YAHOO!

Database Engineering and Operations

BY **Ashwin Nellore**

Yahoo

- Advertising Products
- Publisher Products
- Platforms
- Internal Products

Engineering

- Database as a Service
- Continuous Delivery
- Code Reviews
- Performance Analyzer (Open Sourced)
- Performance Analytics

Database as a Service

- Self Service on Private Cloud
- Multitenant and Dedicated Solutions
- Data Store Guidance
- User Management
- Backups
- Migrations
- Interleaved with dependent systems

Continuous Delivery

- Custom Configuration Management
- Github
- Jenkins Pipeline
- Database version control
- Automated Tests for syntax errors
- Code Reviews
- Developer Notifications

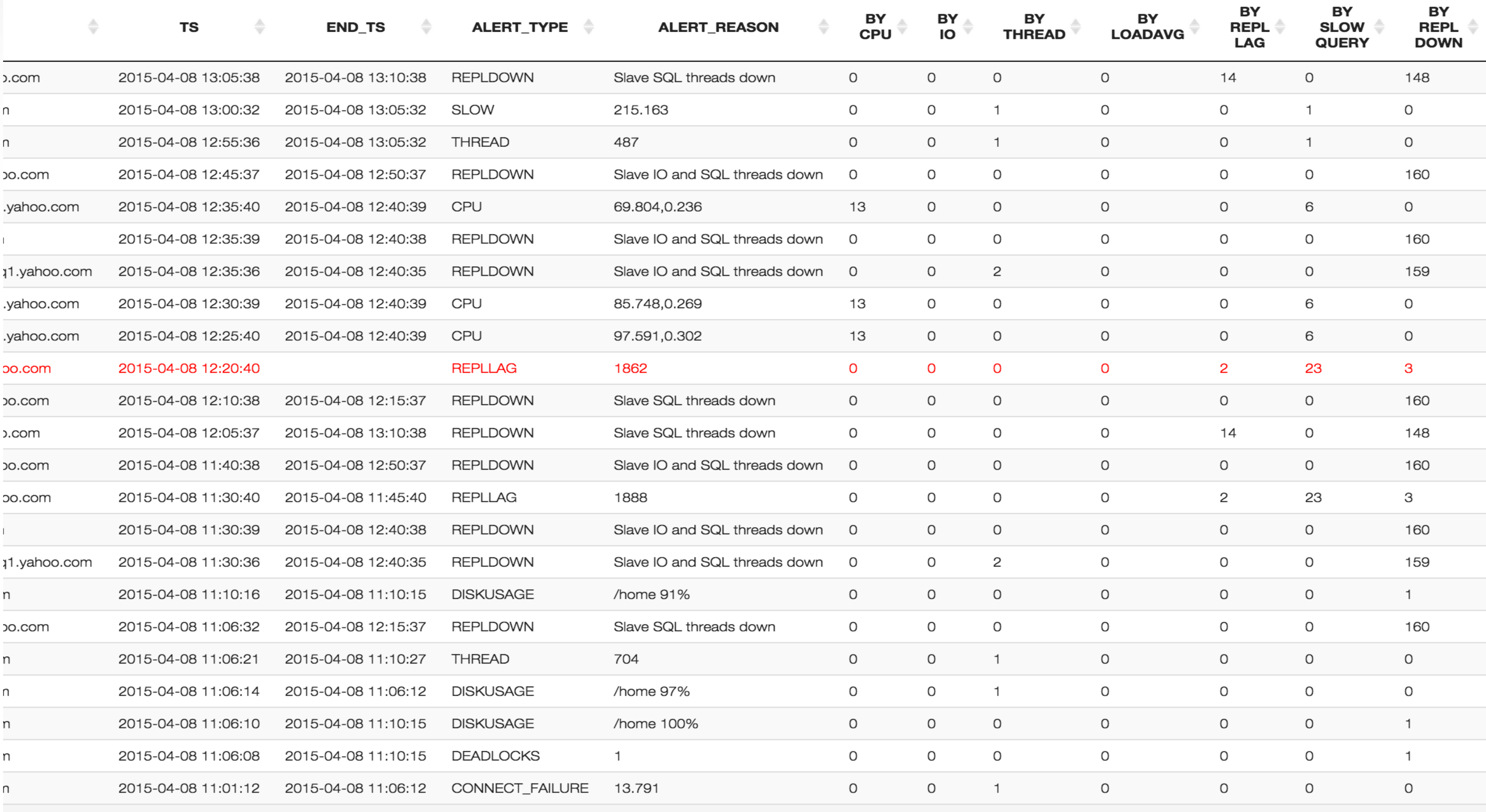
Performance Analyzer

- Lightweight and Agentless Java Web Application
- Self contained and easy to deploy anywhere
- Rich User Interface
- Gather and store performance metrics
- Detect anomalies and raise alerts
- Real time performance data access
- New metrics and alerts can be defined and deployed during runtime.
- Highly agile and extensible software development
- No license required

Dashboard: Alerts From Past 24 Hours

- After login, dashboard will display alerts from past 24 hours and metrics for current database health, for all database servers under management.
- Active alerts are colored in red.
- Built-in alerts are summarized and displayed in the list.
- List is sortable on all columns.
- List can be further restricted to a single server group.
- Forensic data gathered when an alert was detected can be viewed or downloaded from the same page.

Dashboard: Alerts From Past 24 Hours

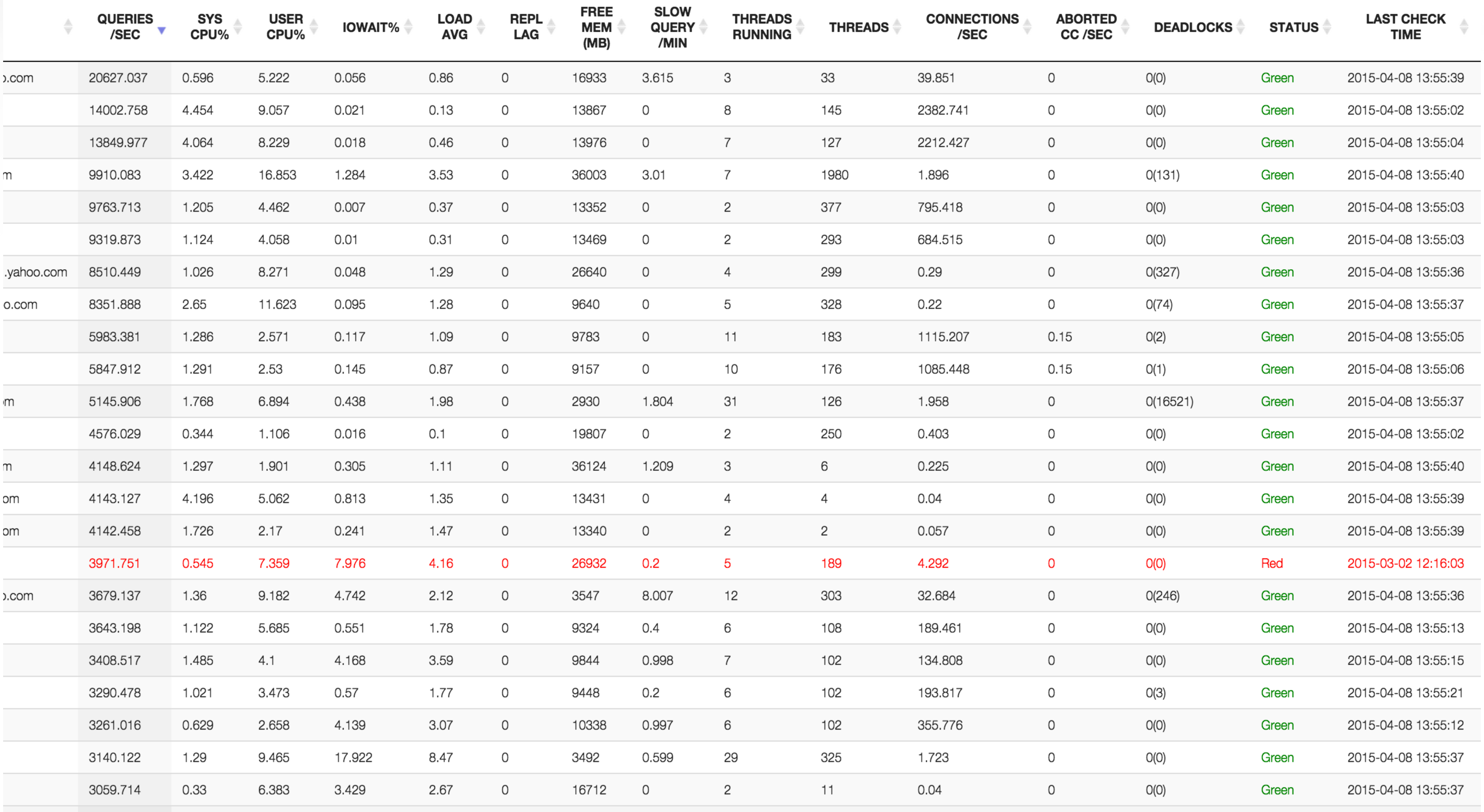


	TS	END_TS	ALERT_TYPE	ALERT_REASON	BY CPU	BY IO	BY THREAD	BY LOADAVG	BY REPL LAG	BY SLOW QUERY	BY REPL DOWN
ɔ.com	2015-04-08 13:05:38	2015-04-08 13:10:38	REPLDOWN	Slave SQL threads down	0	0	0	0	14	0	148
n	2015-04-08 13:00:32	2015-04-08 13:05:32	SLOW	215.163	0	0	1	0	0	1	0
n	2015-04-08 12:55:36	2015-04-08 13:05:32	THREAD	487	0	0	1	0	0	1	0
ɔo.com	2015-04-08 12:45:37	2015-04-08 12:50:37	REPLDOWN	Slave IO and SQL threads down	0	0	0	0	0	0	160
.yahoo.com	2015-04-08 12:35:40	2015-04-08 12:40:39	CPU	69.804,0.236	13	0	0	0	0	6	0
ı	2015-04-08 12:35:39	2015-04-08 12:40:38	REPLDOWN	Slave IO and SQL threads down	0	0	0	0	0	0	160
g1.yahoo.com	2015-04-08 12:35:36	2015-04-08 12:40:35	REPLDOWN	Slave IO and SQL threads down	0	0	2	0	0	0	159
.yahoo.com	2015-04-08 12:30:39	2015-04-08 12:40:39	CPU	85.748,0.269	13	0	0	0	0	6	0
.yahoo.com	2015-04-08 12:25:40	2015-04-08 12:40:39	CPU	97.591,0.302	13	0	0	0	0	6	0
ɔo.com	2015-04-08 12:20:40		REPLLAG	1862	0	0	0	0	2	23	3
ɔo.com	2015-04-08 12:10:38	2015-04-08 12:15:37	REPLDOWN	Slave SQL threads down	0	0	0	0	0	0	160
ɔ.com	2015-04-08 12:05:37	2015-04-08 13:10:38	REPLDOWN	Slave SQL threads down	0	0	0	0	14	0	148
ɔo.com	2015-04-08 11:40:38	2015-04-08 12:50:37	REPLDOWN	Slave IO and SQL threads down	0	0	0	0	0	0	160
ɔo.com	2015-04-08 11:30:40	2015-04-08 11:45:40	REPLLAG	1888	0	0	0	0	2	23	3
ı	2015-04-08 11:30:39	2015-04-08 12:40:38	REPLDOWN	Slave IO and SQL threads down	0	0	0	0	0	0	160
g1.yahoo.com	2015-04-08 11:30:36	2015-04-08 12:40:35	REPLDOWN	Slave IO and SQL threads down	0	0	2	0	0	0	159
n	2015-04-08 11:10:16	2015-04-08 11:10:15	DISKUSAGE	/home 91%	0	0	0	0	0	0	1
ɔo.com	2015-04-08 11:06:32	2015-04-08 12:15:37	REPLDOWN	Slave SQL threads down	0	0	0	0	0	0	160
n	2015-04-08 11:06:21	2015-04-08 11:10:27	THREAD	704	0	0	1	0	0	0	0
n	2015-04-08 11:06:14	2015-04-08 11:06:12	DISKUSAGE	/home 97%	0	0	1	0	0	0	0
n	2015-04-08 11:06:10	2015-04-08 11:10:15	DISKUSAGE	/home 100%	0	0	0	0	0	0	1
n	2015-04-08 11:06:08	2015-04-08 11:10:15	DEADLOCKS	1	0	0	0	0	0	0	1
n	2015-04-08 11:01:12	2015-04-08 11:06:12	CONNECT_FAILURE	13.791	0	0	1	0	0	0	0

Dashboard: Current Health Status

- Display most recent performance metrics for all managed servers in a single screen
- Results can be limited to a single server group.
- List is sortable and color coded to prioritize action and response
- Metrics Included:
  - QPS
  - CPU, Load Average and IO Waits
  - Free Memory
  - Slow Query Count
  - Active and Total Threads
  - Connection Rates and Failures
  - Replication lags
  - Deadlocks
  - Time used for last round of metrics scan.

