

How to configure WEBM System

Architecture of WEBM

WEBM system was designed as “Agent → Server” architecture. Now we have three agent-scripts for three different database platforms (MySQL, Oracle, MongoDB)

How to using three scripts to collect data from different database platforms

MySQL: use mysqlmon (developed by Fangxin Lou , C language)

Oracle: use oramon (a perl script)

MongoDB: use mongostat (a python script)

How to use MySQL as server database to store data

We have designed three tables to store these data

MySQL:

```
CREATE TABLE `mysql_load_statistics` (  
  `day` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  `host` varchar(64) NOT NULL,  
  `port` varchar(8) NOT NULL,  
  `load1m` float DEFAULT NULL,  
  `sys` int(11) DEFAULT NULL,  
  `wio` int(11) DEFAULT NULL,  
  `usr` int(11) DEFAULT NULL,  
  `free` int(11) DEFAULT NULL,  
  `swap` int(11) DEFAULT NULL,  
  `ibytes` int(11) DEFAULT NULL,  
  `obytes` int(11) DEFAULT NULL,  
  `errors` int(11) DEFAULT NULL,  
  `filenr` int(11) DEFAULT NULL,  
  `log` int(11) DEFAULT NULL,  
  `sess` int(11) DEFAULT NULL,  
  `act` int(11) DEFAULT NULL,
```

```
`exec` int(11) DEFAULT NULL,  
`cmmt` int(11) DEFAULT NULL,  
`ins` int(11) DEFAULT NULL,  
`upd` int(11) DEFAULT NULL,  
`del` int(11) DEFAULT NULL,  
`sel` int(11) DEFAULT NULL,  
`blog` int(11) DEFAULT NULL,  
`dump` int(11) DEFAULT NULL,  
`reco` int(11) DEFAULT NULL,  
`fbuf` int(11) DEFAULT NULL,  
`rowi` int(11) DEFAULT NULL,  
`rowu` int(11) DEFAULT NULL,  
`rowd` int(11) DEFAULT NULL,  
`rows` int(11) DEFAULT NULL,  
`disk` int(11) DEFAULT NULL,  
`writ` int(11) DEFAULT NULL,  
`redo` int(11) DEFAULT NULL,  
`enq` int(11) DEFAULT NULL,  
`wait` int(11) DEFAULT NULL,  
`d1readc` int(11) DEFAULT NULL,  
`d1rmerg` int(11) DEFAULT NULL,  
`d1rblks` int(11) DEFAULT NULL,  
`d1readt` int(11) DEFAULT NULL,  
`d1writc` int(11) DEFAULT NULL,  
`d1wmerg` int(11) DEFAULT NULL,  
`d1wblks` int(11) DEFAULT NULL,  
`d1writt` int(11) DEFAULT NULL,  
`d1ioact` int(11) DEFAULT NULL,  
`d1iotime` int(11) DEFAULT NULL,  
`d2readc` int(11) DEFAULT NULL,  
`d2rmerg` int(11) DEFAULT NULL,  
`d2rblks` int(11) DEFAULT NULL,  
`d2readt` int(11) DEFAULT NULL,  
`d2writc` int(11) DEFAULT NULL,  
`d2wmerg` int(11) DEFAULT NULL,  
`d2wblks` int(11) DEFAULT NULL,  
`d2writt` int(11) DEFAULT NULL,  
`d2ioact` int(11) DEFAULT NULL,  
`d2iotime` int(11) DEFAULT NULL,  
`d3readc` int(11) DEFAULT NULL,  
`d3rmerg` int(11) DEFAULT NULL,  
`d3rblks` int(11) DEFAULT NULL,  
`d3readt` int(11) DEFAULT NULL,  
`d3writc` int(11) DEFAULT NULL,
```

```
`d3wmerg` int(11) DEFAULT NULL,  
`d3wblks` int(11) DEFAULT NULL,  
`d3writt` int(11) DEFAULT NULL,  
