opEvents Events Aggregate

One of the big benefits of opEvents is the big data approach to event management, this allows opEvents to perform sophisticated aggregations which combine data from multiple sources into a new data set which the application can use as needed. Millions of records can be queried, aggregated, new fields created in milliseconds.

MongoDB aggregations are not for the faint hearted, if you have had some experience with SQL Group by queries with joins, you will have no problem, and you don't need to use all the features to achieve the results you need.

You can learn more about MongoDB Aggregations Here

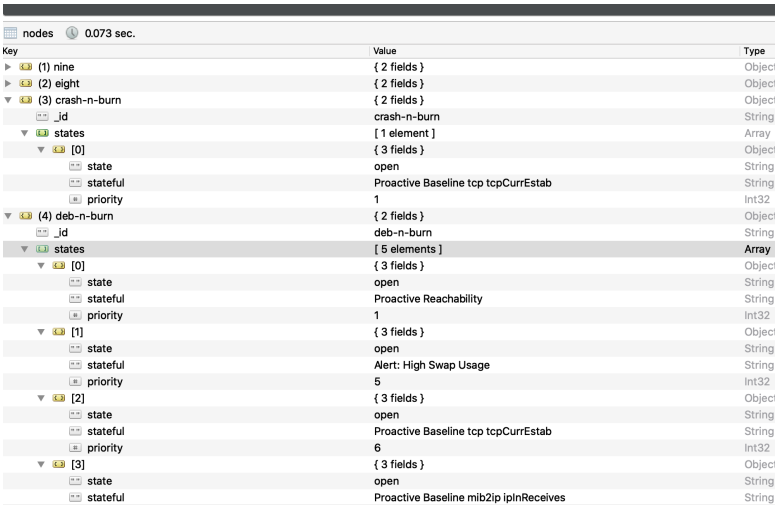
To help accessing MongoDB many of the Opmantek developers use Robo3T from RoboMongo, an Open Source GUI for MongoDB

Example opEvents Aggregations

Aggregation query for events to summarise the nodes by current state.

```
db.getCollection('nodes').aggregate([
  //get all nodes from the group HQDEV
  {
    $match: { group: "HQDev" }
  },
  //Get all the states for the matched nodes
  {
    $lookup: {
      from: "state",
      localField: "_id",
      foreignField: "node",
      as: "states"
    }
  },
  //unwinds the states array creating a document per state with node config data
  {
    $unwind: {
      path: "$states"
    }
  },
  //We only want open states
  {
    $match: { 'states.state': { $eq: 'open' } }
  },
  //Join with the event which created this state, we will need this later for the priority
  {
    $lookup: {
      from: "events",
      localField: "states.eventid_down",
      foreignField: "_id",
      as: "_event"
    }
  },
  {
     $unwind: {
        path: "$_event"
      }
    },
  // group by the node, and accumlate its states
  {
    $group: {
      _id: "$_id",
      states: { $push: { state: "$states.state", stateful: "$states.stateful", priority: "$_event.priority" } }
    }
  },

])
```



nodes 0.073 sec.

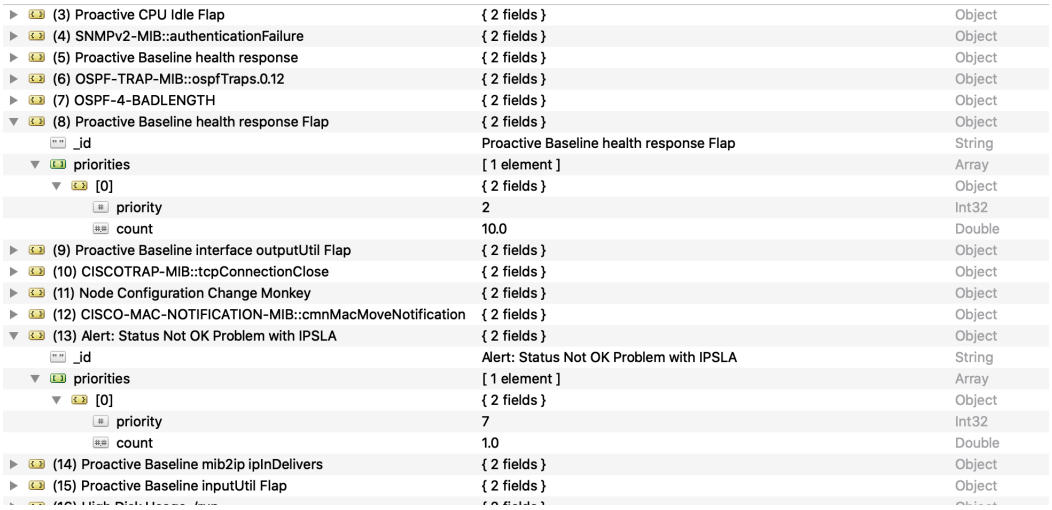
Key	Value	Type
(1) nine	{ 2 fields }	Object
(2) eight	{ 2 fields }	Object
(3) crash-n-burn	{ 2 fields }	Object
_id	crash-n-burn	String
states	[1 element]	Array
[0]	{ 3 fields }	Object
state	open	String
stateful	Proactive Baseline tcp tcpCurrEstab	String
priority	1	Int32
(4) deb-n-burn	{ 2 fields }	Object
_id	deb-n-burn	String
states	[5 elements]	Array
[0]	{ 3 fields }	Object
state	open	String
stateful	Proactive Reachability	String
priority	1	Int32
[1]	{ 3 fields }	Object
state	open	String
stateful	Alert: High Swap Usage	String
priority	5	Int32
[2]	{ 3 fields }	Object
state	open	String
stateful	Proactive Baseline tcp tcpCurrEstab	String
priority	6	Int32
[3]	{ 3 fields }	Object
state	open	String
stateful	Proactive Baseline mib2ip ipInReceives	String