Dashboard: Current Health Status

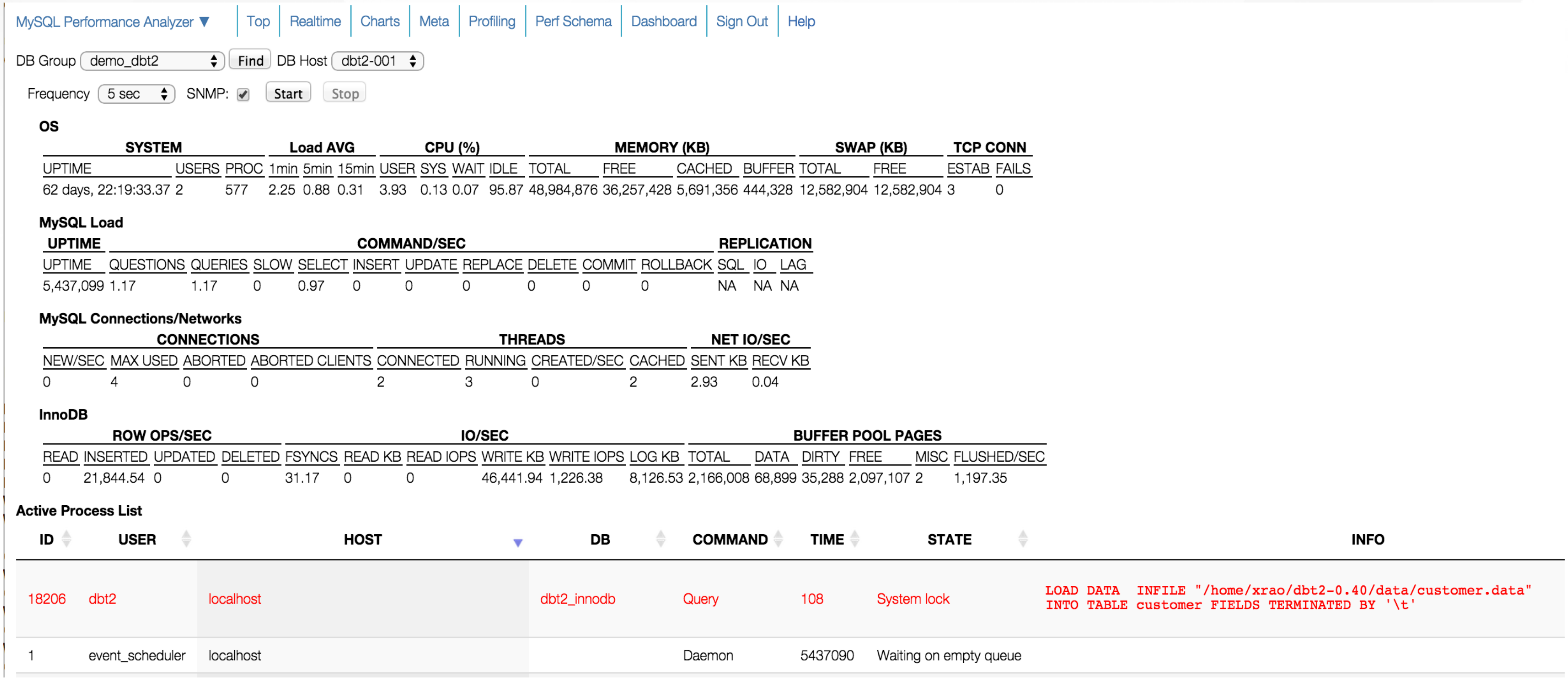


	QUERIES /SEC	SYS CPU%	USER CPU%	IOWAIT%	LOAD AVG	REPL LAG	FREE MEM (MB)	SLOW QUERY /MIN	THREADS RUNNING	THREADS	CONNECTIONS /SEC	ABORTED CC /SEC	DEADLOCKS	STATUS	LAST CHECK TIME
.com	20627.037	0.596	5.222	0.056	0.86	0	16933	3.615	3	33	39.851	0	0(0)	Green	2015-04-08 13:55:39
	14002.758	4.454	9.057	0.021	0.13	0	13867	0	8	145	2382.741	0	0(0)	Green	2015-04-08 13:55:02
	13849.977	4.064	8.229	0.018	0.46	0	13976	0	7	127	2212.427	0	0(0)	Green	2015-04-08 13:55:04
m	9910.083	3.422	16.853	1.284	3.53	0	36003	3.01	7	1980	1.896	0	0(131)	Green	2015-04-08 13:55:40
	9763.713	1.205	4.462	0.007	0.37	0	13352	0	2	377	795.418	0	0(0)	Green	2015-04-08 13:55:03
	9319.873	1.124	4.058	0.01	0.31	0	13469	0	2	293	684.515	0	0(0)	Green	2015-04-08 13:55:03
.yahoo.com	8510.449	1.026	8.271	0.048	1.29	0	26640	0	4	299	0.29	0	0(327)	Green	2015-04-08 13:55:36
o.com	8351.888	2.65	11.623	0.095	1.28	0	9640	0	5	328	0.22	0	0(74)	Green	2015-04-08 13:55:37
	5983.381	1.286	2.571	0.117	1.09	0	9783	0	11	183	1115.207	0.15	0(2)	Green	2015-04-08 13:55:05
	5847.912	1.291	2.53	0.145	0.87	0	9157	0	10	176	1085.448	0.15	0(1)	Green	2015-04-08 13:55:06
m	5145.906	1.768	6.894	0.438	1.98	0	2930	1.804	31	126	1.958	0	0(16521)	Green	2015-04-08 13:55:37
	4576.029	0.344	1.106	0.016	0.1	0	19807	0	2	250	0.403	0	0(0)	Green	2015-04-08 13:55:02
m	4148.624	1.297	1.901	0.305	1.11	0	36124	1.209	3	6	0.225	0	0(0)	Green	2015-04-08 13:55:40
om	4143.127	4.196	5.062	0.813	1.35	0	13431	0	4	4	0.04	0	0(0)	Green	2015-04-08 13:55:39
om	4142.458	1.726	2.17	0.241	1.47	0	13340	0	2	2	0.057	0	0(0)	Green	2015-04-08 13:55:39
	3971.751	0.545	7.359	7.976	4.16	0	26932	0.2	5	189	4.292	0	0(0)	Red	2015-03-02 12:16:03
.com	3679.137	1.36	9.182	4.742	2.12	0	3547	8.007	12	303	32.684	0	0(246)	Green	2015-04-08 13:55:36
	3643.198	1.122	5.685	0.551	1.78	0	9324	0.4	6	108	189.461	0	0(0)	Green	2015-04-08 13:55:13
	3408.517	1.485	4.1	4.168	3.59	0	9844	0.998	7	102	134.808	0	0(0)	Green	2015-04-08 13:55:15
	3290.478	1.021	3.473	0.57	1.77	0	9448	0.2	6	102	193.817	0	0(3)	Green	2015-04-08 13:55:21
	3261.016	0.629	2.658	4.139	3.07	0	10338	0.997	6	102	355.776	0	0(0)	Green	2015-04-08 13:55:12
	3140.122	1.29	9.465	17.922	8.47	0	3492	0.599	29	325	1.723	0	0(0)	Green	2015-04-08 13:55:37
	3059.714	0.33	6.383	3.429	2.67	0	16712	0	2	11	0.04	0	0(0)	Green	2015-04-08 13:55:37

Real Time Top

- Inspired by MyTop/InnoTop
- Display selected OS metrics and MySQL metrics in real time.
- Display MySQL process list in real time.
- OS metrics:
  - Uptime, Load Average, CPU, Memory, Swap, TCP Connections.
- MySQL metrics
  - General: Uptime, QPS, Commands, Replication
  - Network/Threads: Connections, Threads, Network IO
  - InnoDB: Row operation, IO, Buffer Pool

Real Time Top



DB Group demo_dbt2 DB Host dbt2-001

Frequency 5 sec SNMP: ☑

OS

SYSTEM			Load AVG			CPU (%)				MEMORY (KB)				SWAP (KB)		TCP CONN	
UPTIME	USERS	PROC	1min	5min	15min	USER	SYS	WAIT	IDLE	TOTAL	FREE	CACHED	BUFFER	TOTAL	FREE	ESTAB	FAILS
62 days, 22:19:33.37	2	577	2.25	0.88	0.31	3.93	0.13	0.07	95.87	48,984,876	36,257,428	5,691,356	444,328	12,582,904	12,582,904	3	0

MySQL Load

UPTIME	COMMAND/SEC										REPLICATION		
UPTIME	QUESTIONS	QUERIES	SLOW	SELECT	INSERT	UPDATE	REPLACE	DELETE	COMMIT	ROLLBACK	SQL	IO	LAG
5,437,099	1.17	1.17	0	0.97	0	0	0	0	0	0	NA	NA	NA

MySQL Connections/Networks

CONNECTIONS				THREADS				NET IO/SEC	
NEW/SEC	MAX USED	ABORTED	ABORTED CLIENTS	CONNECTED	RUNNING	CREATED/SEC	CACHED	SENT KB	RECV KB
0	4	0	0	2	3	0	2	2.93	0.04

InnoDB

ROW OPS/SEC				IO/SEC						BUFFER POOL PAGES					
READ	INSERTED	UPDATED	DELETED	FSYNCS	READ KB	READ IOPS	WRITE KB	WRITE IOPS	LOG KB	TOTAL	DATA	DIRTY	FREE	MISC	FLUSHED/SEC
0	21,844.54	0	0	31.17	0	0	46,441.94	1,226.38	8,126.53	2,166,008	68,899	35,288	2,097,107	2	1,197.35

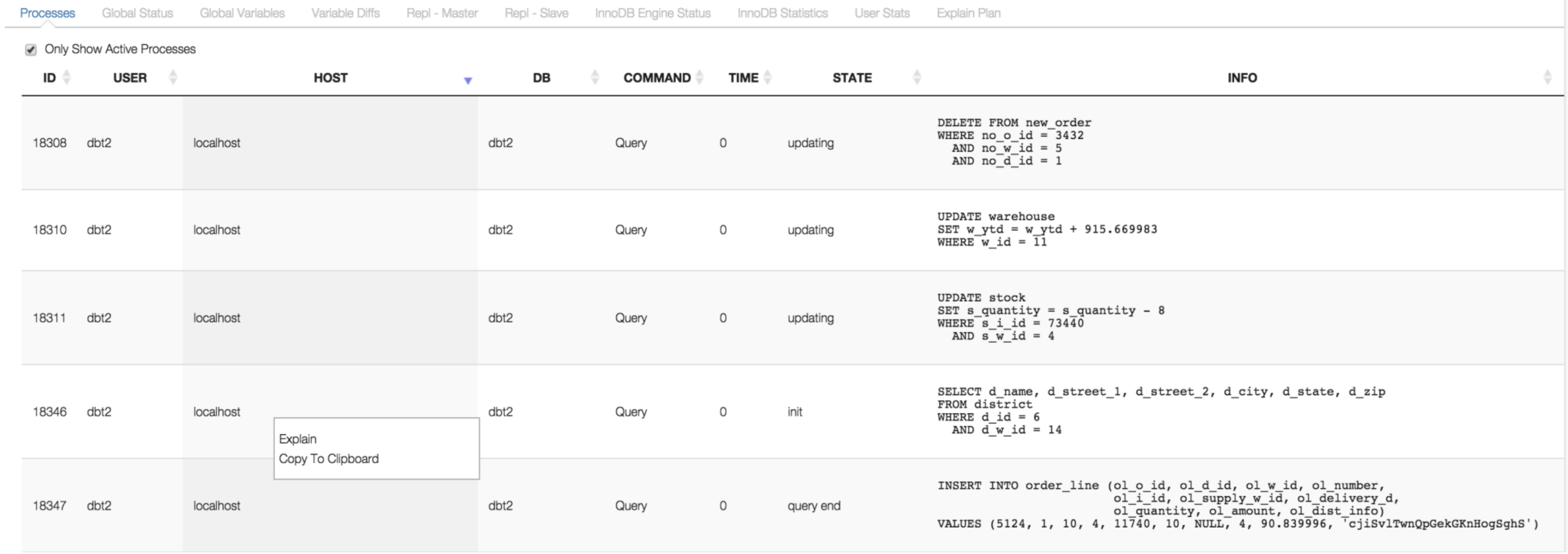
Active Process List

ID	USER	HOST	DB	COMMAND	TIME	STATE	INFO
18206	dbt2	localhost	dbt2_innodb	Query	108	System lock	`LOAD DATA  INFILE "/home/xrao/dbt2-0.40/data/customer.data" INTO TABLE customer FIELDS TERMINATED BY '\t'`
1	event_scheduler	localhost		Daemon	5437090	Waiting on empty queue	

Real Time - Details

- User friendly and safe tool to access various performance related information schema tables and SHOW commands.
- For metrics or status related information, the changes can be calculated and displayed automatically, or triggered manually.
- Context help and context menu can help to digest the information or navigate to other places for further researches.
- Features supported:
  - Process list
  - Global status and changes, can be filtered by partial keyword
  - Configuration variables, the history, and comparison with other MySQL servers.
  - Replication Status
  - Parsed InnoDB engine status
  - InnoDB status
  - User Statistics when available, and the changes to identify hot users, tables, etc.
  - Explain plan, including JSON format, either triggered from process list or input manually.

Real Time Details And Process List



☑ Only Show Active Processes

ID	USER	HOST	DB	COMMAND	TIME	STATE	INFO

18308	dbt2	localhost	dbt2	Query	0	updating	`DELETE FROM new_order`
							`WHERE no_o_id = 3432`
							`  AND no_w_id = 5`
							`  AND no_d_id = 1`

18310	dbt2	localhost	dbt2	Query	0	updating	`UPDATE warehouse`
							`SET w_ytd = w_ytd + 915.669983`
							`WHERE w_id = 11`

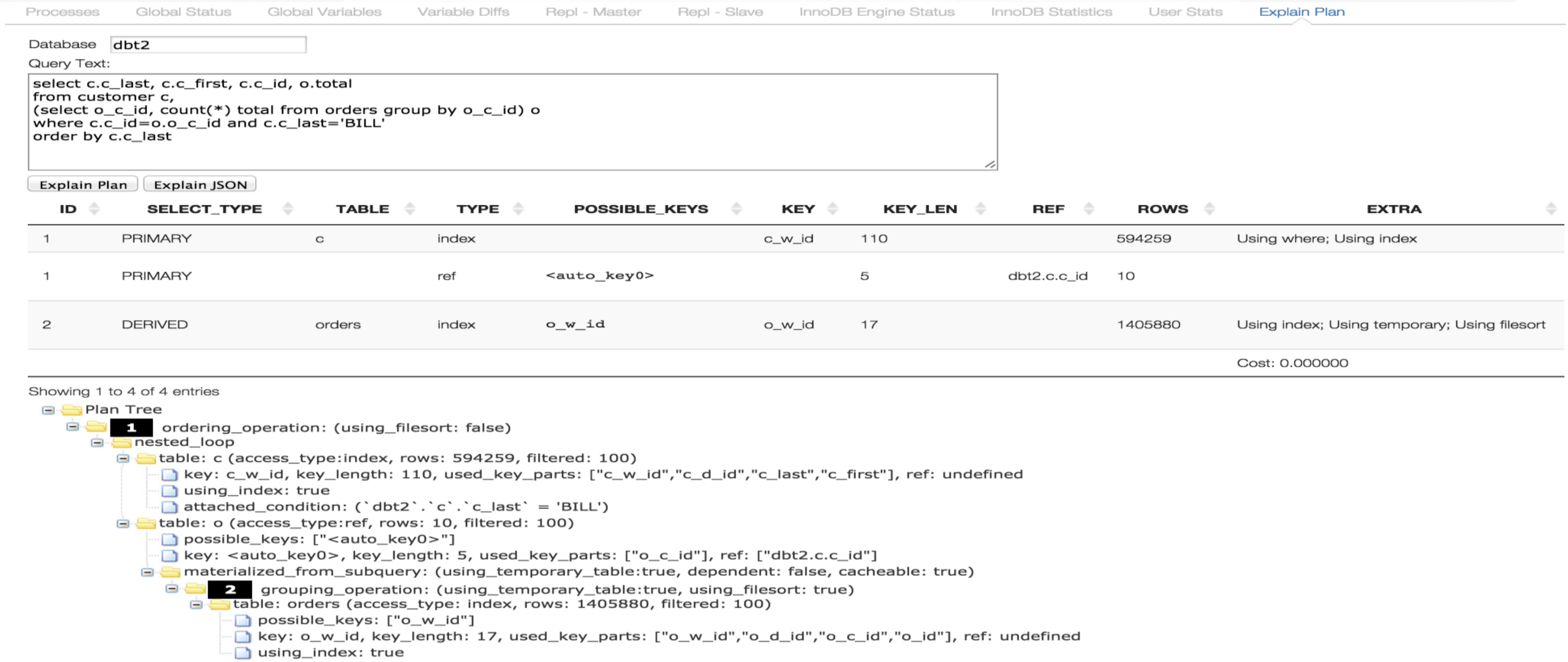
18311	dbt2	localhost	dbt2	Query	0	updating	`UPDATE stock`
							`SET s_quantity = s_quantity - 8`
							`WHERE s_i_id = 73440`
							`  AND s_w_id = 4`

