All collections are now stored in mapReduce db.

> use mapReduce

3. MapReduce to find the number of times each IP accessed a website

> var map = function() {emit(this.ipAddress, {count:1});};

> var reduce = function(key, values) {

var total = 0;

values.forEach(function(value) {

total += value.count;

});

return {count:total}; };

> db.accessLog.mapReduce(map, reduce, {out:"ipAddress\_totals"});

{

"result" : "ipAddress\_totals",

"timeMillis" : 549,

"counts" : {

"input" : 35111,

"emit" : 35111,

"reduce" : 1164,

"output" : 1945

},

"ok" : 1

}

> db.ipAddress\_totals.find();

{ "\_id" : "1.162.207.87", "value" : { "count" : 4 } }

{ "\_id" : "1.170.44.84", "value" : { "count" : 83 } }

{ "\_id" : "1.192.146.100", "value" : { "count" : 1 } }

{ "\_id" : "1.202.184.142", "value" : { "count" : 1 } }

{ "\_id" : "1.202.184.145", "value" : { "count" : 1 } }

{ "\_id" : "1.202.89.134", "value" : { "count" : 2 } }

{ "\_id" : "1.234.2.41", "value" : { "count" : 12 } }

{ "\_id" : "1.56.79.5", "value" : { "count" : 4 } }

{ "\_id" : "1.59.91.151", "value" : { "count" : 4 } }

{ "\_id" : "1.62.189.221", "value" : { "count" : 4 } }

{ "\_id" : "1.85.17.247", "value" : { "count" : 1 } }

{ "\_id" : "10.15.10.129", "value" : { "count" : 2812 } }

{ "\_id" : "10.15.10.135", "value" : { "count" : 2108 } }

{ "\_id" : "10.15.10.144", "value" : { "count" : 2 } }

{ "\_id" : "10.15.10.151", "value" : { "count" : 4 } }

{ "\_id" : "10.15.11.112", "value" : { "count" : 2 } }

{ "\_id" : "10.15.8.173", "value" : { "count" : 3 } }

{ "\_id" : "10.15.8.20", "value" : { "count" : 5 } }

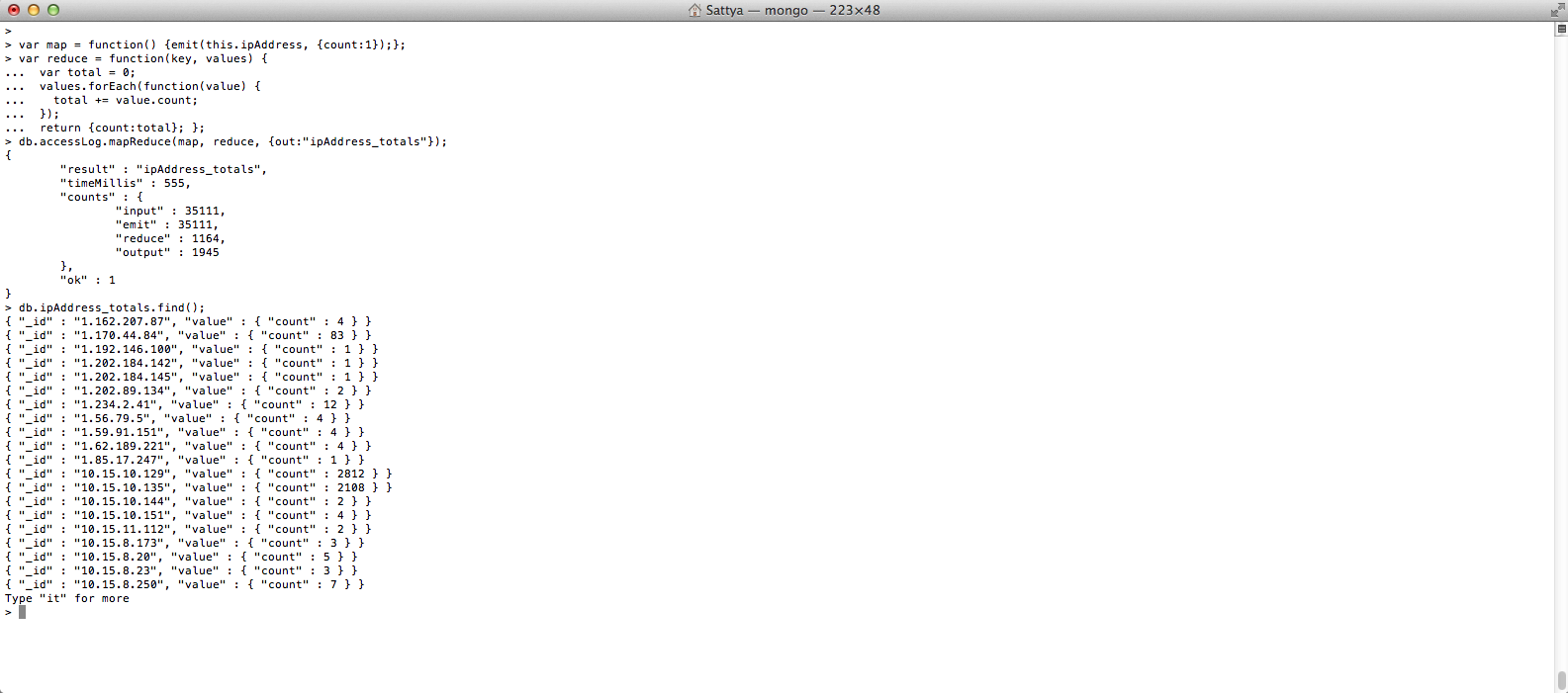
{ "\_id" : "10.15.8.23", "value" : { "count" : 3 } }