Aggregation query for events to summarise the events by the priority.

Takes all events after a certain time and which are not acknowledged

Firstly it groups by the event name and priority, and counts the amount of events

```
db.getCollection('events').aggregate([
  {
    $match: { "time": { "$gte": 1577836800 }, "acknowledged": { "$eq": 0 } }
  },
  {
    $group: {
      _id: { event: "$event", priority: "$priority" },
      priorities: { $push: "$priority" },
      total: { $sum: 1 }
    }
  },
  {
    $group: {
      _id: "$_id.event",
      priorities: { $push: { priority: "$_id.priority", count: "$total" } },
    }
  }
])
```

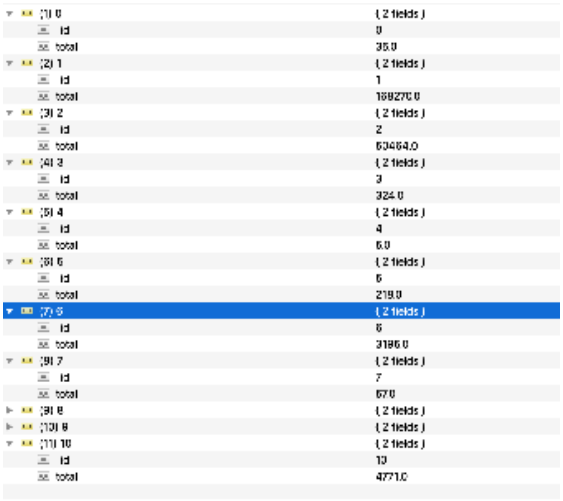


(3) Proactive CPU Idle Flap	{ 2 fields }	Object
(4) SNMPv2-MIB::authenticationFailure	{ 2 fields }	Object
(5) Proactive Baseline health response	{ 2 fields }	Object
(6) OSPF-TRAP-MIB::ospfTraps.0.12	{ 2 fields }	Object
(7) OSPF-4-BADLENGTH	{ 2 fields }	Object
(8) Proactive Baseline health response Flap	{ 2 fields }	Object
_id	Proactive Baseline health response Flap	String
priorities	[1 element]	Array
[0]	{ 2 fields }	Object
priority	2	Int32
count	10.0	Double
(9) Proactive Baseline interface outputUtil Flap	{ 2 fields }	Object
(10) CISCOTRAP-MIB::tcpConnectionClose	{ 2 fields }	Object
(11) Node Configuration Change Monkey	{ 2 fields }	Object
(12) CISCO-MAC-NOTIFICATION-MIB::cmnMacMoveNotification	{ 2 fields }	Object
(13) Alert: Status Not OK Problem with IPSLA	{ 2 fields }	Object
_id	Alert: Status Not OK Problem with IPSLA	String
priorities	[1 element]	Array
[0]	{ 2 fields }	Object
priority	7	Int32
count	1.0	Double
(14) Proactive Baseline mib2ip ipInDelivers	{ 2 fields }	Object
(15) Proactive Baseline inputUtil Flap	{ 2 fields }	Object

Priority sorted for all events which are not acknowledged

```
db.getCollection('events').aggregate([
  {
    $match: { "time": { "$gte": 1577836800 }, "acknowledged": { "$eq": 0 } }
  },
  {
    $group: {
      _id: "$priority",
      total: { $sum: 1 }
    }
  },
  {$sort: {"_id": 1}}

])
```


{1} 0
{ 2 fields }
id
0
total
35.0
{2} 1
{ 2 fields }
id
1
total
189270.0
{3} 2
{ 2 fields }
id
2
total
60464.0
{4} 3
{ 2 fields }
id
3
total
324.0
{5} 4
{ 2 fields }
id
4
total
6.0
{6} 5
{ 2 fields }
id
5
total
219.0
{7} 6
{ 2 fields }
id
6
total
3186.0
{8} 7
{ 2 fields }
id
7
total
670
{9} 8
{ 2 fields }
{10} 9
{ 2 fields }
{11} 10
{ 2 fields }
id
10
total
4771.0