18346	dbt2	localhost	dbt2	Query	0	init	`SELECT d_name, d_street_1, d_street_2, d_city, d_state, d_zip`
							`FROM district`
							`WHERE d_id = 6`
							`  AND d_w_id = 14`

18347	dbt2	localhost	dbt2	Query	0	query end	`INSERT INTO order_line (ol_o_id, ol_d_id, ol_w_id, ol_number,`
							`ol_i_id, ol_supply_w_id, ol_delivery_d,`
							`ol_quantity, ol_amount, ol_dist_info)`
							`VALUES (5124, 1, 10, 4, 11740, 10, NULL, 4, 90.839996, 'cjiSvlTwnQpGekGKnHogSghS')`

- Tabs to access data from various information schema tables and SHOW commands.
- Context Menu to run EXPLAIN on any SELECT query
- Thread level detailed info from Performance Schema screen

Explain Plan and JSON Output



Database: dbt2

Query Text:

```
select c.c_last, c.c_first, c.c_id, o.total
from customer c,
(select o_c_id, count(*) total from orders group by o_c_id) o
where c.c_id=o.o_c_id and c.c_last='BILL'
order by c.c_last
```

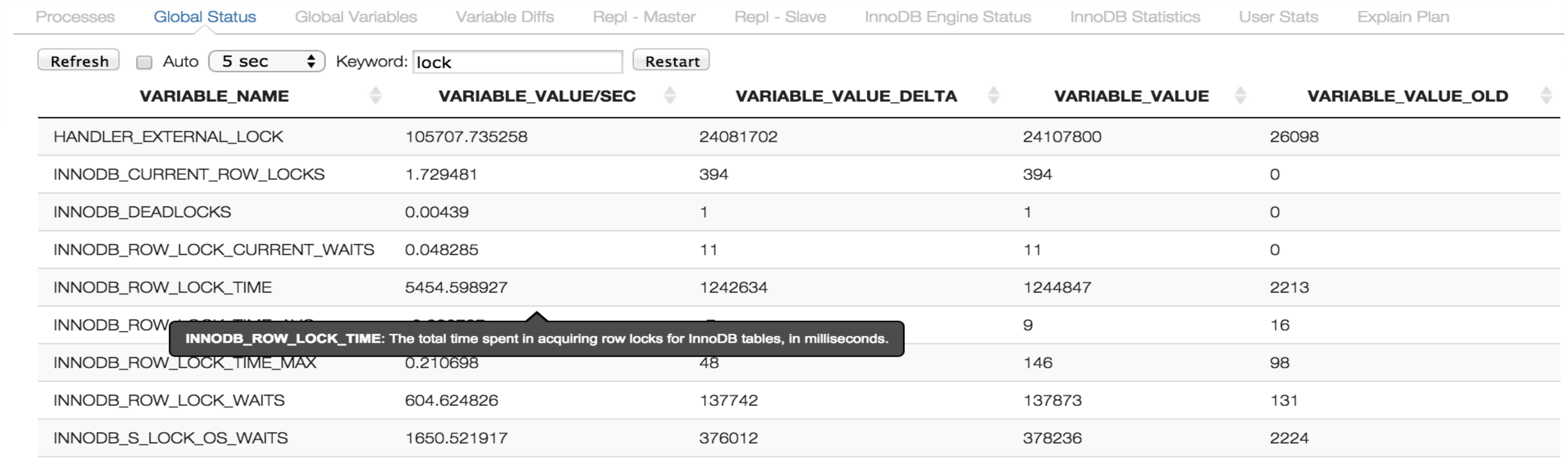
ID	SELECT_TYPE	TABLE	TYPE	POSSIBLE_KEYS	KEY	KEY_LEN	REF	ROWS	EXTRA
1	PRIMARY	c	index		c_w_id	110		594259	Using where; Using index
1	PRIMARY		ref	<auto_key0>		5	dbt2.c.c_id	10	
2	DERIVED	orders	index	o_w_id	o_w_id	17		1405880	Using index; Using temporary; Using filesort
									Cost: 0.000000

Showing 1 to 4 of 4 entries

```
Plan Tree
  1  ordering_operation: (using_filesort: false)
    nested_loop
      table: c (access_type:index, rows: 594259, filtered: 100)
        key: c_w_id, key_length: 110, used_key_parts: ["c_w_id","c_d_id","c_last","c_first"], ref: undefined
        using_index: true
        attached_condition: (`dbt2`.`c`.`c_last` = 'BILL')
      table: o (access_type:ref, rows: 10, filtered: 100)
        possible_keys: ["<auto_key0>"]
        key: <auto_key0>, key_length: 5, used_key_parts: ["o_c_id"], ref: ["dbt2.c.c_id"]
        materialized_from_subquery: (using_temporary_table:true, dependent: false, cacheable: true)
          2  grouping_operation: (using_temporary_table:true, using_filesort: true)
            table: orders (access_type: index, rows: 1405880, filtered: 100)
              possible_keys: ["o_w_id"]
              key: o_w_id, key_length: 17, used_key_parts: ["o_w_id","o_d_id","o_c_id","o_id"], ref: undefined
              using_index: true
```

- Parsed and displayed in tree structure for easy understanding the rich information.
- Bonus: comparing two plan formats can give us better understanding of the old format.

Global Status



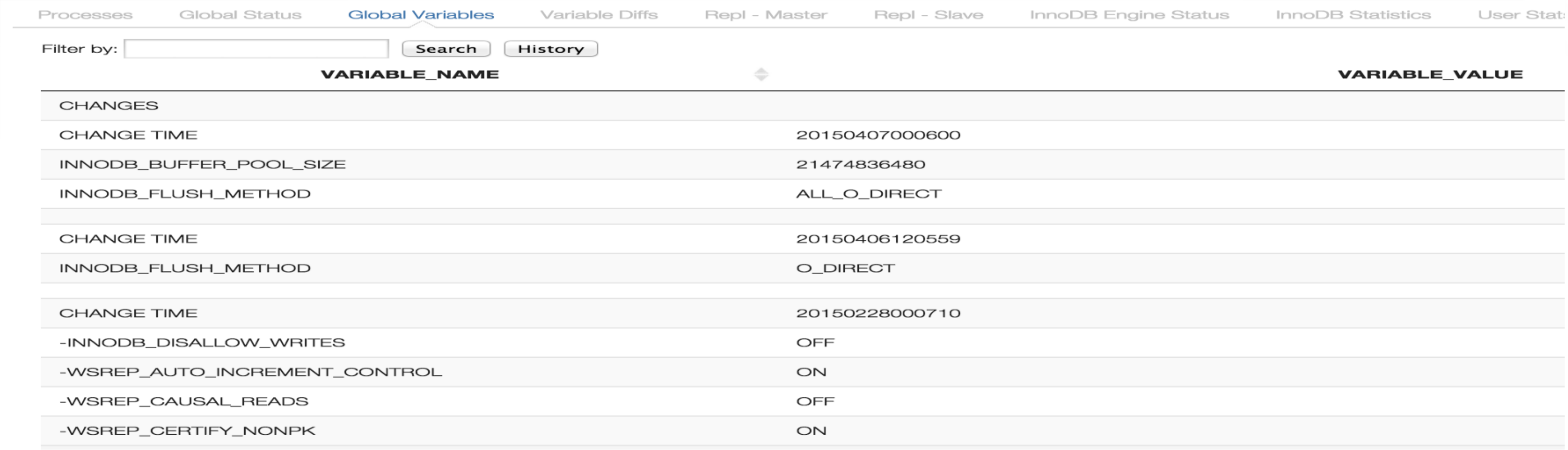
Keyword: lock

VARIABLE_NAME	VARIABLE_VALUE/SEC	VARIABLE_VALUE_DELTA	VARIABLE_VALUE	VARIABLE_VALUE_OLD
HANDLER_EXTERNAL_LOCK	105707.735258	24081702	24107800	26098
INNODB_CURRENT_ROW_LOCKS	1.729481	394	394	0
INNODB_DEADLOCKS	0.00439	1	1	0
INNODB_ROW_LOCK_CURRENT_WAITS	0.048285	11	11	0
INNODB_ROW_LOCK_TIME	5454.598927	1242634	1244847	2213
INNODB_ROW_LOCK_TIME_AVG	[illegible]	[illegible]	9	16
INNODB_ROW_LOCK_TIME_MAX	0.210698	48	146	98
INNODB_ROW_LOCK_WAITS	604.624826	137742	137873	131
INNODB_S_LOCK_OS_WAITS	1650.521917	376012	378236	2224

INNODB_ROW_LOCK_TIME: The total time spent in acquiring row locks for InnoDB tables, in milliseconds.

- Keyword filtering to view only concerned status variables.
- Auto refresh or manual refresh to see changes and change rates.
- Context help to assist understanding of the status variables.

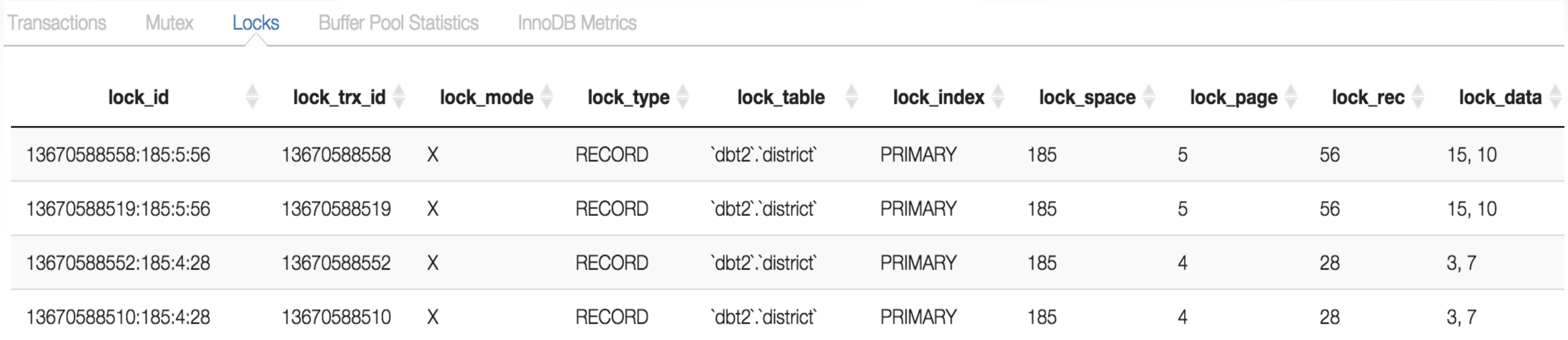
Configuration Management



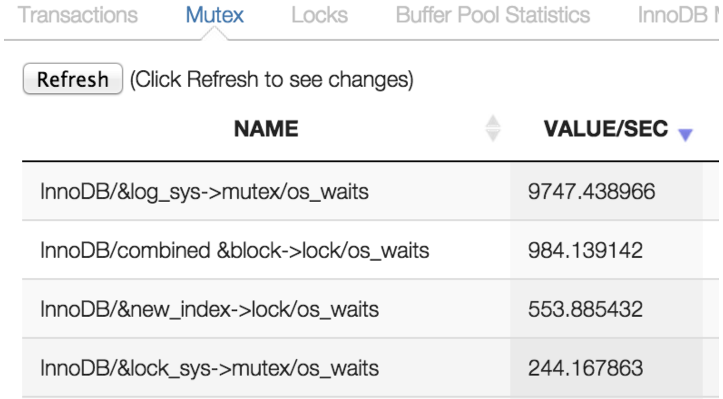
VARIABLE_NAME	VARIABLE_VALUE
CHANGES	
CHANGE TIME	20150407000600
INNODB_BUFFER_POOL_SIZE	21474836480
INNODB_FLUSH_METHOD	ALL_O_DIRECT
CHANGE TIME	20150406120559
INNODB_FLUSH_METHOD	O_DIRECT
CHANGE TIME	20150228000710
-INNODB_DISALLOW_WRITES	OFF
-WSREP_AUTO_INCREMENT_CONTROL	ON
-WSREP_CAUSAL_READS	OFF
-WSREP_CERTIFY_NONPK	ON

- Configuration consistency checks and variances when analyzing performance issues
- Lookup by partial keyword with links to MySQL references
- Change History Tracking.
- Compare parameters between database servers