`d3ioact` int(11) DEFAULT NULL,  
`d3iotime` int(11) DEFAULT NULL,  
PRIMARY KEY (`day`,`host`,`port`),  
KEY `uk_mysql_load_statistics` (`host`,`day`,`port`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8  
/*!50100 PARTITION BY RANGE ( UNIX_TIMESTAMP(day))  
(PARTITION p201401 VALUES LESS THAN (1391184000) ENGINE = InnoDB,  
PARTITION p201402 VALUES LESS THAN (1393603200) ENGINE = InnoDB,  
PARTITION p201403 VALUES LESS THAN (1396281600) ENGINE = InnoDB,  
PARTITION p201404 VALUES LESS THAN (1398873600) ENGINE = InnoDB,  
PARTITION p201405 VALUES LESS THAN (1401552000) ENGINE = InnoDB,  
PARTITION p201406 VALUES LESS THAN (1404144000) ENGINE = InnoDB,  
PARTITION p201407 VALUES LESS THAN (1406822400) ENGINE = InnoDB,  
PARTITION p201408 VALUES LESS THAN (1409500800) ENGINE = InnoDB,  
PARTITION p201409 VALUES LESS THAN (1412092800) ENGINE = InnoDB,  
PARTITION p201410 VALUES LESS THAN (1414771200) ENGINE = InnoDB,  
PARTITION p201411 VALUES LESS THAN (1417363200) ENGINE = InnoDB,  
PARTITION p201412 VALUES LESS THAN (1420041600) ENGINE = InnoDB,  
PARTITION p201501 VALUES LESS THAN (1422720000) ENGINE = InnoDB,  
PARTITION p201502 VALUES LESS THAN (1425139200) ENGINE = InnoDB,  
PARTITION p201503 VALUES LESS THAN (1427817600) ENGINE = InnoDB,  
PARTITION p201504 VALUES LESS THAN (1430409600) ENGINE = InnoDB,  
PARTITION p201505 VALUES LESS THAN (1433088000) ENGINE = InnoDB,  
PARTITION p201506 VALUES LESS THAN (1435680000) ENGINE = InnoDB,  
PARTITION p201507 VALUES LESS THAN (1438358400) ENGINE = InnoDB,  
PARTITION p201508 VALUES LESS THAN (1441036800) ENGINE = InnoDB,  
PARTITION p201509 VALUES LESS THAN (1443628800) ENGINE = InnoDB,  
PARTITION p201510 VALUES LESS THAN (1446307200) ENGINE = InnoDB,  
PARTITION p201511 VALUES LESS THAN (1448899200) ENGINE = InnoDB,  
PARTITION p201512 VALUES LESS THAN (1451577600) ENGINE = InnoDB,  
PARTITION p201601 VALUES LESS THAN (1454256000) ENGINE = InnoDB,  
PARTITION p201602 VALUES LESS THAN (1456761600) ENGINE = InnoDB,  
PARTITION p201603 VALUES LESS THAN (1459440000) ENGINE = InnoDB,  
PARTITION p201604 VALUES LESS THAN (1462032000) ENGINE = InnoDB,  
PARTITION p201605 VALUES LESS THAN (1464710400) ENGINE = InnoDB,  
PARTITION p201606 VALUES LESS THAN (1467302400) ENGINE = InnoDB,  
PARTITION p201607 VALUES LESS THAN (1469980800) ENGINE = InnoDB,  
PARTITION p201608 VALUES LESS THAN (1472659200) ENGINE = InnoDB,  
PARTITION p201609 VALUES LESS THAN (1475251200) ENGINE = InnoDB,  
PARTITION p201610 VALUES LESS THAN (1477929600) ENGINE = InnoDB,  
PARTITION p201611 VALUES LESS THAN (1480521600) ENGINE = InnoDB,
```

```
PARTITION p201612 VALUES LESS THAN (1483200000) ENGINE = InnoDB) */
```

