PERCONA

Performance Consulting Experts

Goal Driven Performance Optimization

OSCON

July 20-24 2009

San Jose,CA

Peter Zaitsev

Percona Inc

Understanding Performance

- Latency/Response Time
  - Always Important
  - Can be very different
    - 50ms of Ajax Request
    - 30minutes for report
- Capacity/Throughtput
  - Often important for multi-user systems
  - System can do 1000 transactions/second

Throughput/Latency Relation

- Response time tends to increase with throughput
  - When system overload response time goes to infinity
- Call Center analogy
  - Fewer people servicing calls = better utilization
    - Same as throughput per person
  - More people servicing calls = better response time
    - Calls spend less time waiting in the queue
- Performance Optimization Goal
  - Maximizing Capacity/Throughput/Utilization while maintaining Response time within a guidelines

Response Time Metrics

- Average/Medium/Response Time
  - Not a good metric for adequate performance
  - Same as average person temperature in hospital
  - Can be helpful for historical trending
- Maximum Response Time
  - Good in theory. We want No requests taking longer than X
  - Hard to work in practice – some requests will take too long
- Define Percentile response time
  - 95% or requests serviced within 500ms
  - 99% or requests serviced within 1000ms

Even Response Time

- 95% response time goal will allow your system to be non responsive for an hour every day
  - Ie extremely bad performance when taking backup
- You want to ensure there is no stalls/performance dips.
- If page loads slow and user presses reload and it loads quickly it is OK – there are always network glitches.
- Define your performance goals at short intervals.
  - Goals should be met at ALL 5 minutes intervals.

Response Time and an Object

- Not all the pages are created Equal
- Complexity and User Requirement Differ
- Ajax Pop Ups
  - 50ms
- Profile Page Generation
  - 150ms
- Search
  - 300ms
- Site Usage Report
  - 1000ms

Responses by Type of Client

- Human Being
  - Actual Human waiting and being impatient
  - Response Time critical
- Bots
  - Some systems have over 80% of bot traffic
  - Bot response time is less critical
    - Though should be good enough to be indexed
- Interactive Web Services
  - Can be used to generate pages on other sites
  - Low Response time is even more critical

Avoid Performance Holes

- Imagine multi-vendor online store
- 99.9% vendors have 100 items or less
- 5 vendors have over 1.000.000 items
  - So their admin pages are slow
- Because they are so few they do not affect your 99% response time.
- They can be extremely important for business
  - Need to ensure they get adequate performance
- Other choice is placing restriction
  - No more than 100000 items allowed in the shop

What Performance to Measure

- Client Side Performance
  - Keynote, Gomez
  - Monitoring for Selected Pages
- Client Side Instrumentation
  - http://code.google.com/p/jiffy-web/
  - Show user level response for all pages
- Back End Instrumentation
  - Understanding Response Time from Backend

Summary of the Goal

- Define 95%, 99% etc response time
- For each User Interaction/Class
- Measured/Monitored each 5 minutes
- From Front End and Backend observation
- Avoiding Performance Holes
  - Some actions always/often slow for some users.

Production Instrumentation

- Many People Instrument Test System
  - Option to print out Queries/Web Service Requests
  - Great for Debugging/Testing
  - Will not show a lot of performance problems
    - Cold vs hot requests
    - Contention happening in production
    - Special User Cases
- Run Instrumented App in Production and Store Data
  - Can instrument only one of Web servers if overhead is large.
  - Can log only 1% of user sessions if can't handle all data

What to Instrument

- Total Response Time
- CPU Time
- “Wait Time”
  - Connections/Database Queries
  - MemCache
  - Web Services Request
  - Other Network Requests
- Additional Information
  - Number and Nature of different queries
  - Hits/Misses for Queries
  - Options which can affect performance

Where to Store

- Plain old log files
  - Or directly to the database for smaller systems
- Load them to the database
- Or Hadoop on the larger scale
- Generate standard reports
- Provide Ad-Hoc way to do deep data analyses