InnoDB Statistics

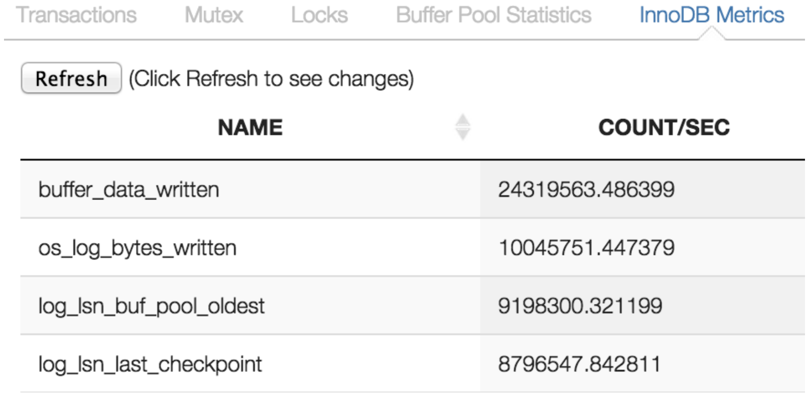


lock_id	lock_trx_id	lock_mode	lock_type	lock_table	lock_index	lock_space	lock_page	lock_rec	lock_data
13670588558:185:5:56	13670588558	X	RECORD	\`dbt2\`.\`district\`	PRIMARY	185	5	56	15, 10
13670588519:185:5:56	13670588519	X	RECORD	\`dbt2\`.\`district\`	PRIMARY	185	5	56	15, 10
13670588552:185:4:28	13670588552	X	RECORD	\`dbt2\`.\`district\`	PRIMARY	185	4	28	3, 7
13670588510:185:4:28	13670588510	X	RECORD	\`dbt2\`.\`district\`	PRIMARY	185	4	28	3, 7



(Click Refresh to see changes)

NAME	VALUE/SEC
InnoDB/&log_sys->mutex/os_waits	9747.438966
InnoDB/combined &block->lock/os_waits	984.139142
InnoDB/&new_index->lock/os_waits	553.885432
InnoDB/&lock_sys->mutex/os_waits	244.167863

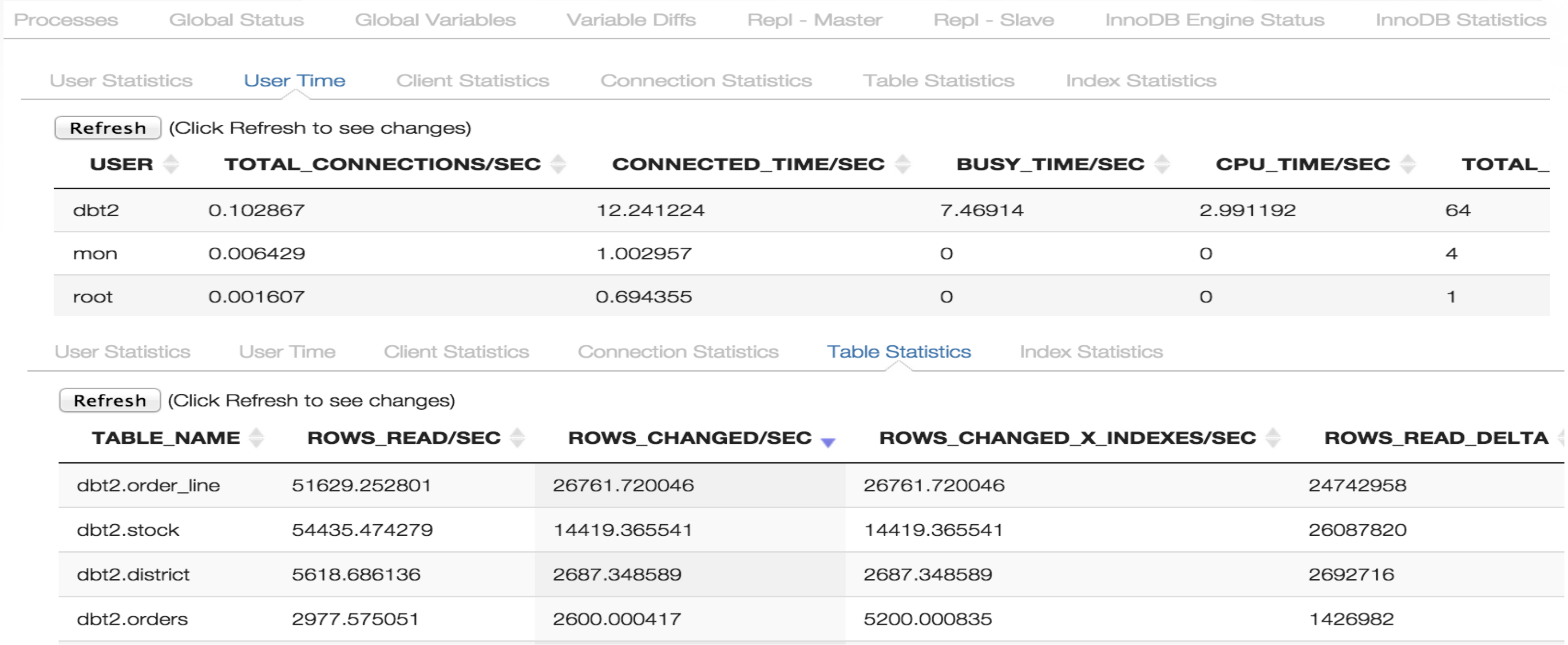


(Click Refresh to see changes)

NAME	COUNT/SEC
buffer_data_written	24319563.486399
os_log_bytes_written	10045751.447379
log_lsn_buf_pool_oldest	9198300.321199
log_lsn_last_checkpoint	8796547.842811

- Analyze performance issues, such as locks and mutexes
- Mutex statistics to understand contentions

User Statistics



USER	TOTAL_CONNECTIONS/SEC	CONNECTED_TIME/SEC	BUSY_TIME/SEC	CPU_TIME/SEC	TOTAL_
dbt2	0.102867	12.241224	7.46914	2.991192	64
mon	0.006429	1.002957	0	0	4
root	0.001607	0.694355	0	0	1

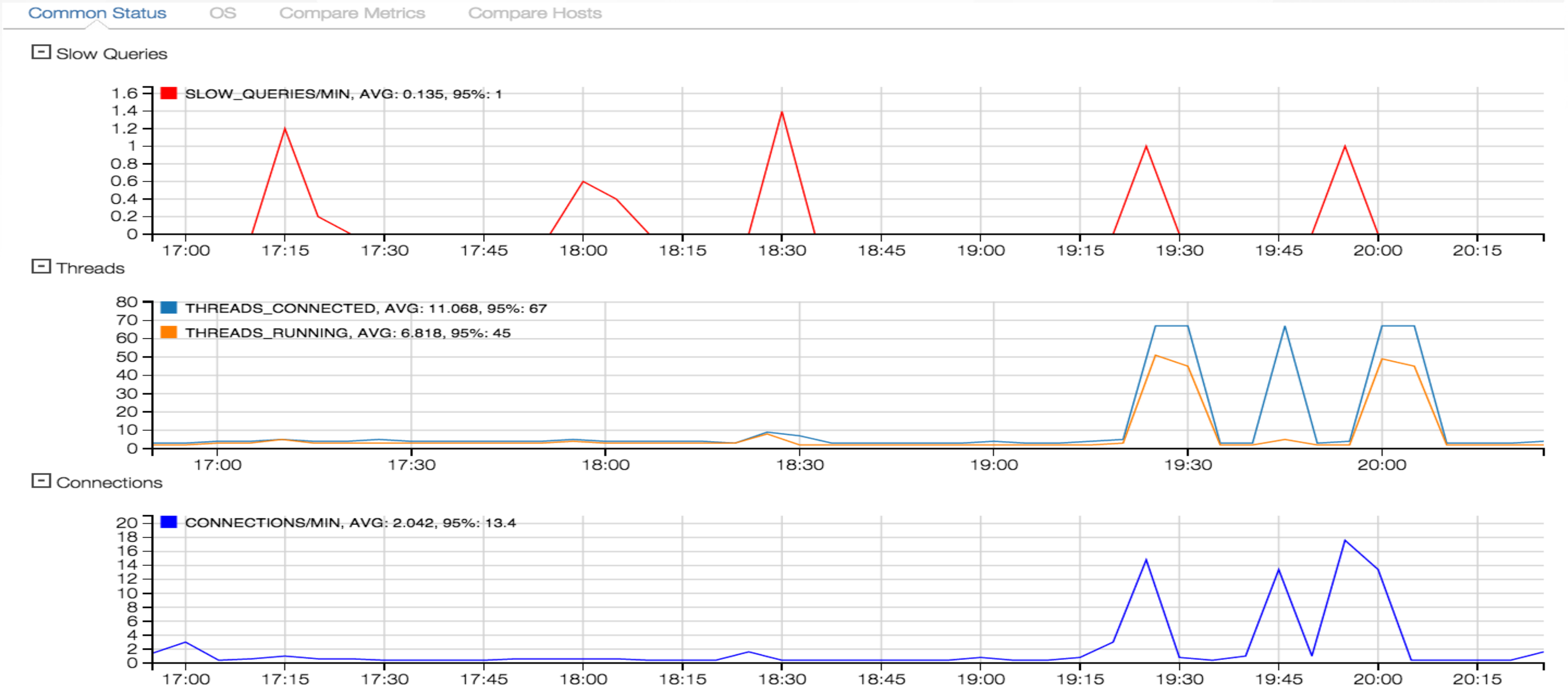
TABLE_NAME	ROWS_READ/SEC	ROWS_CHANGED/SEC	ROWS_CHANGED_X_INDEXES/SEC	ROWS_READ_DELTA
dbt2.order_line	51629.252801	26761.720046	26761.720046	24742958
dbt2.stock	54435.474279	14419.365541	14419.365541	26087820
dbt2.district	5618.686136	2687.348589	2687.348589	2692716
dbt2.orders	2977.575051	2600.000417	5200.000835	1426982

- When available, user statistics provide very useful time metrics, especially at per user level to identify hot users.
- Table statistics can also help to identify hot tables.

Metrics Gathering And Display

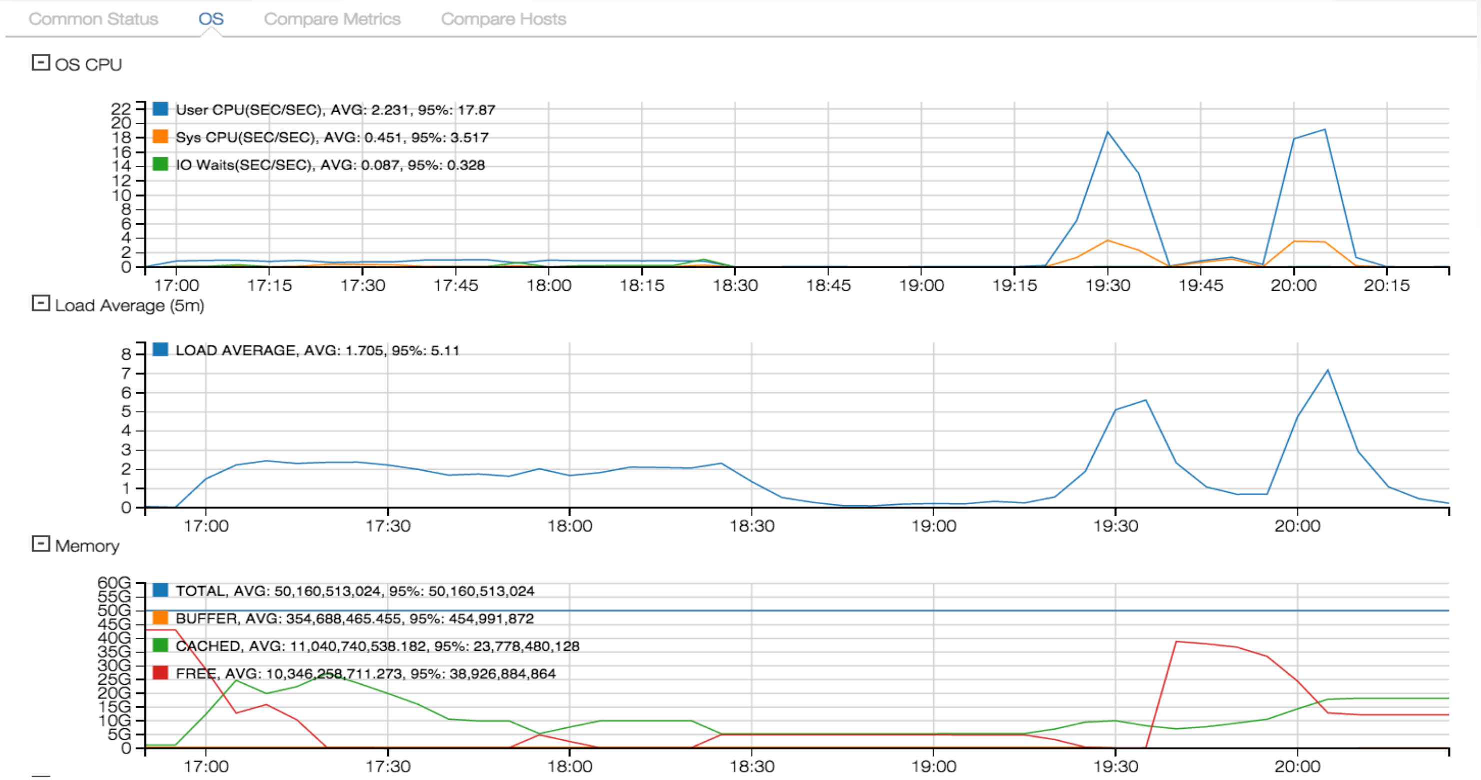
- Metrics are gathered from all managed servers based on configurable interval.
- Metrics are stored in either embedded Java DerbyDB for very small deployment or MySQL database for more formal deployment. concerned metrics are grouped and metrics from a single group are stored in a single table.
- Metrics sources:
  - information_schema, especially global status, for MySQL,
  - SNMP for OS level data when available
  - User defined.
- Predefined metrics:
  - MySQL common status, command, InnoDB, replication status
  - InnoDB Mutex (optional)
  - SNMP: system, disk, network, storage
  - Additional metrics can be defined and associated with individual server group or server, using global status variables, or customized SQL statements.