Oracle:

```
CREATE TABLE `orastat` (
  `Time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `host` varchar(64) NOT NULL,
  `sid` varchar(20) NOT NULL,
  `Load` float DEFAULT NULL,
  `Logicr` int(11) DEFAULT NULL,
  `cget` int(11) DEFAULT NULL,
  `Phyr` int(11) DEFAULT NULL,
  `Phyw` int(11) DEFAULT NULL,
  `BlkCg` int(11) DEFAULT NULL,
  `Logcum` int(11) DEFAULT NULL,
  `Logcur` int(11) DEFAULT NULL,
  `CPU` int(11) DEFAULT NULL,
  `Redo` int(11) DEFAULT NULL,
  `Execs` int(11) DEFAULT NULL,
  `HParse` int(11) DEFAULT NULL,
  `Parse` int(11) DEFAULT NULL,
  `Comit` int(11) DEFAULT NULL,
  `Rollbk` int(11) DEFAULT NULL,
  `UsCall` int(11) DEFAULT NULL,
  `bget` int(11) DEFAULT NULL,
  `rwrt` int(11) DEFAULT NULL,
  `rwrs` int(11) DEFAULT NULL,
  `send` int(11) DEFAULT NULL,
  `users` int(11) DEFAULT NULL,
  `sys` int(11) DEFAULT NULL,
  `idle` int(11) DEFAULT NULL,
  `wait` int(11) DEFAULT NULL,
  `Netsed` int(11) DEFAULT NULL,
  `Netrec` int(11) DEFAULT NULL,
  PRIMARY KEY (`Time`, `host`, `sid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
/*!50100 PARTITION BY RANGE ( UNIX_TIMESTAMP(time))
(PARTITION p201401 VALUES LESS THAN (1391184000) ENGINE = InnoDB,
PARTITION p201402 VALUES LESS THAN (1393603200) ENGINE = InnoDB,
PARTITION p201403 VALUES LESS THAN (1396281600) ENGINE = InnoDB,
PARTITION p201404 VALUES LESS THAN (1398873600) ENGINE = InnoDB,
```

```
PARTITION p201405 VALUES LESS THAN (1401552000) ENGINE = InnoDB,  
PARTITION p201406 VALUES LESS THAN (1404144000) ENGINE = InnoDB,  
PARTITION p201407 VALUES LESS THAN (1406822400) ENGINE = InnoDB,  
PARTITION p201408 VALUES LESS THAN (1409500800) ENGINE = InnoDB,  
PARTITION p201409 VALUES LESS THAN (1412092800) ENGINE = InnoDB,  
PARTITION p201410 VALUES LESS THAN (1414771200) ENGINE = InnoDB,  
PARTITION p201411 VALUES LESS THAN (1417363200) ENGINE = InnoDB,  
PARTITION p201412 VALUES LESS THAN (1420041600) ENGINE = InnoDB,  
PARTITION p201501 VALUES LESS THAN (1422720000) ENGINE = InnoDB,  
PARTITION p201502 VALUES LESS THAN (1425139200) ENGINE = InnoDB,  
PARTITION p201503 VALUES LESS THAN (1427817600) ENGINE = InnoDB,  
PARTITION p201504 VALUES LESS THAN (1430409600) ENGINE = InnoDB,  
PARTITION p201505 VALUES LESS THAN (1433088000) ENGINE = InnoDB,  
PARTITION p201506 VALUES LESS THAN (1435680000) ENGINE = InnoDB,  
PARTITION p201507 VALUES LESS THAN (1438358400) ENGINE = InnoDB,  
PARTITION p201508 VALUES LESS THAN (1441036800) ENGINE = InnoDB,  
PARTITION p201509 VALUES LESS THAN (1443628800) ENGINE = InnoDB,  
PARTITION p201510 VALUES LESS THAN (1446307200) ENGINE = InnoDB,  
PARTITION p201511 VALUES LESS THAN (1448899200) ENGINE = InnoDB,  
PARTITION p201512 VALUES LESS THAN (1451577600) ENGINE = InnoDB,  
PARTITION p201601 VALUES LESS THAN (1454256000) ENGINE = InnoDB,  
PARTITION p201602 VALUES LESS THAN (1456761600) ENGINE = InnoDB,  
PARTITION p201603 VALUES LESS THAN (1459440000) ENGINE = InnoDB,  
PARTITION p201604 VALUES LESS THAN (1462032000) ENGINE = InnoDB,  
PARTITION p201605 VALUES LESS THAN (1464710400) ENGINE = InnoDB,  
PARTITION p201606 VALUES LESS THAN (1467302400) ENGINE = InnoDB,  
PARTITION p201607 VALUES LESS THAN (1469980800) ENGINE = InnoDB,  
PARTITION p201608 VALUES LESS THAN (1472659200) ENGINE = InnoDB,  
PARTITION p201609 VALUES LESS THAN (1475251200) ENGINE = InnoDB,  
PARTITION p201610 VALUES LESS THAN (1477929600) ENGINE = InnoDB,  
PARTITION p201611 VALUES LESS THAN (1480521600) ENGINE = InnoDB,  
PARTITION p201612 VALUES LESS THAN (1483200000) ENGINE = InnoDB) */
```