{ "\_id" : "10.15.8.250", "value" : { "count" : 7 } }

Type "it" for more

>

Screenshot:



4. MapReduce to find the number of times each HTTP status code appeared in logfile.

> var map = function() {emit(this.statusCode, {count:1});};

> var reduce = function(key, values) {

var total = 0;

values.forEach(function(value) {

total += value.count;

});

return {count:total}; };

> db.accessLog.mapReduce(map, reduce, {out:"statusCode\_totals"});

{

"result" : "statusCode\_totals",

"timeMillis" : 391,

"counts" : {

"input" : 35111,

"emit" : 35111,

"reduce" : 861,

"output" : 11

},

"ok" : 1

}

> db.statusCode\_totals.find();

{ "\_id" : "200", "value" : { "count" : 5002 } }

{ "\_id" : "206", "value" : { "count" : 407 } }

{ "\_id" : "301", "value" : { "count" : 249 } }

{ "\_id" : "302", "value" : { "count" : 6 } }

{ "\_id" : "304", "value" : { "count" : 9816 } }

{ "\_id" : "400", "value" : { "count" : 122 } }

{ "\_id" : "403", "value" : { "count" : 72 } }

{ "\_id" : "404", "value" : { "count" : 19390 } }

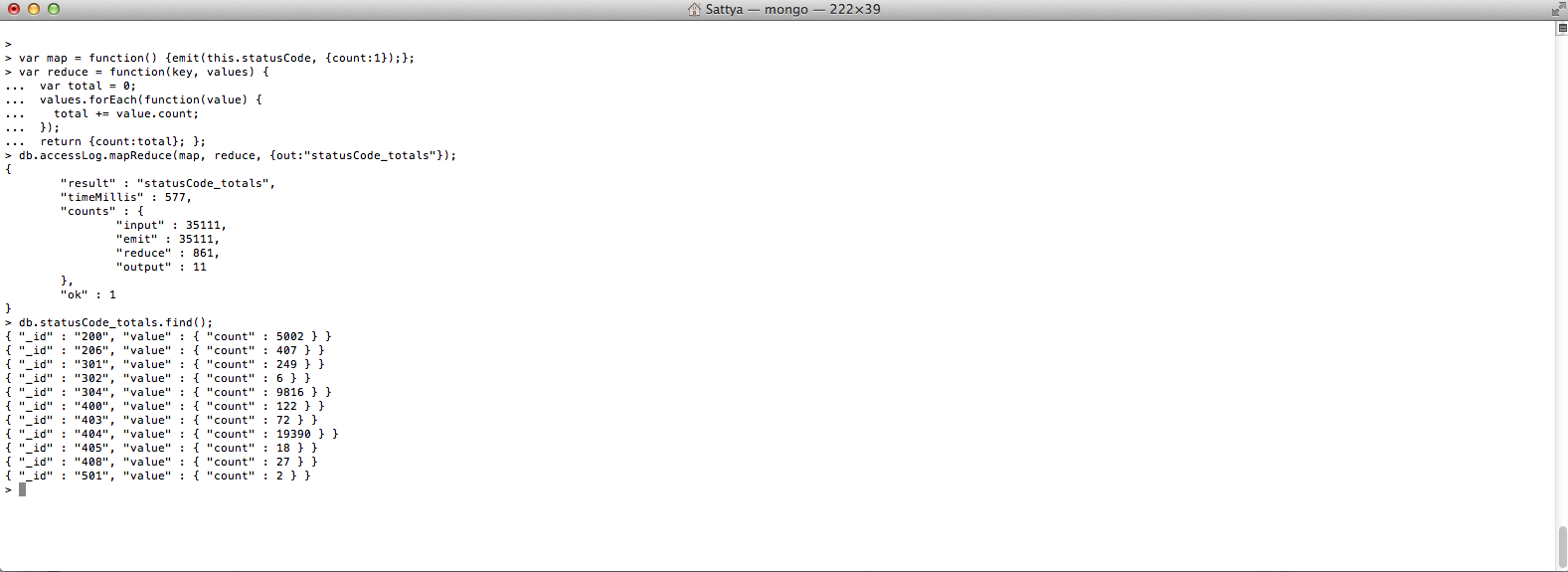
{ "\_id" : "405", "value" : { "count" : 18 } }

{ "\_id" : "408", "value" : { "count" : 27 } }

{ "\_id" : "501", "value" : { "count" : 2 } }

>

Screenshot:



5.

> var map = function() {emit(this.messageType, {count:1});};

> var reduce = function(key, values) { var total = 0; values.forEach(function(value) { total += value.count; }); return {count:total}; };

> db.errorLog.mapReduce(map, reduce, {out:"messageType\_totals"});

{

"result" : "messageType\_totals",

"timeMillis" : 282,

"counts" : {

"input" : 19980,

"emit" : 19980,

"reduce" : 235,

"output" : 4

},

"ok" : 1

}

> db.messageType\_totals.find();