Metrics Charts – Common Global Status



- Periodically poll global status, InnoDB mutex and user defined metrics
- Metrics are stored in built-in embedded Java DB for a small deployment or in MySQL DB for a large deployment

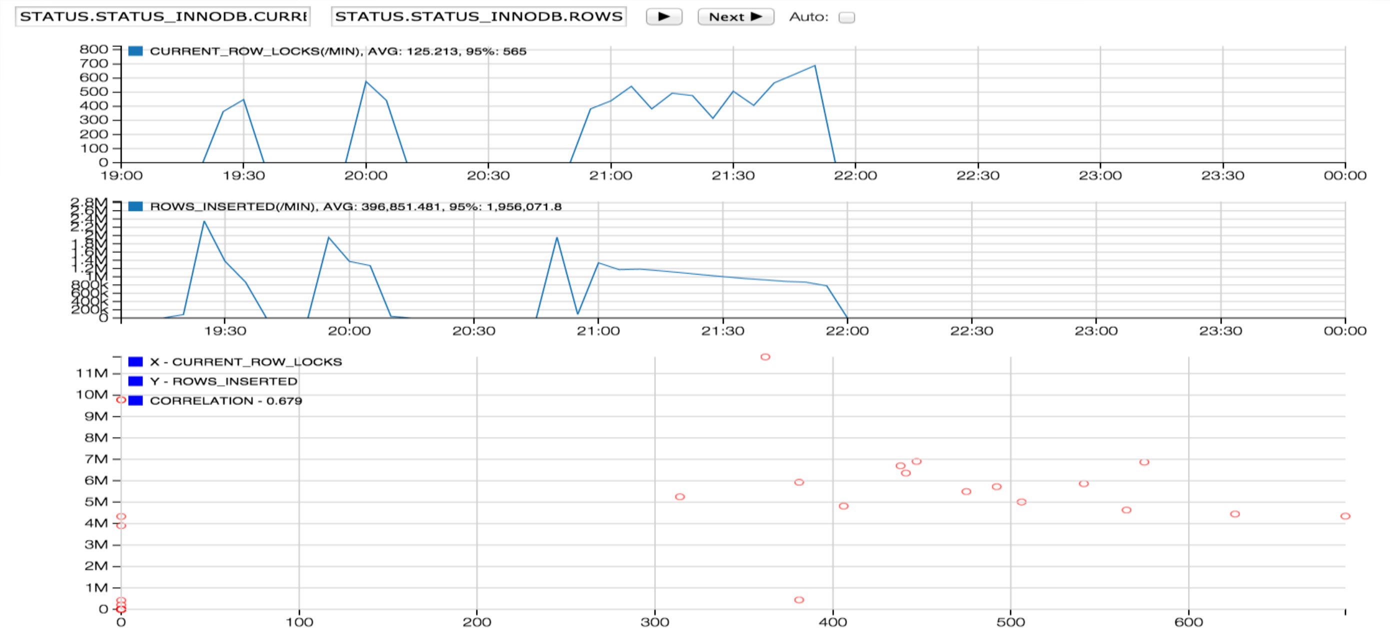
Metrics Charts – OS using SNMP



Common Status
OS
Compare Metrics
Compare Hosts
⊟ OS CPU
🟦 User CPU(SEC/SEC), AVG: 2.231, 95%: 17.87
🟧 Sys CPU(SEC/SEC), AVG: 0.451, 95%: 3.517
🟩 IO Waits(SEC/SEC), AVG: 0.087, 95%: 0.328
22
20
18
16
14
12
10
8
6
4
2
0
17:00
17:15
17:30
17:45
18:00
18:15
18:30
18:45
19:00
19:15
19:30
19:45
20:00
20:15
⊟ Load Average (5m)
🟦 LOAD AVERAGE, AVG: 1.705, 95%: 5.11
8
7
6
5
4
3
2
1
0
17:00
17:30
18:00
18:30
19:00
19:30
20:00
⊟ Memory
🟦 TOTAL, AVG: 50,160,513,024, 95%: 50,160,513,024
🟧 BUFFER, AVG: 354,688,465.455, 95%: 454,991,872
🟩 CACHED, AVG: 11,040,740,538.182, 95%: 23,778,480,128
🟥 FREE, AVG: 10,346,258,711.273, 95%: 38,926,884,864
60G
55G
50G
45G
40G
35G
30G
25G
20G
15G
10G
5G
0
17:00
17:30
18:00
18:30
19:00
19:30
20:00

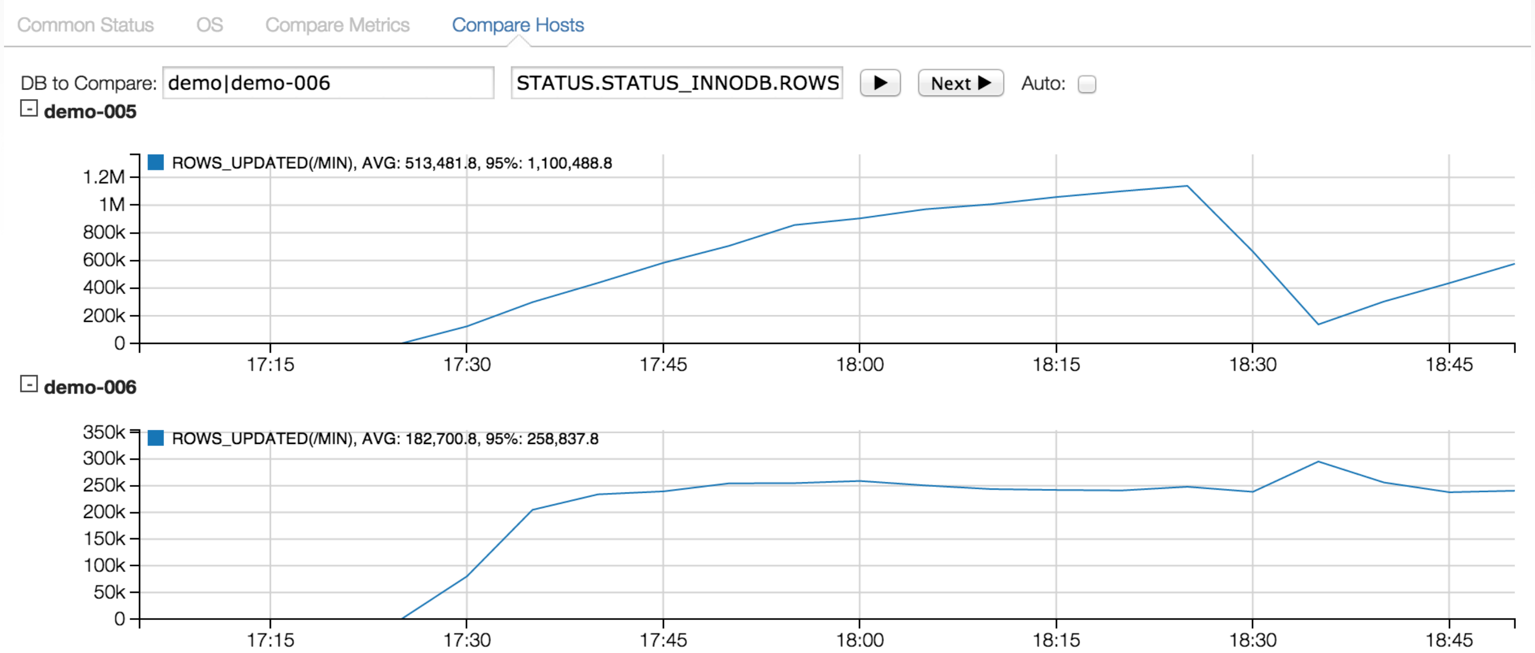
- OS level metrics are polled from SNMP
- Metrics include CPU, Load Average, Context switches, Interrupts, IO Waits, Disk, Memory Usage, network and storage usages, etc.

Metrics Charts: Single Chart Or Comparison



- Display chart for any available metric.
- Compare two metrics of the same server during the same period to identify correlations, which frequently help to identify root cause during troubleshooting.
- Auto play option to display the second metrics sequentially

Metrics Comparison between A Group Of Servers



DB to Compare: demo|demo-006

STATUS.STATUS_INNODB.ROWS

⊟ **demo-005**

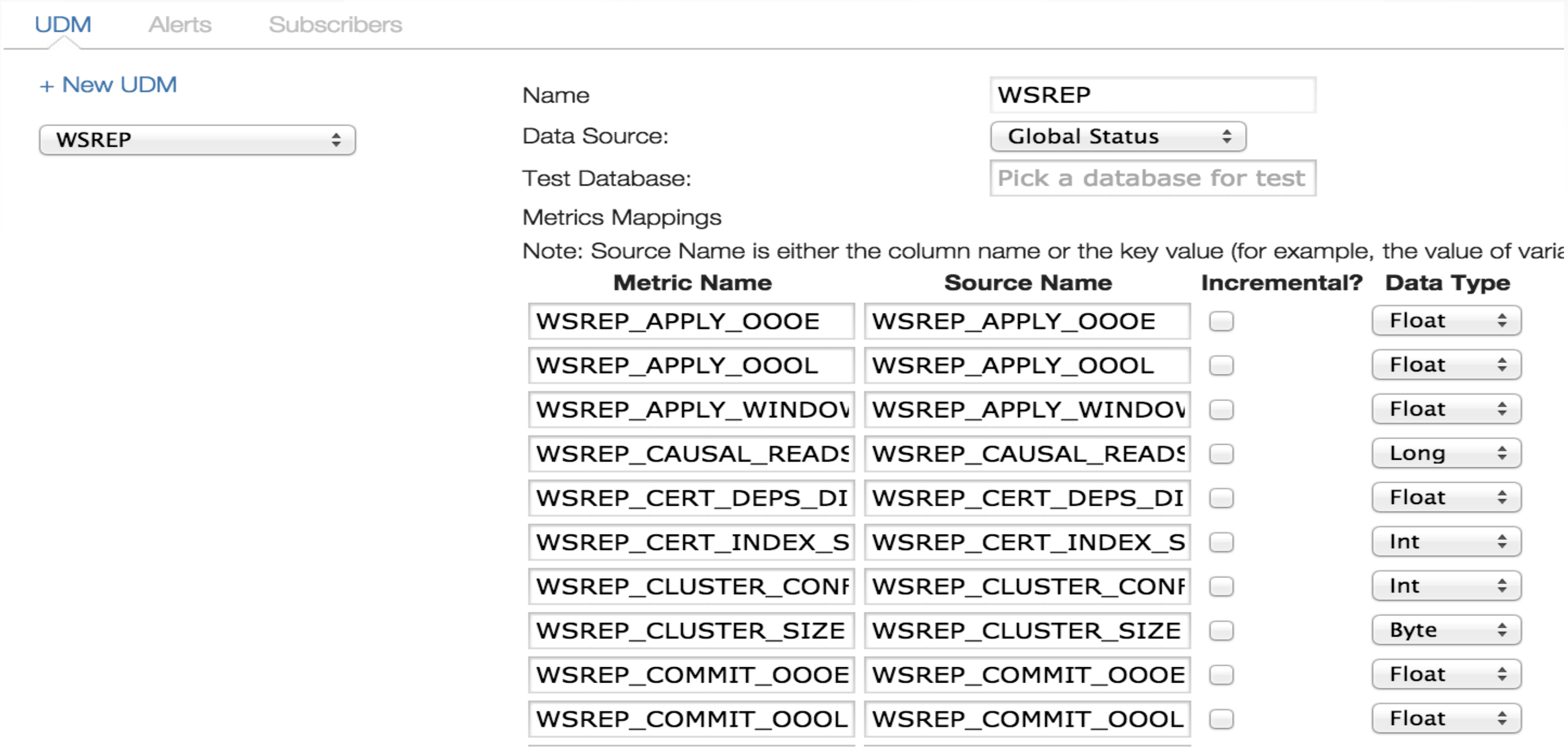
🟦 ROWS_UPDATED(/MIN), AVG: 513,481.8, 95%: 1,100,488.8

⊟ **demo-006**

🟦 ROWS_UPDATED(/MIN), AVG: 182,700.8, 95%: 258,837.8

- Metrics can be viewed and compared on a pair of servers or multiple servers of the same group.
- This feature can be used to understand how loads are balanced, or capacity differences between two servers.
- Above sample is a master/slave comparison. Replication cannot catch up the very high update rates on the master.

User Defined Metrics (UDM)



Name	WSREP
Data Source:	Global Status
Test Database:	Pick a database for test

Metrics Mappings

Note: Source Name is either the column name or the key value (for example, the value of varia