MongoDB:

```
db.serverstatus.insert({"connections":temp.conn,"locks":temp.locks,"indexCounters":temp.indexc  
ount,"recordStats":temp.record,"opcounter":temp.opcounter,"mem":temp.mem,"iowait":temp.io  
w,"load1m":temp.load1m,"load5m":temp.load5m,"swapout":temp.swapout,"free":temp.free,"us  
":temp.us,"sys":temp.sys,"idle":temp.idle,"swapin":temp.swapin,"netin":temp.netin,"net  
out":temp.netout,"sertime":a})
```

```
rs1:PRIMARY> db.serverstatus.findOne()
{
  "_id" : ObjectId("55070c13284bbc6bfd000000"),
  "load5m" : "0.03",
  "swapout" : 0,
  "us" : "0.29",
  "swapin" : 0,
  "mem" : {
    "resident" : 12200,
    "supported" : true,
    "virtual" : 39575,
    "mappedWithJournal" : 37204,
    "mapped" : 18602,
    "bits" : 64
  },
  "iowait" : "0.00",
  "opcounter" : {
    "getmore" : 9918972,
    "insert" : 2094217,
    "update" : 15093402,
    "command" : 246088918,
    "query" : 6142225,
    "delete" : 227
  },
  "netin" : 126517,
  "indexCounters" : {
    "missRatio" : 0,
    "resets" : 0,
    "hits" : 84777410,
    "misses" : 0,
    "accesses" : 84777495
  },
  "recordStats" : {
    "users" : {
      "pageFaultExceptionsThrown" : 0,
      "accessesNotInMemory" : 0
    },
    "admin" : {
      "pageFaultExceptionsThrown" : 0,
      "accessesNotInMemory" : 0
    },
    "fencing" : {
      "pageFaultExceptionsThrown" : 0,
```

```
    "accessesNotInMemory" : 1
  },
  "project211_kdb" : {
    "pageFaultExceptionsThrown" : 0,
    "accessesNotInMemory" : 0
  },
  "pageFaultExceptionsThrown" : 0,
  "local" : {
    "pageFaultExceptionsThrown" : 0,
    "accessesNotInMemory" : 1
  },
  "xiaoan_server" : {
    "pageFaultExceptionsThrown" : 0,
    "accessesNotInMemory" : 1
  },
  "test" : {
    "pageFaultExceptionsThrown" : 0,
    "accessesNotInMemory" : 0
  },
  "drugstore" : {
    "pageFaultExceptionsThrown" : 0,
    "accessesNotInMemory" : 0
  },
  "accessesNotInMemory" : 3
},
"connections" : {
  "current" : 257,
  "available" : 4743,
  "totalCreated" : 49911204
},
"locks" : {
  "users" : {
    "timeAcquiringMicros" : {
      "r" : 16133090,
      "w" : 0
    },
    "timeLockedMicros" : {
      "r" : 32124611,
      "w" : 0
    }
  }
},
"drugstore" : {
  "timeAcquiringMicros" : {
    "r" : 30623789,
```

```
        "w" : 23457292
      },
      "timeLockedMicros" : {
        "r" : NumberLong("23228401119"),
        "w" : NumberLong("2178329232")
      }
    },
    "admin" : {
      "timeAcquiringMicros" : {
        "r" : 205682615,
        "w" : 0
      },
      "timeLockedMicros" : {
        "r" : NumberLong("4494569997"),
        "w" : 0
      }
    },
    "fencing" : {
      "timeAcquiringMicros" : {
        "r" : 72952187,
        "w" : 470241378
      },
      "timeLockedMicros" : {
        "r" : 122807882,
        "w" : 34860110
      }
    },
    "project211_kdb" : {
      "timeAcquiringMicros" : {
        "r" : 311271844,
        "w" : 75778694
      },
      "timeLockedMicros" : {
        "r" : NumberLong("50012967772"),
        "w" : 907770392
      }
    },
    "ban" : {
      "timeAcquiringMicros" : {
        "r" : 162497,
        "w" : 0
      },
      "timeLockedMicros" : {
        "r" : 214824,
```



```
        "w" : 0
      }
    },
    "xiaoan_server" : {
      "timeAcquiringMicros" : {
        "r" : 107003258,
        "w" : 19127840
      },
      "timeLockedMicros" : {
        "r" : 281518700,
        "w" : 83518927
      }
    },
    "test" : {
      "timeAcquiringMicros" : {
        "r" : 52842491,
        "w" : 9
      },
      "timeLockedMicros" : {
        "r" : 107362705,
        "w" : 75
      }
    },
    "local" : {
      "timeAcquiringMicros" : {
        "r" : 593867688,
        "w" : 68321535
      },
      "timeLockedMicros" : {
        "r" : NumberLong("4225155636"),
        "w" : 1070888569
      }
    },
    "ba" : {
      "timeAcquiringMicros" : {
        "r" : 1993,
        "w" : 0
      },
      "timeLockedMicros" : {
        "r" : 2892,
        "w" : 0
      }
    }
  }
}
```

```
"idle" : 99,  
"sys" : "0.29",  
"free" : "105872464",  
"sertime" : ISODate("2015-03-17T01:00:01Z"),  
"netout" : 2161981,  
"load1m" : "0.01"  
}
```

