aug
29-Aug
30-Aug
31-Aug
1-Sep
%user
%nice
%iowait
%steal
%idle

cpu%


100
90
80
70
60
50
40
30
20
10
0
12:00
12:01
12:02
12:03
12:04
12:05
12:06
12:07
12:08
12:09


cpu%
100
90
80
70
60
50
40
30
20
10
0
12:00
12:01
12:02
12:03
12:04
12:05
12:06
12:07
12:08
12:09

CPU utilization vs. Scheduler

The Linux Scheduler: a Decade of Wasted Cores

http://www.i3s.unice.fr/~jplozi/wastedcores/files/extended_talk.pdf

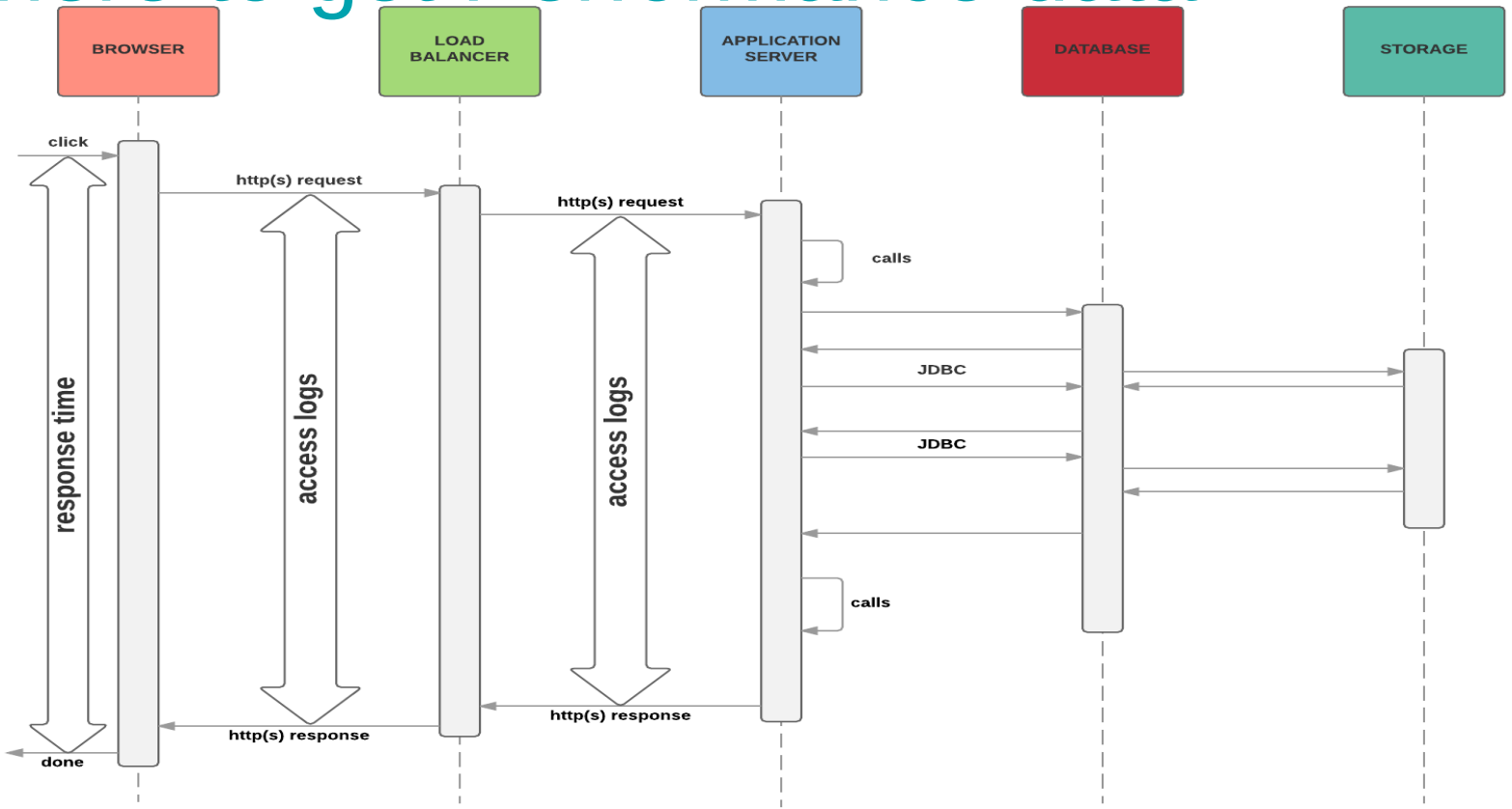
http://www.ece.ubc.ca/~sasha/papers/eurosys16-final29.pdf

Load

Its a silly number but people think its important.