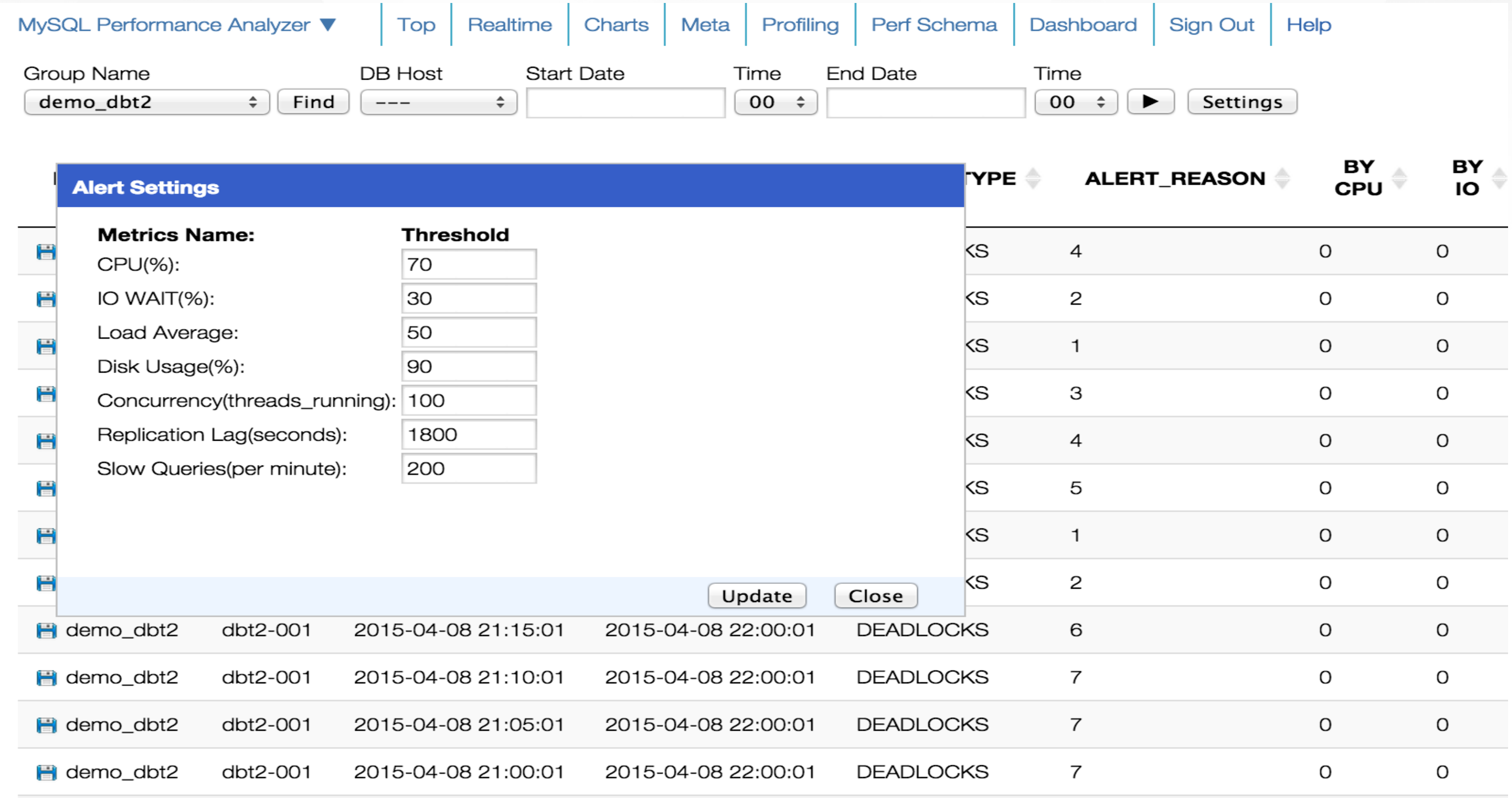
Metric Name	Source Name	Incremental?	Data Type
WSREP_APPLY_OOOE	WSREP_APPLY_OOOE	☐	Float
WSREP_APPLY_OOOL	WSREP_APPLY_OOOL	☐	Float
WSREP_APPLY_WINDOV	WSREP_APPLY_WINDOV	☐	Float
WSREP_CAUSAL_READS	WSREP_CAUSAL_READS	☐	Long
WSREP_CERT_DEPS_DI	WSREP_CERT_DEPS_DI	☐	Float
WSREP_CERT_INDEX_S	WSREP_CERT_INDEX_S	☐	Int
WSREP_CLUSTER_CONF	WSREP_CLUSTER_CONF	☐	Int
WSREP_CLUSTER_SIZE	WSREP_CLUSTER_SIZE	☐	Byte
WSREP_COMMIT_OOOE	WSREP_COMMIT_OOOE	☐	Float
WSREP_COMMIT_OOOL	WSREP_COMMIT_OOOL	☐	Float

- Customized metrics can be added either using status variables from global status, which are not included in the built in metrics, or using customized SQL statement.
- Manual setup is required to associate concerned servers or server groups with any UDM.
- Current implementation will store all metrics defined within one UDM in a single table.

Anomaly Detections and Alerts

- Anomalies will be checked for a set of predefined metrics against thresholds. Thresholds can be adjusted at server group level or host level.
- When anomalies are detected, forensic data will be gathered and logged, such as process lists, InnoDB engine status, innodb locks, etc.
- Alert detail reports can be viewed and downloaded from dashboard and Alert page.
- Alerts will be logged and notifications can be sent out using email and web notifications.
- Predefined alerts:
  - CPU, Load Average, IO Waits, Running Threads, Replication Status and lag, Slow Query Count, Connection Failure, Deadlocks and Disk Usages
- Additional customized alerts can be defined and attached to concerned database server, using either a SQL statement, or against metrics already defined, or just against any global status variable.

Alerts and Settings



Group Name: demo_dbt2
DB Host: ---
Start Date:
Time: 00
End Date:
Time: 00

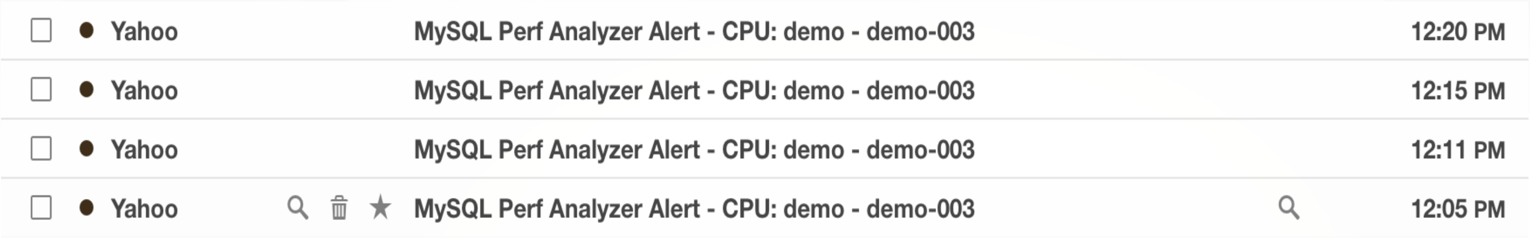
Alert Settings

Metrics Name:	Threshold
CPU(%):	70
IO WAIT(%):	30
Load Average:	50
Disk Usage(%):	90
Concurrency(threads_running):	100
Replication Lag(seconds):	1800
Slow Queries(per minute):	200

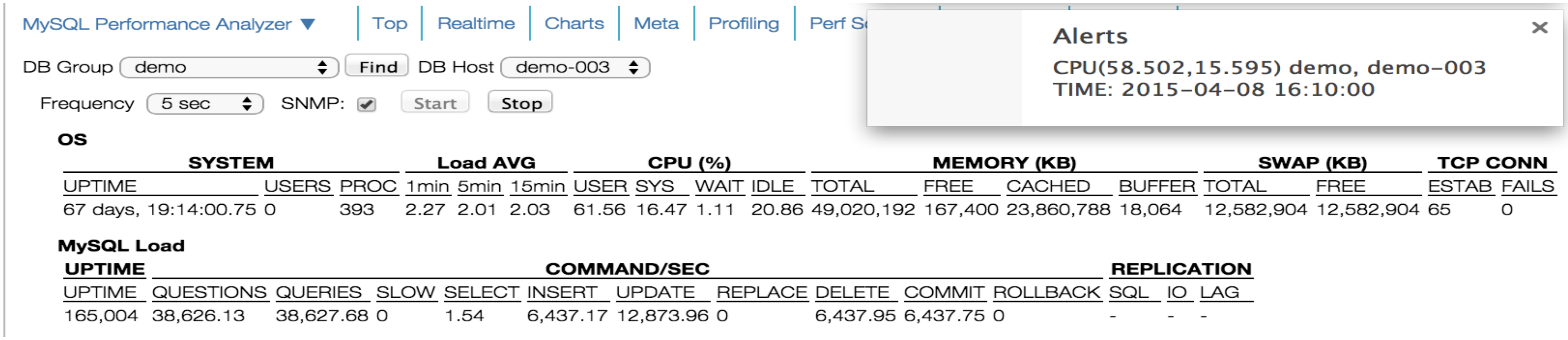
					YPE	ALERT_REASON	BY CPU	BY IO
💾					KS	4	0	0
💾					KS	2	0	0
💾					KS	1	0	0
💾					KS	3	0	0
💾					KS	4	0	0
💾					KS	5	0	0
💾					KS	1	0	0
💾					KS	2	0	0
💾	demo_dbt2	dbt2-001	2015-04-08 21:15:01	2015-04-08 22:00:01	DEADLOCKS	6	0	0
💾	demo_dbt2	dbt2-001	2015-04-08 21:10:01	2015-04-08 22:00:01	DEADLOCKS	7	0	0
💾	demo_dbt2	dbt2-001	2015-04-08 21:05:01	2015-04-08 22:00:01	DEADLOCKS	7	0	0
💾	demo_dbt2	dbt2-001	2015-04-08 21:00:01	2015-04-08 22:00:01	DEADLOCKS	7	0	0

- All alerts for past 24 hours will be displayed in dashboard after login.
- Alerts for all servers, an individual server group, or a single host, can be accessed from Alert page.
- Thresholds can be configured at server group or host level.

Alert Notifications



□	●	Yahoo	MySQL Perf Analyzer Alert - CPU: demo - demo-003	12:20 PM
□	●	Yahoo	MySQL Perf Analyzer Alert - CPU: demo - demo-003	12:15 PM
□	●	Yahoo	MySQL Perf Analyzer Alert - CPU: demo - demo-003	12:11 PM
□	●	Yahoo	MySQL Perf Analyzer Alert - CPU: demo - demo-003	12:05 PM



DB Group demo

DB Host demo-003

Frequency 5 sec SNMP: ☑

Alerts

CPU(58.502,15.595) demo, demo-003
TIME: 2015-04-08 16:10:00

OS

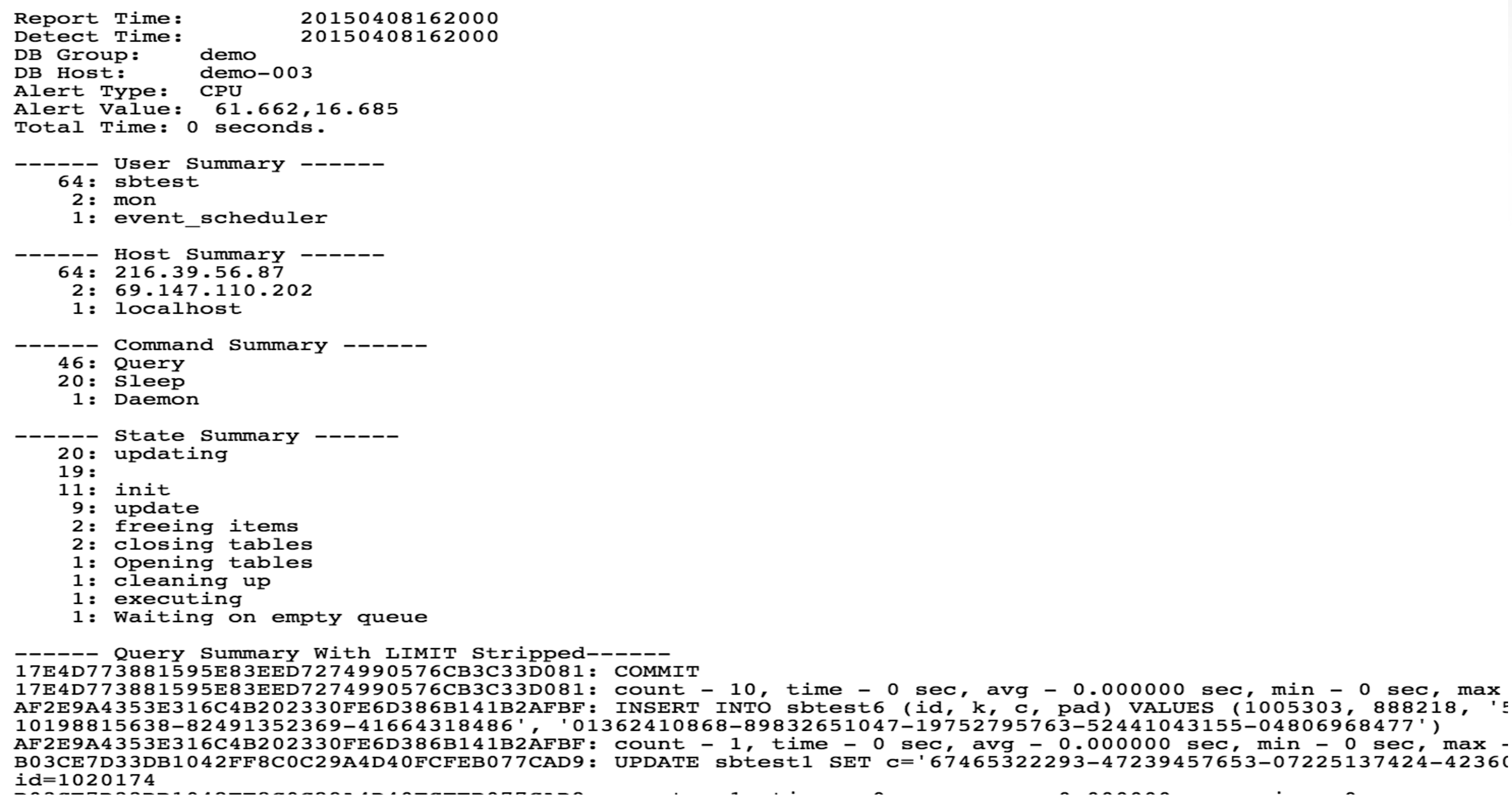
SYSTEM			Load AVG			CPU (%)				MEMORY (KB)				SWAP (KB)		TCP CONN	
UPTIME	USERS	PROC	1min	5min	15min	USER	SYS	WAIT	IDLE	TOTAL	FREE	CACHED	BUFFER	TOTAL	FREE	ESTAB	FAILS
67 days, 19:14:00.75	0	393	2.27	2.01	2.03	61.56	16.47	1.11	20.86	49,020,192	167,400	23,860,788	18,064	12,582,904	12,582,904	65	0

MySQL Load

UPTIME	COMMAND/SEC										REPLICATION		
UPTIME	QUESTIONS	QUERIES	SLOW	SELECT	INSERT	UPDATE	REPLACE	DELETE	COMMIT	ROLLBACK	SQL	IO	LAG
165,004	38,626.13	38,627.68	0	1.54	6,437.17	12,873.96	0	6,437.95	6,437.75	0	-	-	-

- Alert notifications will be sent to email if configured, with minimum information.
- Web notification is also supported on modern browsers when the application is open.