Where to download these scripts

Github: <https://github.com/ylouis83/webm>

How to show graphs

We use amcharts (<http://www.amcharts.com>) to draw graphs on the web

You must write some PHP scripts to show these graph

eg:

```
#cat dml_bi.php  
<?php  
// Connect to MySQL  
$c = parse_ini_file ( "config.ini" , 1);  
$user= $c['mysql']['user'];  
$pass=$c['mysql']['password'];  
$ip=$c['mysql']['ip'];  
$port=$c['mysql']['port'];  
$link = mysql_connect( "$ip:$port", "$user", "$pass");  
if ( !$link ) {  
    die( 'Could not connect: ' . mysql_error() );  
}  
  
// Select the data base  
$db = mysql_select_db( 'dbmon', $link );  
if ( !$db ) {
```

```

die ( 'Error selecting database \'dbmon\' : ' . mysql_error() );
}
//数据库取sql
$oracle_comm="select `query_sql` from dbmon_web_config where name='my23_3307_TPS'
limit 1 ";
$oracle_comm_res=mysql_query($oracle_comm);
if(!$oracle_comm_res){
// Nope
$message = 'Invalid query: ' . mysql_error() . "\n";
$message .= 'Whole query: ' . $oracle_comm;
die( $message );
}
$oracle_comm_arr = mysql_fetch_array($oracle_comm_res,MYSQL_ASSOC);
$query=@$oracle_comm_arr['query_sql'];
if(empty($query)){
die("执行sql不存在");
}
// Fetch the data
//$query = "
//SELECT comit,execs ,time FROM ( SELECT comit,execs,time FROM
//`orastat` WHERE HOST='orayali3' AND sid = 'orayali3' AND time -1 < NOW() ORDER BY time
DESC LIMIT 1440 ) a ORDER BY time";
$result = mysql_query( $query );

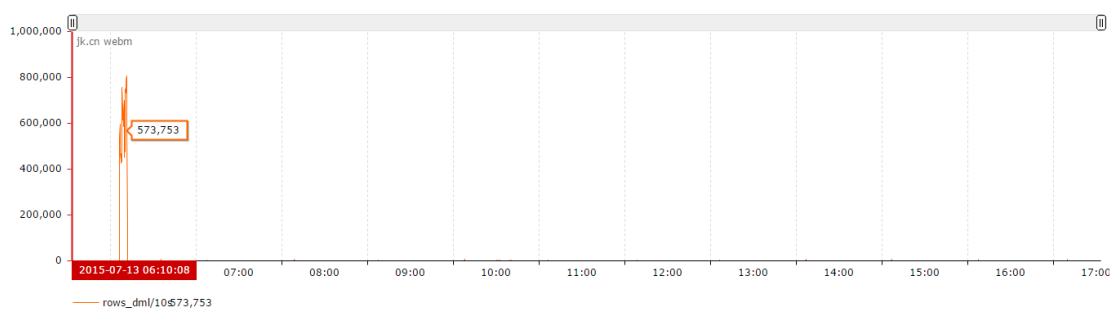
// All good?
if ( !$result ) {
// Nope
$message = 'Invalid query: ' . mysql_error() . "\n";
$message .= 'Whole query: ' . $query;
die( $message );
}
// Print out rows
// Print out rows
$prefix = "";
echo "\n";
while ( $row = mysql_fetch_assoc( $result ) ) {
echo $prefix . " {\n";
echo ' "category": "' . $row['DAY'] . "', ' . "\n";
echo ' "value1": ' . $row['TPS'] . " . "\n";
echo " }";
$prefix = ",\n";
}
echo "\n";