Sample Logging Table

- Sample logging table from boardreader.com

```
CREATE TABLE `performance_log_090721` (
  `ip` varchar(15) NOT NULL,
  `server_ip` varchar(25) NOT NULL,
  `page` varchar(3000) NOT NULL,
  `utime` float NOT NULL,
  `stime` float NOT NULL,
  `wtime` float NOT NULL,
  `mysql_time` float NOT NULL,
  `sphinx_time` float NOT NULL,
  `mysql_count_queries` int(11) NOT NULL,
  `mysql_queries` text NOT NULL,
  `sphinx_count_queries` int(11) NOT NULL,
  `sphinx_real_count_queries` int(11) NOT NULL,
  `sphinx_queries` text NOT NULL,
  `logged` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
  `user_agent` varchar(255) NOT NULL,
  `referer` varchar(255) NOT NULL,
  `bot` enum('','google','yahoo','msn','lycos','other') NOT NULL,
  `js_cookie` tinyint(1) unsigned NOT NULL default '0',
  `page_type` enum('','search','ajax','forumprofile','siteprofile','threadprofile','topicprofile','domainprofile','other') NOT
NULL,
  `id` char(32) NOT NULL default ''
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Sample Table Row

- The result may look like this.

```
*************************** 5. row ***************************
                    ip: 91.148.82.211
             server_ip: web08.boardreader.com
                  page: boardreader.com/s/nba29k.html?f=47977&extended_search=1
                 utime: 0.129981
                 wtime: 0.242401
            mysql_time: 0.004417
           sphinx_time: 0.083193
    sphinx_results_time: 0.078
    mysql_count_queries: 15
          mysql_queries:
   sphinx_count_queries: 3
sphinx_real_count_queries: 3
         sphinx_queries:
                 stime: 0.008998
                logged: 2009-07-20 20:55:48
            user_agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB6; .NET CLR 2.0.50727; InfoPath.2)
              referer: http://boardreader.com/fp/FileForums_14910/PC_Games_CD_2_DVD_Conversion_47977.html
                  bot:
           js_cookie: 1
           page_type: search
                  id: 5ab03bc440ffa0c62610a62db988cb81
```

Search Response Time

- Average Distribution
- About 6% is time unaccounted for

- select avg(wtime) r, avg(stime+utime)/avg(wtime) cpup ,avg(mysql_time)/avg(wtime) mp, avg(sphinx_time)/avg(wtime) sp, avg(wtime-stime-utime-sphinx_time-mysql_time)/avg(wtime) rst from performance_log_090721 where page_type='search' \G
- *************************** 1. row ***************************
- r: 1.2175869055517
- cpup: 0.16983144536072
- mp: 0.1544487152423
- sp: 0.61537297006254
- rst: 0.060346869334443
- 1 row in set (4.16 sec)

See Hourly Trend

- Not All hours are created Equal

```
mysql> select date_format(logged,"%H") h ,round(avg(wtime),3) r, round(avg(stime+utime)/avg(wtime),2) cpup
,round(avg(mysql_time)/avg(wtime),2) mp, round(avg(sphinx_time)/avg(wtime),2) sp, round(avg(wtime-stime-utime-sphinx_time-
mysql_time)/avg(wtime),2) rst  from performance_log_090721 where page_type='search' group by 1;
+------+-------+------+------+------+------+
| h    | r     | cpup | mp   | sp   | rst  |
+------+-------+------+------+------+------+
| 00   | 1.816 | 0.11 | 0.14 | 0.70 | 0.05 |
| 01   | 1.480 | 0.17 | 0.18 | 0.59 | 0.06 |
| 02   | 1.394 | 0.16 | 0.22 | 0.53 | 0.09 |
....
| 08   | 1.384 | 0.13 | 0.09 | 0.74 | 0.04 |
| 09   | 1.315 | 0.17 | 0.11 | 0.67 | 0.04 |
| 10   | 0.950 | 0.20 | 0.15 | 0.60 | 0.05 |
| 11   | 0.874 | 0.21 | 0.16 | 0.57 | 0.06 |
| 12   | 1.139 | 0.17 | 0.13 | 0.65 | 0.05 |
| 13   | 1.191 | 0.16 | 0.14 | 0.65 | 0.05 |
| 14   | 1.349 | 0.16 | 0.19 | 0.58 | 0.06 |
| 15   | 1.076 | 0.20 | 0.21 | 0.53 | 0.06 |
| 16   | 1.526 | 0.14 | 0.14 | 0.58 | 0.13 |
| 17   | 0.853 | 0.24 | 0.19 | 0.50 | 0.07 |
| 18   | 0.978 | 0.25 | 0.23 | 0.43 | 0.09 |
| 19   | 0.924 | 0.23 | 0.17 | 0.54 | 0.06 |
| 20   | 1.310 | 0.18 | 0.26 | 0.47 | 0.09 |
| 21   | 1.211 | 0.17 | 0.24 | 0.51 | 0.08 |
| 22   | 1.538 | 0.14 | 0.19 | 0.59 | 0.08 |
| 23   | 1.450 | 0.15 | 0.18 | 0.60 | 0.06 |
+------+-------+------+------+------+------+
24 rows in set (4.33 sec)
```

Apply The Goal

- We want search to be faster than 1 sec
  - Analyze pages which are slower than that
- Focus on optimizing the largest response time contributor