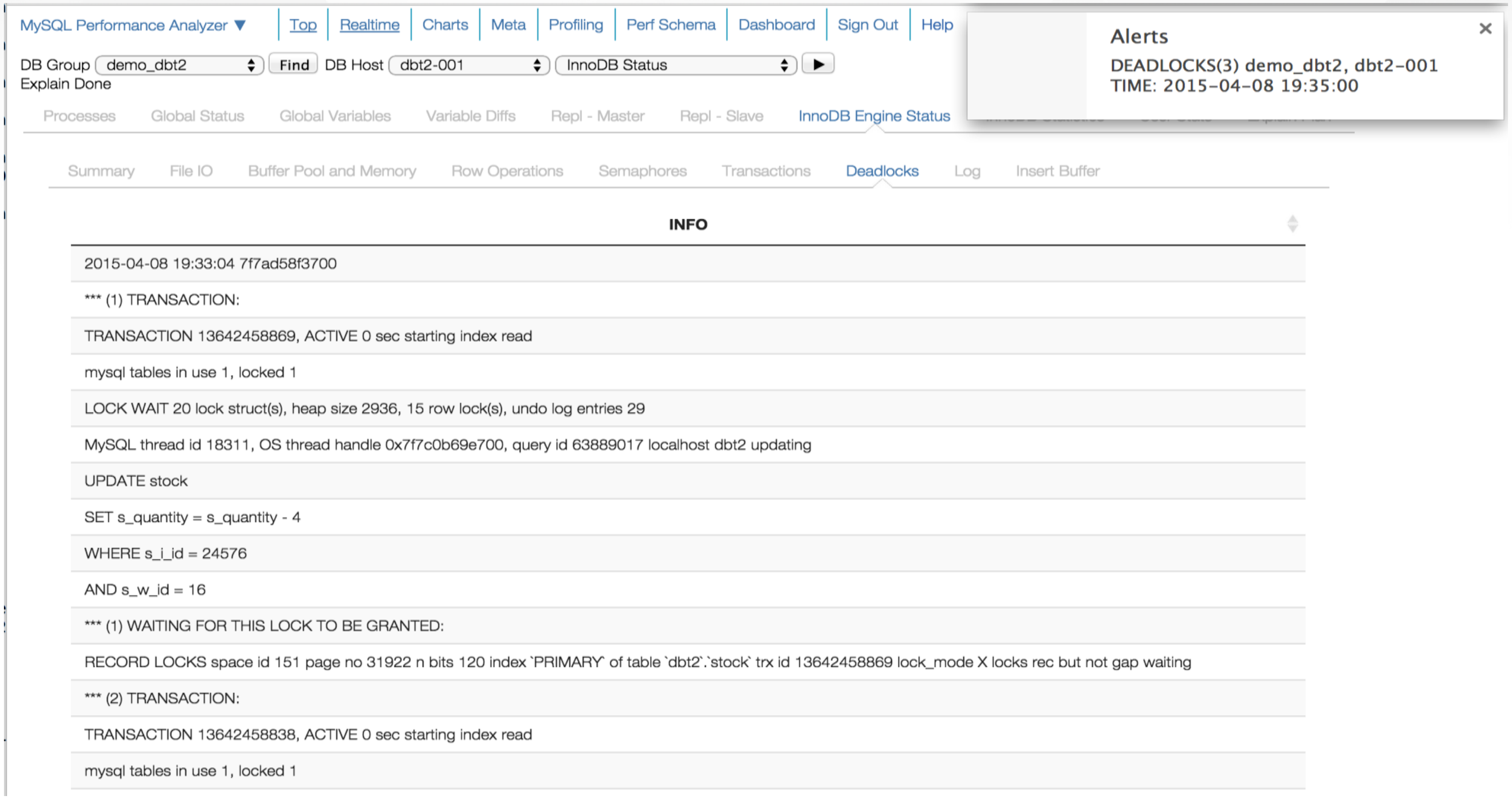
Alert Reports



```
Report Time:        20150408162000
Detect Time:        20150408162000
DB Group:     demo
DB Host:      demo-003
Alert Type:   CPU
Alert Value:   61.662,16.685
Total Time: 0 seconds.

------ User Summary ------
   64: sbtest
    2: mon
    1: event_scheduler

------ Host Summary ------
   64: 216.39.56.87
    2: 69.147.110.202
    1: localhost

------ Command Summary ------
   46: Query
   20: Sleep
    1: Daemon

------ State Summary ------
   20: updating
   19:
   11: init
    9: update
    2: freeing items
    2: closing tables
    1: Opening tables
    1: cleaning up
    1: executing
    1: Waiting on empty queue

------ Query Summary With LIMIT Stripped------
17E4D773881595E83EED7274990576CB3C33D081: COMMIT
17E4D773881595E83EED7274990576CB3C33D081: count - 10, time - 0 sec, avg - 0.000000 sec, min - 0 sec, max
AF2E9A4353E316C4B202330FE6D386B141B2AFBF: INSERT INTO sbtest6 (id, k, c, pad) VALUES (1005303, 888218, '5
10198815638-82491352369-41664318486', '01362410868-89832651047-19752795763-52441043155-04806968477')
AF2E9A4353E316C4B202330FE6D386B141B2AFBF: count - 1, time - 0 sec, avg - 0.000000 sec, min - 0 sec, max -
B03CE7D33DB1042FF8C0C29A4D40FCFEB077CAD9: UPDATE sbtest1 SET c='67465322293-47239457653-07225137424-4236
id=1020174
```

- For most of the alerts, an alert report will be generated with some forensic information.
- The information includes aggregated and original data from process list and InnoDB engine status, etc.

Deadlock Detection



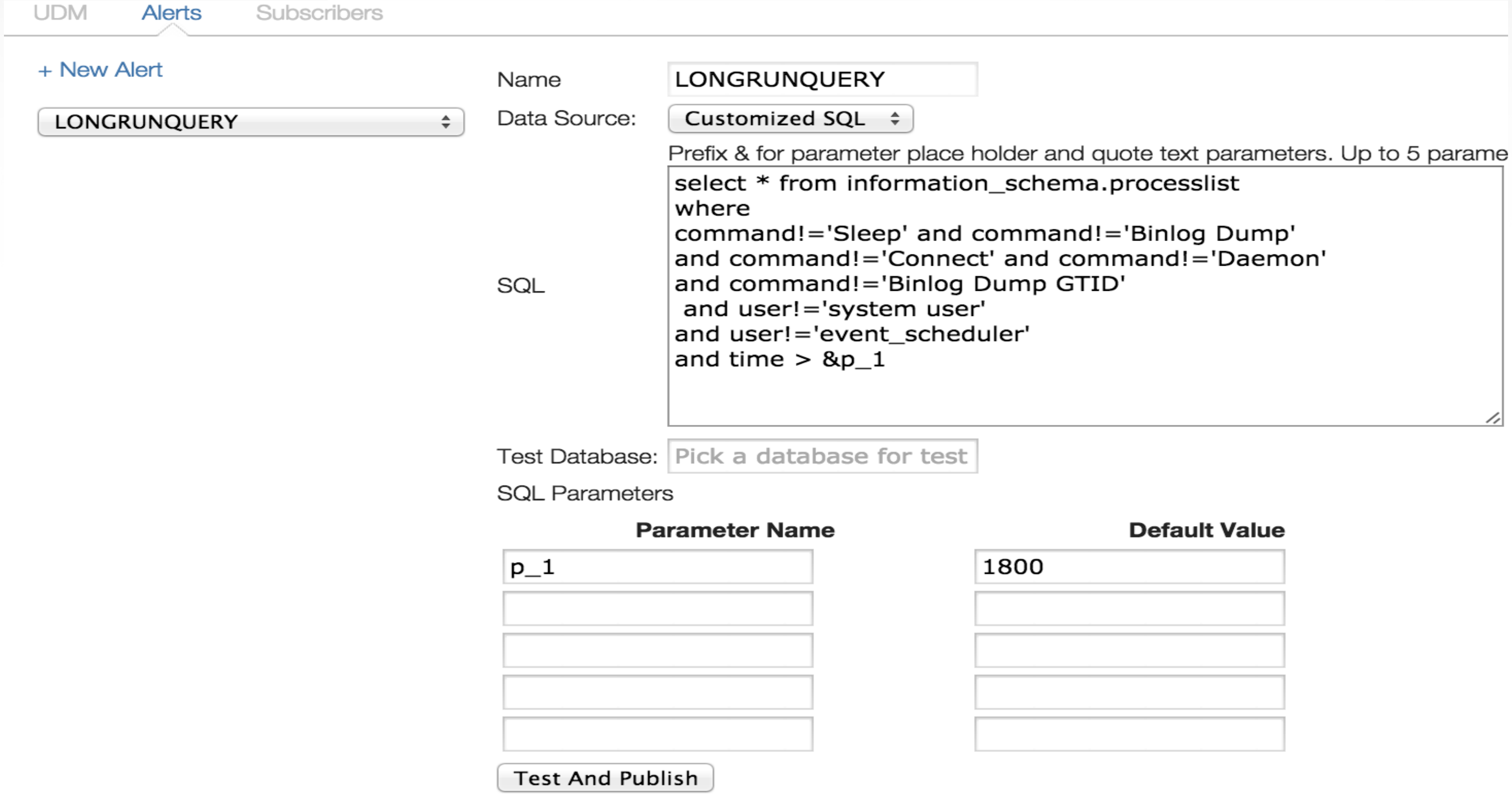
Alerts

DEADLOCKS(3) demo_dbt2, dbt2-001
TIME: 2015-04-08 19:35:00

```
2015-04-08 19:33:04 7f7ad58f3700
*** (1) TRANSACTION:
TRANSACTION 13642458869, ACTIVE 0 sec starting index read
mysql tables in use 1, locked 1
LOCK WAIT 20 lock struct(s), heap size 2936, 15 row lock(s), undo log entries 29
MySQL thread id 18311, OS thread handle 0x7f7c0b69e700, query id 63889017 localhost dbt2 updating
UPDATE stock
SET s_quantity = s_quantity - 4
WHERE s_i_id = 24576
AND s_w_id = 16
*** (1) WAITING FOR THIS LOCK TO BE GRANTED:
RECORD LOCKS space id 151 page no 31922 n bits 120 index `PRIMARY` of table `dbt2`.`stock` trx id 13642458869 lock_mode X locks rec but not gap waiting
*** (2) TRANSACTION:
TRANSACTION 13642458838, ACTIVE 0 sec starting index read
mysql tables in use 1, locked 1
```

- Deadlock detection is done by comparing INNODB_DEADLOCKS status variable (available in Percona server).
- When detected, an alert will be raised and logged. Detail can be found either from InnoDB engine status, or associated alert reports.

User Defined Alerts



LONGRUNQUERY

Name: LONGRUNQUERY

Data Source: Customized SQL

Prefix & for parameter place holder and quote text parameters. Up to 5 parame

SQL

```
select * from information_schema.processlist
where
command!='Sleep' and command!='Binlog Dump'
and command!='Connect' and command!='Daemon'
and command!='Binlog Dump GTID'
 and user!='system user'
and user!='event_scheduler'
and time > &p_1
```

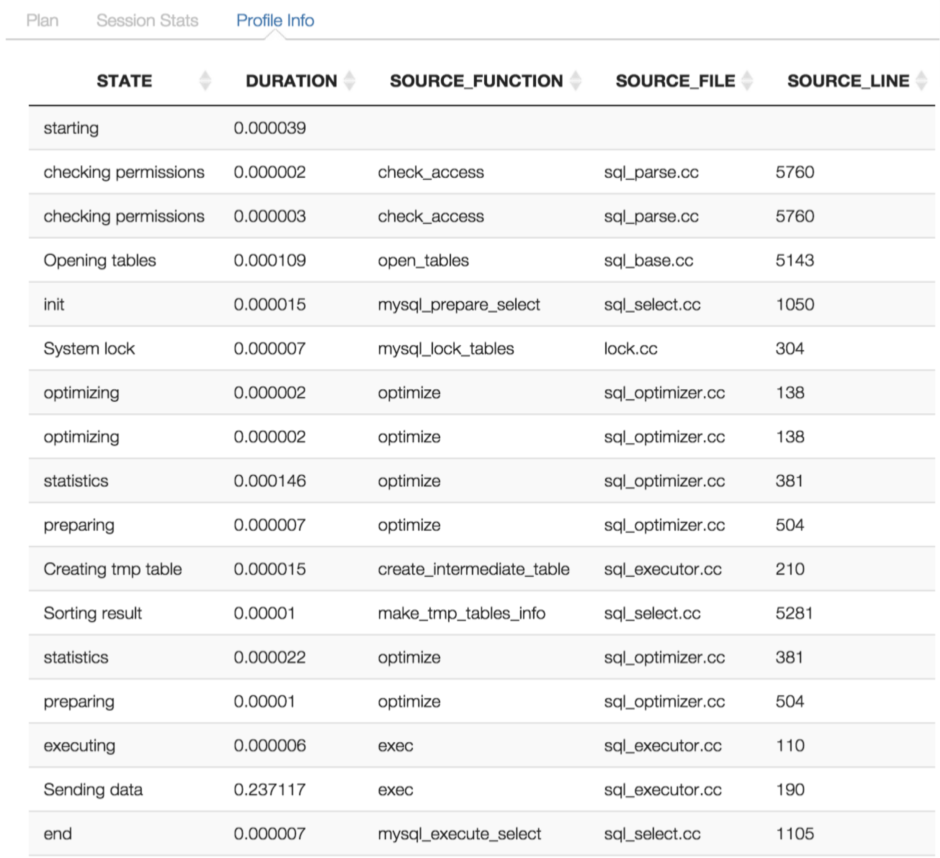
Test Database: Pick a database for test

SQL Parameters

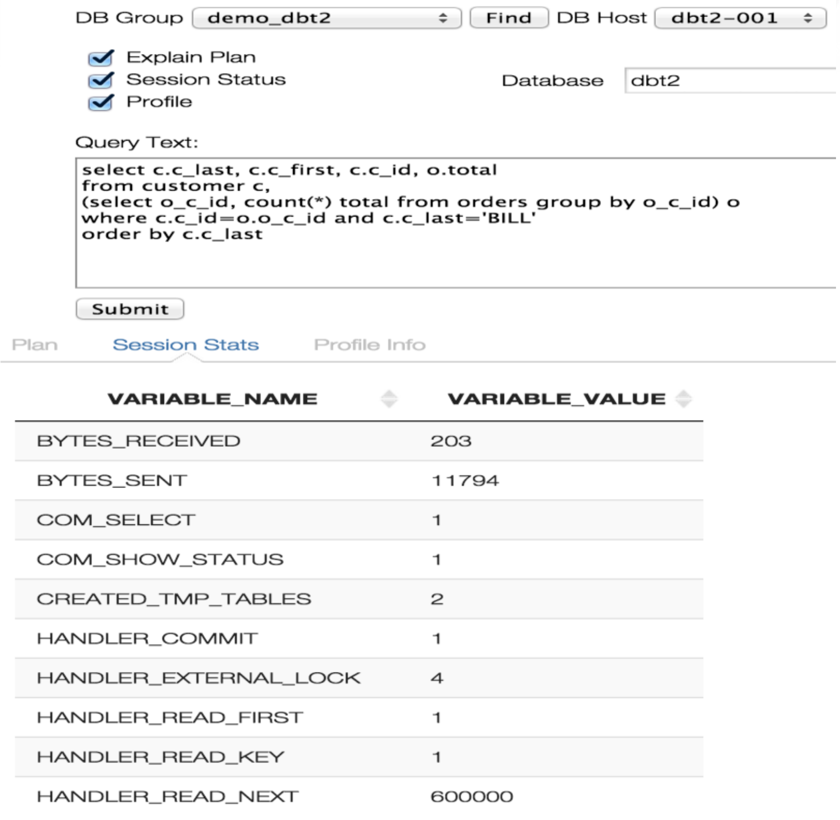
Parameter Name	Default Value
p_1	1800

- Customized alerts can be defined using SQL statements, global status variables or metrics gathered by the analyzer.
- Customized alerts will not be applied to all servers automatically. Requires manually setup to associate them with concerned servers or server groups.