```

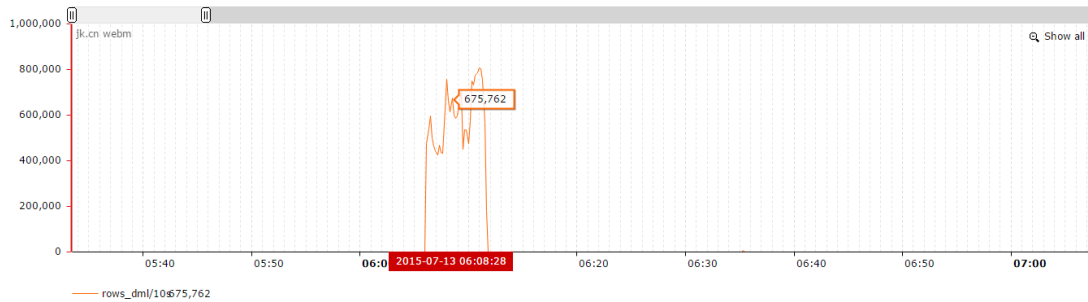
```
// Close the connection
mysql_close($link);
?>

#php dml_bi.php /more
[
{
"category": "2015-07-13 05:28:28",
"value1": 122
},
{
"category": "2015-07-13 05:28:38",
"value1": 45
},
{
"category": "2015-07-13 05:28:48",
"value1": 49
},
{
"category": "2015-07-13 05:28:58",
"value1": 2
},
{
"category": "2015-07-13 05:29:08",
```

We will get this graph



You can even pull-up this graph



You should write HTML files to show graphs from these PHP collected data. If you want to get more examples reference please visit amcharts website.

```
[17:35:20root@a1-dba-manager01b /usr/local/apache/htdocs/mywebsql/webm/mysql/user02]
```

```
#ls -l
```

```
total 84
```

```
-rw-r--r-- 1 root root 6761 Apr 30 10:35 active_sess.html
-rw-r--r-- 1 root root 7838 Apr 30 10:35 data_written.html
-rw-r--r-- 1 root root 7780 Apr 30 10:35 lock_time.html
-rw-r--r-- 1 root root 7744 Apr 30 10:35 logsize.html
-rw-r--r-- 1 root root 6744 Apr 30 10:35 rows_dml.html
-rw-r--r-- 1 root root 8282 Apr 30 10:35 rows_select.html
-rw-r--r-- 1 root root 7737 Apr 30 10:35 sess_stat.html
-rw-r--r-- 1 root root  192 Apr 28 13:48 style.css
-rw-r--r-- 1 root root 6760 Apr 30 10:35 total_rows_dml.html
-rw-r--r-- 1 root root 8310 Apr 30 10:35 total_rows_sel.html
```

How to show MongoDB performance state

```
#cat mongo_indexstat.php
```

```
<?php
```

```
$c = parse_ini_file ( "config.ini" , 1);
```

```
$user= $c['mongodb']['user'];
```

```
$pass=$c['mongodb']['password'];
```

```
$ip=$c['mongodb']['ip'];
```

```
$port=$c['mongodb']['port'];
```

```
$conn=new MongoClient( "mongodb://$user:$pass@$ip:$port" );
```

```
$db = $conn->local;
```

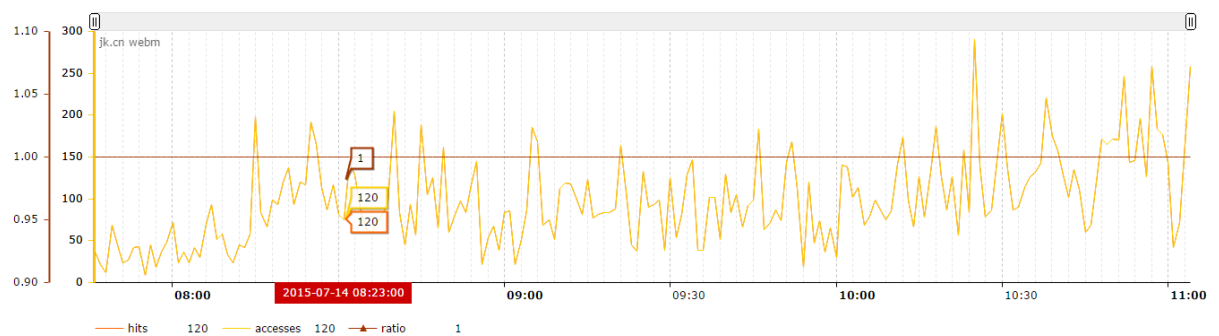
```
$collection = $db->serverstatus;
```

```

$cursor=$collection
->find()->fields(array("indexCounters"=>true,"sertime"=>true,"_id"=>false))->sort(array('sertime'
=>-1))->limit(200);
$prefix = "";
$tempsort=array();
$tempsort1=array();
foreach ($cursor as $doc){
    $array = array_values($doc);
    $category = date('Y-m-d H:i',$array[1]->sec);
    $tempsort[$category] = $array[0]['hits'];
    $tempsort1[$category] = $array[0]['accesses'];
}
ksort($tempsort, SORT_STRING);
ksort($tempsort1, SORT_STRING);
$tempsort=array_merge_recursive($tempsort,$tempsort1);
$arr = array_values($tempsort);
$arr1 = array_keys($tempsort);
echo "\n";
for ($x=0; $x<199; $x++) {
    echo $prefix . " {\n";
    echo '  "category": "' . $arr1[$x] . '", ' . "\n";
    echo '  "value1": ' . ($arr[$x+1][0]-$arr[$x][0]) . ', ' . "\n";
    echo '  "value2": ' . ($arr[$x+1][1]-$arr[$x][1]) . ', ' . "\n";
    echo '  "value3": ' . @round(($arr[$x+1][0]-$arr[$x][0])/($arr[$x+1][1]-$arr[$x][1]),2) . " .
"\n";
    echo " }";
    $prefix = ",\n";
}
echo "\n]";
?>