From `kernel/sched/loadavg.c`

Where to get Performance data



Access Logs

All you need is to parse them:

- Load into database, parse, and query
- Use a tool which can parse / visualize data
- Parse it in-place

Access Logs Parsing

- pre-parse to filter only “interesting” rows
- get rid of columns with spaces / use " "
- sum time spent per type of page with a histogram
- report Top pages by Total Time, Longest, etc

Access Logs Parsing Example

```
printf "%-10s %-11s %-12s %-7s %-6s %-6s %-6s %-6s %-6s %s\n" "Total, sec" "Total count" "Average, ms" "<1s" "1-4s" "4-8s" "8-16s" "16-32s" "32s+"
"URL"

printf "%-10s %-11s %-12s %-7s %-6s %-6s %-6s %-6s %-6s %s\n" "----------" "-----------" "------------" "------" "------" "------" "------" "------" "-
-----" "---"

./p.sh |
  awk 'BEGIN { format = "%10d %11s %12.2f %6d %6d %6d %6d %6d %6d %s\n" }
{
  i=index($4, "RF.jsp");  page=$4;
  if (i<=0) { i=index($4, "OA.jsp") }
  if (i>0) {
    i=index($4, "&"); if (i>0) {page = substr($4, 1, i-1)}
  } else {
    i=index($4,   "?"); if (i>0) {page = substr($4, 1, i-1)}
    i=index(page, ";"); if (i>0) {page = substr(page, 1, i-1)}
  }
  total_time[page]+=$10; total_cnt[page]++;
  if ($10 < 1e6) {total1[page]++}
  else { if ($10 < 4e6 ) {total4[page]++}
  else { if ($10 < 8e6 ) {total8[page]+=1}
  else { if ($10 < 16e6) {total16[page]+=1}
  else { if ($10 < 32e6) {total32[page]+=1}
  else {totalInf[page]+=1}}}}}}
END {for (p in total_time) printf format, total_time[p]/1000000, total_cnt[p], total_time[p]/total_cnt[p]/1000, total1[p], total4[p], total8[p],
total16[p], total32[p], totalInf[p], p }' |
  sort -n -r | head -30
```

Access Logs Parsing Example

Total, sec	Total count	Average, ms	<1s	1-4s	4-8s	8-16s	16-32s	32s+	URL
9365	155997	60.04	152686	3311	0	0	0	0	/page1
4309	206	20920.19	175	17	0	0	0	14	/page2
2669	1639	1628.90	1610	17	1	2	0	9	/page3
1361	856	1589.97	614	140	48	51	2	1	/page4
1326	901	1472.56	864	32	1	0	0	4	/page5
1210	173	6995.27	172	0	0	0	0	1	/page6
1082	6404	169.07	6341	51	6	4	2	0	/page7

...

Access Logs: what's missing

- Currently running requests are not there
- v$session for Apache? Yes: mod_status
- Harder to parse (html), but doable

Application Server

- Start with access logs as well
- Instrumented applications? Rare beasts
- Application Performance Management tools
- If nothing else, “poor man’s profiler”

Application Server profiling

```
N threadstacks | aggregate | sort
```

More details:

http://tiny.cc/tmmdv-profiler

Application Server profiling

```
[oracle@oel6u4-2 test]$ ./prof.sh 7599 7601
Sampling PID=7599 every 0.5 seconds for 10 samples
      6  "main" prio=10 tid=0x00007f05c4007000 nid=0x1db1 runnable [0x00007f05c82f4000]
java.lang.Thread.State: RUNNABLE
    at java.io.FileInputStream.readBytes(Native Method)
    at java.io.FileInputStream.read(FileInputStream.java:220)
    at sun.security.provider.NativePRNG$RandomIO.readFully(NativePRNG.java:185)
    at sun.security.provider.NativePRNG$RandomIO.ensureBufferValid(NativePRNG.java:247)
    at sun.security.provider.NativePRNG$RandomIO.implNextBytes(NativePRNG.java:261)
    - locked <address> (a java.lang.Object)
    at sun.security.provider.NativePRNG$RandomIO.access$200(NativePRNG.java:108)
    at sun.security.provider.NativePRNG.engineNextBytes(NativePRNG.java:97)
    at java.security.SecureRandom.nextBytes(SecureRandom.java:433)
    - locked <address> (a java.security.SecureRandom)
    at java.security.SecureRandom.next(SecureRandom.java:455)
    at java.util.Random.nextInt(Random.java:189)
    at RandomUser.main(RandomUser.java:9)
```