Profiling and Tuning



STATE	DURATION	SOURCE_FUNCTION	SOURCE_FILE	SOURCE_LINE
starting	0.000039			
checking permissions	0.000002	check_access	sql_parse.cc	5760
checking permissions	0.000003	check_access	sql_parse.cc	5760
Opening tables	0.000109	open_tables	sql_base.cc	5143
init	0.000015	mysql_prepare_select	sql_select.cc	1050
System lock	0.000007	mysql_lock_tables	lock.cc	304
optimizing	0.000002	optimize	sql_optimizer.cc	138
optimizing	0.000002	optimize	sql_optimizer.cc	138
statistics	0.000146	optimize	sql_optimizer.cc	381
preparing	0.000007	optimize	sql_optimizer.cc	504
Creating tmp table	0.000015	create_intermediate_table	sql_executor.cc	210
Sorting result	0.00001	make_tmp_tables_info	sql_select.cc	5281
statistics	0.000022	optimize	sql_optimizer.cc	381
preparing	0.00001	optimize	sql_optimizer.cc	504
executing	0.000006	exec	sql_executor.cc	110
Sending data	0.237117	exec	sql_executor.cc	190
end	0.000007	mysql_execute_select	sql_select.cc	1105



DB Group: demo_dbt2
DB Host: dbt2-001

☑ Explain Plan
☑ Session Status
☑ Profile

Database: dbt2

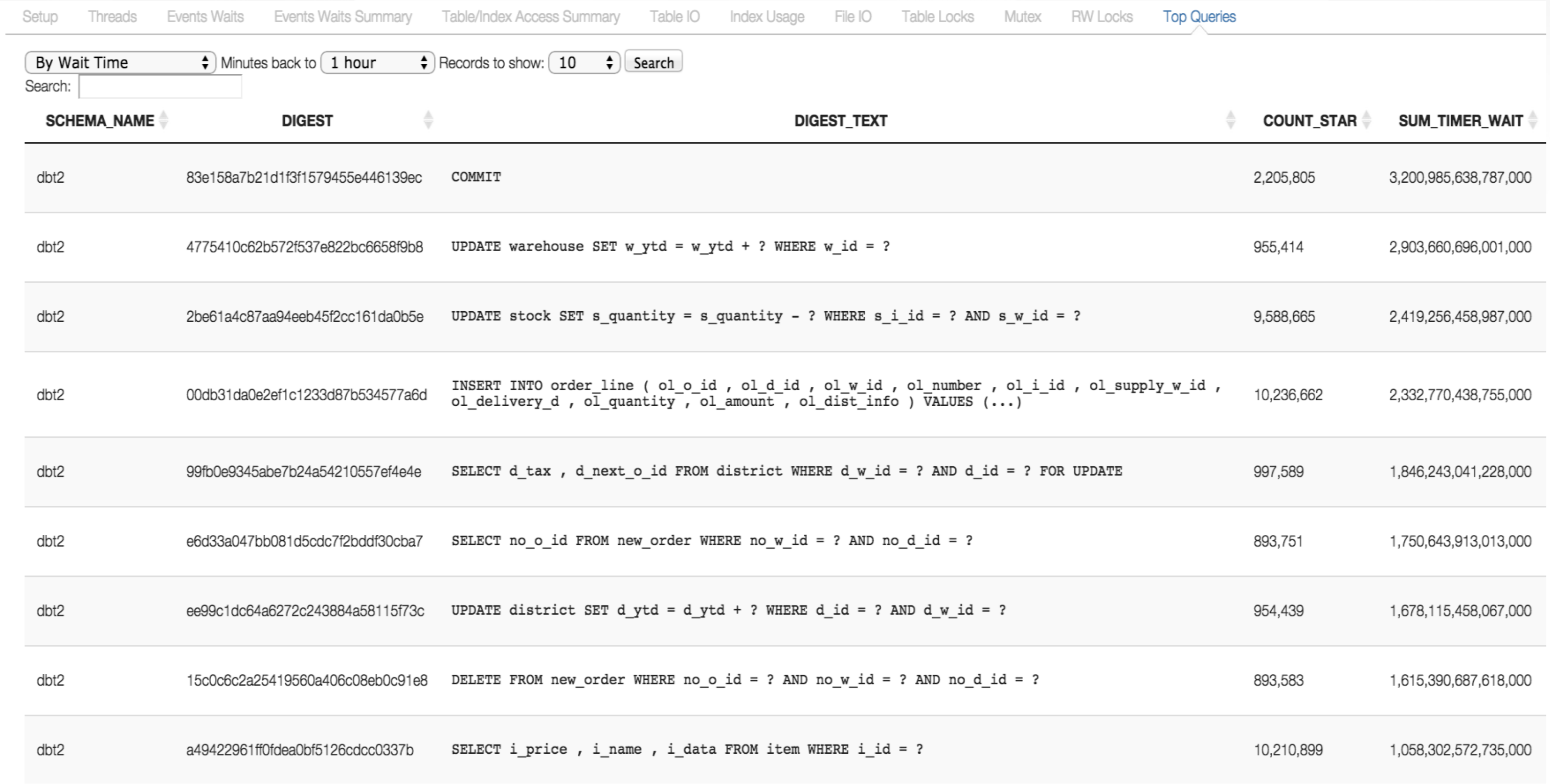
Query Text:

```
select c.c_last, c.c_first, c.c_id, o.total
from customer c,
(select o_c_id, count(*) total from orders group by o_c_id) o
where c.c_id=o.o_c_id and c.c_last='BILL'
order by c.c_last
```

VARIABLE_NAME	VARIABLE_VALUE
BYTES_RECEIVED	203
BYTES_SENT	11794
COM_SELECT	1
COM_SHOW_STATUS	1
CREATED_TMP_TABLES	2
HANDLER_COMMIT	1
HANDLER_EXTERNAL_LOCK	4
HANDLER_READ_FIRST	1
HANDLER_READ_KEY	1
HANDLER_READ_NEXT	600000

- A simple and safe interface to run explain plan, MySQL profiling, and execute MySQL SELECT statement.

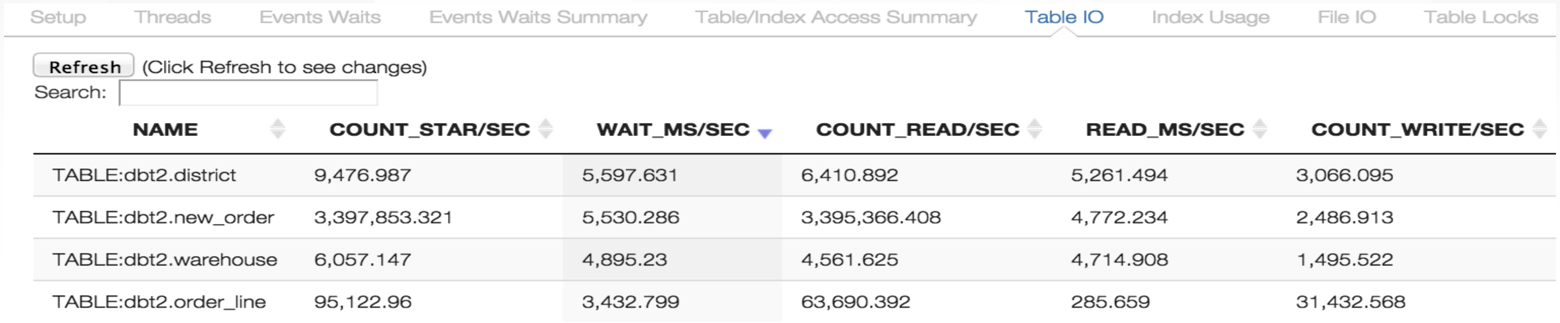
Performance Schema – Top Queries



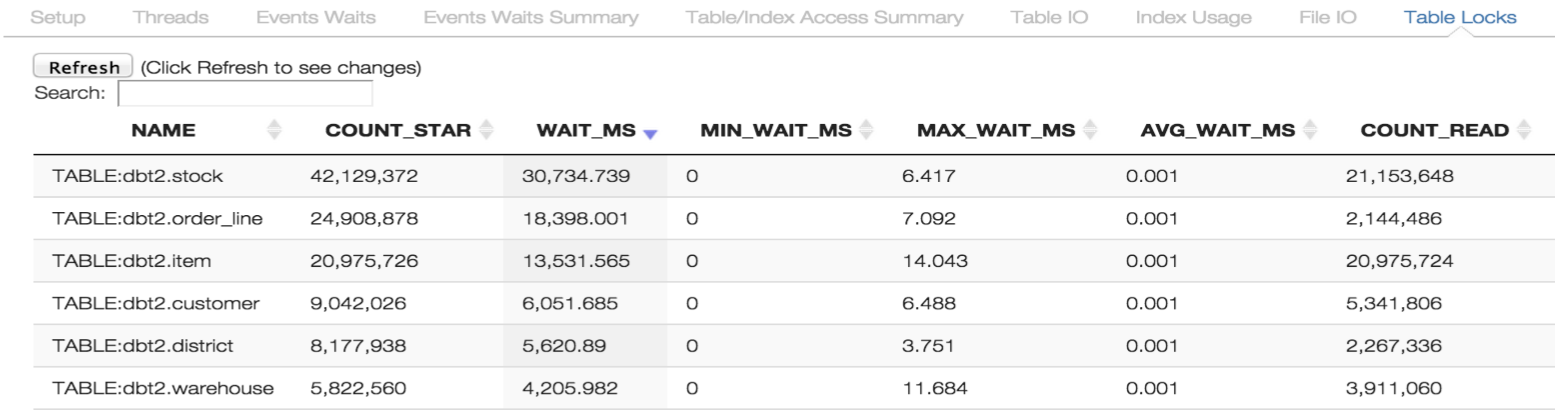
SCHEMA_NAME	DIGEST	DIGEST_TEXT	COUNT_STAR	SUM_TIMER_WAIT
dbt2	83e158a7b21d1f3f1579455e446139ec	COMMIT	2,205,805	3,200,985,638,787,000
dbt2	4775410c62b572f537e822bc6658f9b8	UPDATE warehouse SET w_ytd = w_ytd + ? WHERE w_id = ?	955,414	2,903,660,696,001,000
dbt2	2be61a4c87aa94eeb45f2cc161da0b5e	UPDATE stock SET s_quantity = s_quantity - ? WHERE s_i_id = ? AND s_w_id = ?	9,588,665	2,419,256,458,987,000
dbt2	00db31da0e2ef1c1233d87b534577a6d	INSERT INTO order_line (ol_o_id , ol_d_id , ol_w_id , ol_number , ol_i_id , ol_supply_w_id , ol_delivery_d , ol_quantity , ol_amount , ol_dist_info) VALUES (...)	10,236,662	2,332,770,438,755,000
dbt2	99fb0e9345abe7b24a54210557ef4e4e	SELECT d_tax , d_next_o_id FROM district WHERE d_w_id = ? AND d_id = ? FOR UPDATE	997,589	1,846,243,041,228,000
dbt2	e6d33a047bb081d5cdc7f2bddf30cba7	SELECT no_o_id FROM new_order WHERE no_w_id = ? AND no_d_id = ?	893,751	1,750,643,913,013,000
dbt2	ee99c1dc64a6272c243884a58115f73c	UPDATE district SET d_ytd = d_ytd + ? WHERE d_id = ? AND d_w_id = ?	954,439	1,678,115,458,067,000
dbt2	15c0c6c2a25419560a406c08eb0c91e8	DELETE FROM new_order WHERE no_o_id = ? AND no_w_id = ? AND no_d_id = ?	893,583	1,615,390,687,618,000
dbt2	a49422961ff0fdea0bf5126cdcc0337b	SELECT i_price , i_name , i_data FROM item WHERE i_id = ?	10,210,899	1,058,302,572,735,000

- Top queries by various criteria

Performance Schema – Hot Tables



NAME	COUNT_STAR/SEC	WAIT_MS/SEC	COUNT_READ/SEC	READ_MS/SEC	COUNT_WRITE/SEC
TABLE:dbt2.district	9,476.987	5,597.631	6,410.892	5,261.494	3,066.095
TABLE:dbt2.new_order	3,397,853.321	5,530.286	3,395,366.408	4,772.234	2,486.913
TABLE:dbt2.warehouse	6,057.147	4,895.23	4,561.625	4,714.908	1,495.522
TABLE:dbt2.order_line	95,122.96	3,432.799	63,690.392	285.659	31,432.568



NAME	COUNT_STAR	WAIT_MS	MIN_WAIT_MS	MAX_WAIT_MS	AVG_WAIT_MS	COUNT_READ
TABLE:dbt2.stock	42,129,372	30,734.739	0	6.417	0.001	21,153,648
TABLE:dbt2.order_line	24,908,878	18,398.001	0	7.092	0.001	2,144,486
TABLE:dbt2.item	20,975,726	13,531.565	0	14.043	0.001	20,975,724
TABLE:dbt2.customer	9,042,026	6,051.685	0	6.488	0.001	5,341,806
TABLE:dbt2.district	8,177,938	5,620.89	0	3.751	0.001	2,267,336
TABLE:dbt2.warehouse	5,822,560	4,205.982	0	11.684	0.001	3,911,060

- Table performance metrics are always powerful tools to identify IO bottleneck, lock contentions and SQL inefficiency.

Internal Analytics

- Metrics logged over time into Cassandra
- Capex Planning
- Proactive Performance Diagnosis