```

Using this PHP script you will get this graph



How to deploy these agents

MySQL:

```
#ps -ef |grep mysqlmon
```

```
root      1916  1832  0 14:02 pts/2    00:00:00 grep mysqlmon
root      41120                1    0    2014  ?                01:47:37
/usr/local/dbadmin/monitor/mysqlmon.bin parfile=/usr/local/dbadmin/conf/my3309.cfg
root      41206                1    0    2014  ?                01:48:15
/usr/local/dbadmin/monitor/mysqlmon.bin parfile=/usr/local/dbadmin/conf/my3310.cfg
```

```
#cat /usr/local/dbadmin/conf/my3309.cfg
```

```
USER=98473bfb251e10a2525c7cbf18a750f8bcd86335a208bb5bc1cfef9ddf027d7fdabf36519c6a
03f6a037eb1a5297eb6
LOAD=8eba347dad047be30ecf67858827ef3af6a38af83693022a6565c393d87da096000c26c369c
fac13
log=/tmp/mysql_21_3309.log
wait=10
```

Oracle:

Run oramon script to get this outputs:

```
$./oramon --orastat
-----
Time      Load   Logior  Phyr   Phyw  BlkCg  Logcum  Logcur  CPU   Redo   Execs  HParse  Parse  Comit  Rollbk  UsCall
07-14 14:54:01 3.0     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:03 3.0     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:05 3.0     1.9e+02 0      0      0      3      0     0s    0k     25     0      38     2      0      90
07-14 14:54:07 3.0     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:09 3.0     5      0      0      0      1      0     0s    0k     4      0      5      0      0      86
07-14 14:54:11 3.0     3      0      0      0      1      0     0s    0k     7      0      4      0      0      86
07-14 14:54:13 3.0     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:15 3.0     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:17 3.0     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:19 2.8     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
07-14 14:54:21 2.8     3      0      0      0      1      0     0s    0k     3      0      4      0      0      86
```

After running PHP scripts you will get this graph:



MongoDB:

```
#crontab -l
```

```
####mongodb_awr_report add by louis liu
```

```
*/1 * * * * /usr/local/dbadmin/monitor/mongostat1.py >>
```

```
/storage/sas/mongodb/log/mongostat.log
```