Application Server restarts

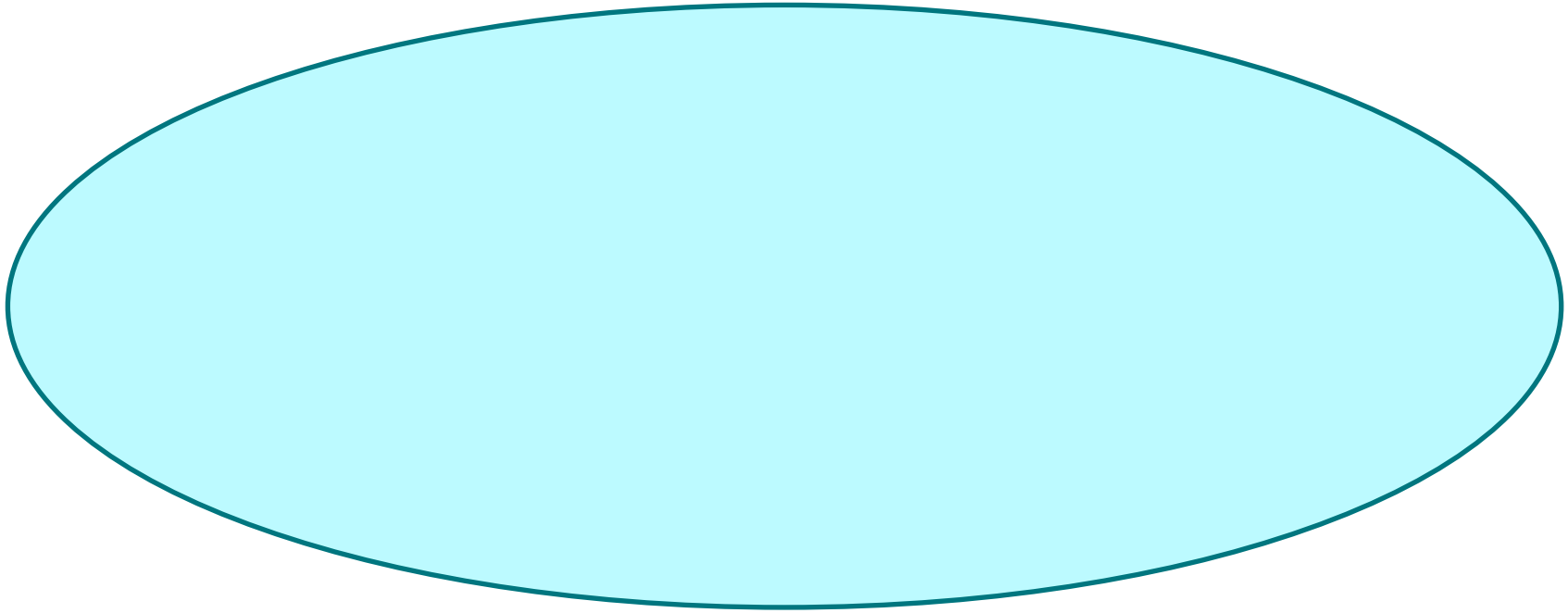
- Restart clears everything
- Including data for diagnostics
- tiny.cc/jvm8-troubleshooting – very useful
- Thread dumps & head dump (sometimes) are required minimum prior to restart

Application Server scaling

- Horizontal scaling is the “answer” to every performance issue
- Nobody knows how well application scales
- In most cases scaling up is required