```
mysql> select round(avg(wtime),3) r, round(avg(stime+utime)/avg(wtime),2) cpup
,round(avg(mysql_time)/avg(wtime),2) mp, round(avg(sphinx_time)/avg(wtime),2) sp,
round(avg(wtime-stime-utime-sphinx_time-mysql_time)/avg(wtime),2) rst  from
performance_log_090721 where date_format(logged,"%H")=0 and
page_type='search' and wtime>1;
+-------+------+------+------+------+
| r     | cpup | mp   | sp   | rst  |
+-------+------+------+------+------+
| 2.571 | 0.09 | 0.14 | 0.72 | 0.05 |
+-------+------+------+------+------+
1 row in set (4.37 sec)
```

Find Pages worth attention

- Focus on popular pages outside desired response time

```
mysql> select count(*) cnt, avg(wtime) wt , avg(sphinx_time) sp , avg(mysql_time) my,page from performance_log_090721 where
page_type='search' and wtime>1 group by page  order by cnt desc limit 2,1 \G
*************************** 1. row ***************************
 cnt: 189
  wt: 1.7529319978896
  sp: 1.2289490875685
  my: 0.060628327807145
page: boardreader.com/s/%D8%A7%D9%81%D9%84%D8%A7%D9%85%20%D8%B3%D9%83%D8%B3%2089.html?f=393284
1 row in set (4.65 sec)
```

The Next Step

- Analyze which Sphinx Queries are slow
  - There are multiple queries per page
- Use request_id to “connect” to the data in its logs
  - You need to understand why those exact queries are slow
  - [Fri Jul 24 04:03:26.466 2009] 0.011 sec [ext2/5/rel 30 (0,1000) @published] [linksfull_node1,linksinc_node1] [ios=2 kb=8.9 ioms=7.2]
    - We can see the request took 11ms and out of that we had 2 ios which took 7ms
- Sphinx Support passing request_id via API
  - In MySQL can just add comment to the query

Start from what is most important

- Optimize Most important User Interactions first
- Pick What case to focus in
  - Queries which do not meet response time
  - But not Worse Case Scenario
    - There are always going to be outliers
- Do not analyze just queries above response time threshold
  - It is much easier to reach 95% of 1 second if 50% of the queries are below 500ms.

Benefits of Such Approach

- Direct connection to the business goals
- High Priority problems targeted first
- Focus on real stuff
  - No guess work like “is my buffer pool hit ratio bad?” or “am I doing too much full table scans ?”
  - If these there the issues you will find and fix them anyway.
- Understandable and predictable result
  - If MySQL contributes 15% to the response time I can't possibly double performance focusing on MySQL optimization.

Questions ?

- Thanks For coming !
- http://www.percona.com
  - We do Performance/HA/Scalability Consulting and Support
- pz@percona.com