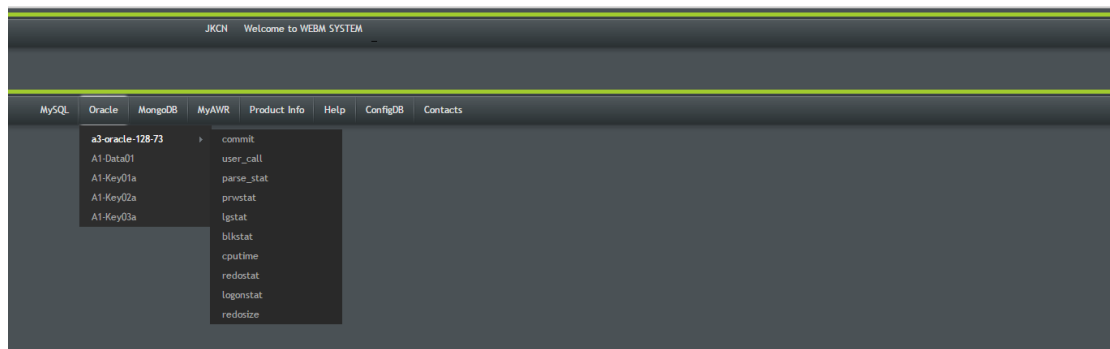
Run cronjob to collect data very minute.

```
> db.serverstatus.find().limit(1)
{ "_id" : ObjectId("55a38e02284bbc9966ae36bb"), "load5m" : "0.14", "swapout" : 0, "us" : "0.08", "swpin" : 0, "mem" : { "resident" : 12368, "supported" : true, "virtual" : 42656, "mappedWithJournal" : 37204, "mapped" : 28602, "bits" : 64 }, "iowait" : "0.00", "opcounter" : { "getmore" : 17610818, "insert" : 2189879, "update" : 16995397, "command" : 399549414, "query" : 31023862, "delete" : 437 }, "ratio" : 2210, "indexCounters" : [ { "missRatio" : 0, "resets" : 0, "hits" : 85084421, "misses" : 0, "accesses" : 95844507 }, { "admin" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 0 }, "fencing" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 1 }, "project211_kdb" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 0, "pageFaultExceptionsThrown" : 0, "local" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 1 }, "xiaoan_server" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 1 }, "test" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 0 }, "drugstore" : { "pageFaultExceptionsThrown" : 0, "accessesNotInMemory" : 0 }, "connections" : { "current" : 592, "available" : 4408, "totalCreated" : NumberLong(64973652) }, "locks" : { "users" : { "timeAcquiringMicros" : { "r" : NumberLong(16495407), "w" : NumberLong(0) }, "timeLockedMicros" : { "r" : NumberLong(32907102), "w" : NumberLong(0) }, "drugstore" : { "timeAcquiringMicros" : { "r" : NumberLong(178120642), "w" : NumberLong(32912479) }, "timeLockedMicros" : { "r" : NumberLong(9371766936), "w" : NumberLong(7812150563) } }, "admin" : { "timeAcquiringMicros" : { "r" : NumberLong(294792920), "w" : NumberLong(0) }, "timeLockedMicros" : { "r" : NumberLong("5296217443"), "w" : NumberLong(0) }, "fencing" : { "timeAcquiringMicros" : { "r" : NumberLong(105535162), "w" : NumberLong(470247159) }, "timeLockedMicros" : { "r" : NumberLong(188864282), "w" : NumberLong(35542894) }, "project211_kdb" : { "timeAcquiringMicros" : { "r" : NumberLong(471732256), "w" : NumberLong(75959539) }, "timeLockedMicros" : { "r" : NumberLong("163917455145"), "w" : NumberLong(910038857) }, "system" : { "timeAcquiringMicros" : { "r" : NumberLong(160), "w" : NumberLong(0) }, "timeLockedMicros" : { "r" : NumberLong(789), "w" : NumberLong(0) }, "ban" : { "timeAcquiringMicros" : { "r" : NumberLong(162497), "w" : NumberLong(0) }, "timeLockedMicros" : { "r" : NumberLong(214824), "w" : NumberLong(0) }, "xiaoan_server" : { "timeAcquiringMicros" : { "r" : NumberLong(166995770), "w" : NumberLong(15006335) }, "timeLockedMicros" : { "r" : NumberLong(440421035), "w" : NumberLong(89791995) } }, "user" : { "timeAcquiringMicros" : { "r" : NumberLong(83301), "w" : NumberLong(0) }, "timeLockedMicros" : { "r" : NumberLong(212237), "w" : NumberLong(0) }, "test" : { "timeAcquiringMicros" : { "r" : NumberLong(7769552), "w" : NumberLong(9) }, "timeLockedMicros" : { "r" : NumberLong(161158876), "w" : NumberLong(75) }, "local" : { "timeAcquiringMicros" : { "r" : NumberLong(997395950), "w" : NumberLong(112774479) }, "timeLockedMicros" : { "r" : NumberLong("6848242451"), "w" : NumberLong(1531289071) }, "ba" : { "timeAcquiringMicros" : { "r" : NumberLong(1993), "w" : NumberLong(0) }, "timeLockedMicros" : { "r" : NumberLong(2892), "w" : NumberLong(0) }, "idle" : 100, "sys" : "0.04", "free" : "103922768", "sertime" : ISODate("2015-07-13T18:08:01Z"), "netout" : 2918, "loadm" : "0.14" }
```

ALL of these data were stored in another MongoDB database.

How to configure WEBM navigation page

We have a beautiful navigation page which called webm-guide page. All of these pages were made up of JS code. I download a sample JS code to do this work.



It is cool☺ You can configure monitored items by yourself.

WEBM_GUIDE supports MySQL Oracle MongoDB platforms so far. You can add redis memcached Hbase items if you need

All of scripts have already been uploaded to github:

<https://github.com/ylouis83/webm>

Totally, WEBM is a smart, light tool. It's easy to configure. You can design graphs whatever you want and add monitor items if you actually need.