{ "\_id" : "--", "value" : { "count" : 10 } }

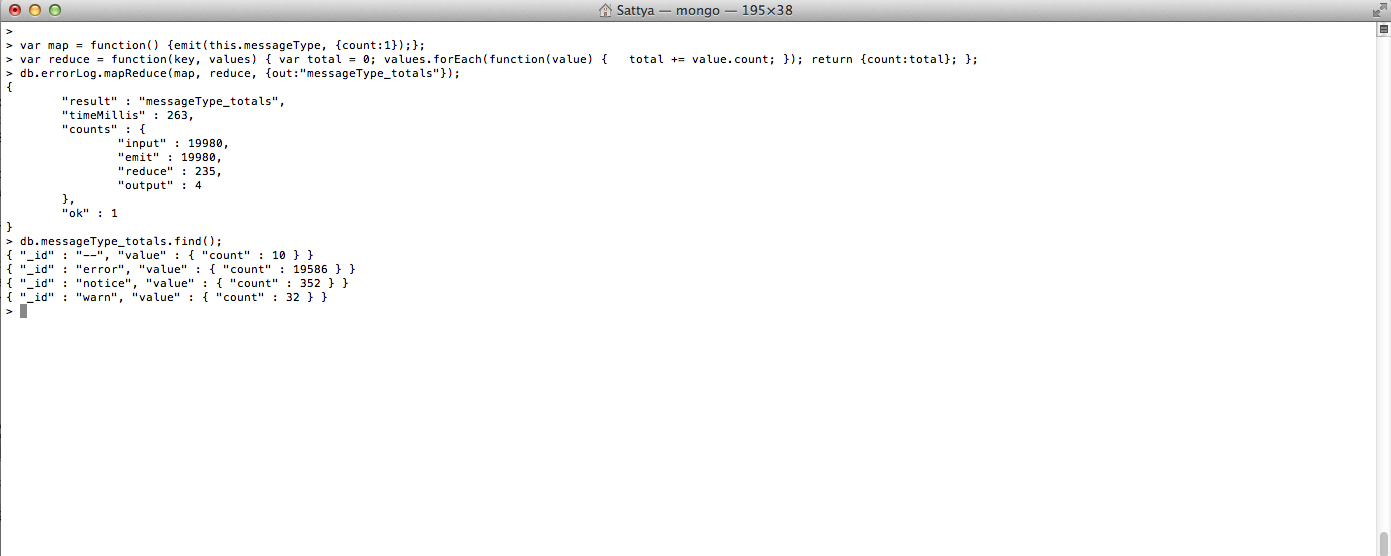
{ "\_id" : "error", "value" : { "count" : 19586 } }

{ "\_id" : "notice", "value" : { "count" : 352 } }

{ "\_id" : "warn", "value" : { "count" : 32 } }

>

Screenshot:



6. Import csv file using

$ mongoimport -d mapReduce -c nyseStocks --type csv --file ~/Downloads/NYSE\_daily\_prices\_A.csv –headerline

> var map = function() { if(this.stock\_symbol.match(new RegExp(/^A.\*/))) { emit(this.stock\_symbol, {stockHighPrice:this.stock\_price\_high, count:1}); }};

> var reduce = function(key, values) { var total = {stockHighPrice:0, count:0}; values.forEach(function(value) { total.stockHighPrice += value.stockHighPrice; total.count += value.count; }); return total; };

> var finalize = function(key, total) { total.avg = total.stockHighPrice / total.count; return total.avg; };

> db.nyseStocks.mapReduce(map, reduce, {out:"stock\_avg", finalize:finalize});

{

"result" : "stock\_avg",

"timeMillis" : 12378,

"counts" : {

"input" : 735026,

"emit" : 735026,

"reduce" : 7551,

"output" : 203

},

"ok" : 1

}

> db.stock\_avg.find();

{ "\_id" : "AA", "value" : 52.459682054670246 }

{ "\_id" : "AAI", "value" : 10.518446478515234 }

{ "\_id" : "AAN", "value" : 19.847593646277623 }

{ "\_id" : "AAP", "value" : 44.721311953352796 }

{ "\_id" : "AAR", "value" : 19.208936170212834 }

{ "\_id" : "AAV", "value" : 12.498480836236944 }

{ "\_id" : "AB", "value" : 30.64627297543216 }

{ "\_id" : "ABA", "value" : 25.99447019867549 }

{ "\_id" : "ABB", "value" : 12.583610986042329 }

{ "\_id" : "ABC", "value" : 47.7895740691136 }

{ "\_id" : "ABD", "value" : 15.721916592724055 }

{ "\_id" : "ABG", "value" : 15.429047379032308 }

{ "\_id" : "ABK", "value" : 51.31720845792452 }

{ "\_id" : "ABM", "value" : 24.528461061122712 }

{ "\_id" : "ABR", "value" : 18.430806896551726 }

{ "\_id" : "ABT", "value" : 48.188009450679914 }

{ "\_id" : "ABV", "value" : 31.986431181486065 }

{ "\_id" : "ABVT", "value" : 49.196723684210596 }

{ "\_id" : "ABX", "value" : 22.683009677931274 }

{ "\_id" : "ACC", "value" : 25.51832005792907 }

Type "it" for more

>