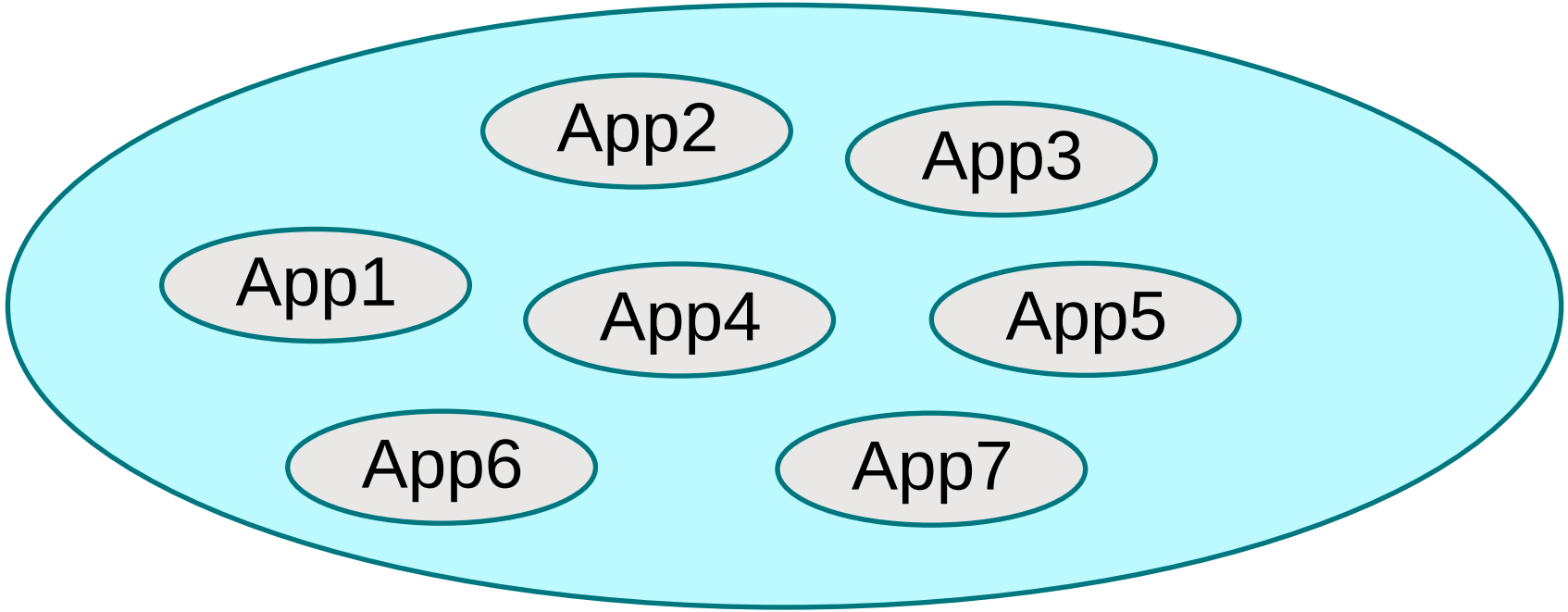
Database profiling

DB Time is *database* time



Database profiling

DB Time is *database* time



Database profiling

- ASH data has context to isolate active application DB sessions
- Correlate start date & response time from access logs with ASH data (script in the next slide)

Database profiling

```
set termout off
var t_date varchar2(30)
exec :t_date := '&1'
var t_secs number
exec :t_secs := '&2'
break on session_id skip 1
compute sum of cnt on session_id
set termout on

select session_id, sql_id, event, count(*) cnt, count(distinct sql_exec_id) exec_cnt
from ( select session_id, sql_id, event, sql_exec_id, dense_rank() over (order by s_cnt desc) d_rank
  from ( select s.session_id, s.sql_id, s.event, s.sql_exec_id, count(*) over (partition by s.session_id) s_cnt
    from v$active_session_history s
where s.user_id = :user_id and s.sample_time between to_date(:t_date, 'DD/Mon/YYYY:HH24:MI:SS')-(:t_secs/24/60/60) and to_date(:t_date,
'DD/Mon/YYYY:HH24:MI:SS')))
where d_rank <= 2
group by
  session_id, sql_id, event
order by
  session_id, count(*) desc
;

clear breaks
```

Database profiling

```
@ash_x 27/Feb/2017:00:09:49 302

SESSION_ID SQL_ID        Event                          Count EXEC_CNT
---------- ------------- ------------------------------ -------- --------
      2972 1prxxxxxxxxgv enq: TX - row lock contention      288        1
           4h8xxxxxxxx8m                                     6        5
           56kxxxxxxxxr8                                     2        1
           a6cxxxxxxxxqk                                     2        1
           8g6xxxxxxxxyu                                     1        1
           d5sxxxxxxxxz2                                     1        1

**********                                           --------
sum                                                       301
```

Summary

- Performance is measured in seconds or (rarely) #/second
- Capacity ≠ Performance
- Capacity metrics are often misinterpreted in many ways
- Sampling is an easy & powerful approach to Performance diagnostics
- Time & context is important

Thank You!

Questions?

Comments are Welcome
timur.akhmadeev@gmail.com

© 2017 Pythian. Confidential