# 1 目标

清除MongoDB指定Collection中N天前的数据。

# 2 可行性分析

我们知道，MongoDB的shell可执行JS脚本完成mongo客户端命令，所以我们可以使用JS脚本来配置指定Collection并完成remove操作；而使用JS的函数，我们可以轻易的获取指定日期格式的N天前的日期。

使用Linux的crontab计划任务，定时执行Shell脚本，在脚本中使用mongo执行JS脚本即可完成目标。注意，下列操作都是在MongoDB复制集环境下的primary节点上进行！所有在主节点上的操作均可同步到从节点上。

# 3 编写Shell脚本

Shell脚本内容很简单，举例如下：

/root/mongodb-linux-x86\_64-3.0.6/bin/mongo 192.168.8.26:27017/dev -u devDBAdmin -p E4yunMon90 -quiet /root/staleDataCleaner.js

解释如下（注意空格）：

<mongo绝对路径> <ip>:<port>/<dbname> -u <username> -p <password> -quiet <JS脚本绝对路径>

其中dbname表示要清理的Collection所在数据库，username和password是我们指定数据库的dbOwner用户或者更高权限的用户。一般情况下，我们的数据库均为dev，用户和密码则是devDBAdmin用户及其密码。

编写完脚本后，保存，并将脚本的权限修改为755：

[root@localhost ~]# chmod 755 mongoDBStaleDataCleaner.sh

# 4 添加crontab定时任务

[root@localhost ~]# crontab –e

写入

0 0 \* \* \* /bin/sh /root/mongoDBStaleDataCleaner.sh >> /root/stale\_data\_cleaning.log

即每天00：00执行脚本，日志append到/root/stale\_data\_cleaning.log中。

# 5 附JS脚本

Date.prototype.format = function(format)

{

var o =

{

"M+" : this.getMonth()+1, //month

"d+" : this.getDate(), //day

"h+" : this.getHours(), //hour

"m+" : this.getMinutes(), //minute

"s+" : this.getSeconds(), //second

"q+" : Math.floor((this.getMonth()+3)/3), //quarter

"S" : this.getMilliseconds() //millisecond

}

if(/(y+)/.test(format))

format=format.replace(RegExp.$1,(this.getFullYear()+"").substr(4 - RegExp.$1.length));

for(var k in o)

if(new RegExp("("+ k +")").test(format))

format = format.replace(RegExp.$1,RegExp.$1.length==1 ? o[k] : ("00"+ o[k]).substr((""+ o[k]).length));

return format;

}

function getDay(day){

var today = new Date();

var targetday\_milliseconds=today.getTime() + 1000\*60\*60\*24\*day;

today.setTime(targetday\_milliseconds); //core code

var tYear = today.getFullYear();

var tMonth = today.getMonth();

var tDate = today.getDate();

tMonth = doHandleMonth(tMonth + 1);

tDate = doHandleMonth(tDate);

return tYear+"-"+tMonth+"-"+tDate+" 00:00:00";

}

function doHandleMonth(month){

var m = month;

if(month.toString().length == 1){

m = "0" + month;

}

return m;

}

function doDeleteStaleData (arr, colName, preDay) {

if(colName=='time'){

for (var i = arr.length - 1; i >= 0; i--) {

var count=db[arr[i]].find({'time':{'$lt':preDay}}).count();

print("Stale data in [\""+arr[i]+"\"] counts " + count);

db[arr[i]].remove({'time':{'$lt':preDay}});

}

}else if(colName=='date'){

for (var i = arr.length - 1; i >= 0; i--) {

var count=db[arr[i]].find({'date':{'$lt':preDay}}).count();

print("Stale data in [\""+arr[i]+"\"] counts " + count);

db[arr[i]].remove({'date':{'$lt':preDay}});

}

}else if(colName=='timestamp'){

for (var i = arr.length - 1; i >= 0; i--) {

var count=db[arr[i]].find({'timestamp':{'$lt':preDay}}).count();

print("Stale data in [\""+arr[i]+"\"] counts " + count);

db[arr[i]].remove({'timestamp':{'$lt':preDay}});

}

}

}

var myDate = new Date();

var datetime = myDate.format("yyyy-MM-dd 00:00:00");

var daysAgo = -90;

var preDay = getDay(daysAgo);

print("\n--------------------BEGIN--------------------\n");

print("Today is "+datetime+"\nBegin deleting stale data before "+preDay+" ...\n");

datetime= new Date(datetime);

preDay = new Date(preDay);

var col\_time = ["ecmc.monitor.alarm.item","monitor.alarm.item"];

var col\_date = ["bandwidth.network.incoming.summary","bandwidth.network.outgoing.summary"];

var col\_timestamp = [

"cpu\_util.10min",

"cpu\_util.1d",

"cpu\_util.1h",

"cpu\_util.2h",

"cpu\_util.30min",

"cpu\_util.3min",

"cpu\_util.5min",

"cpu\_util.detail",

"disk.read.bytes.rate.10min",

"disk.read.bytes.rate.12h",

"disk.read.bytes.rate.1d",

"disk.read.bytes.rate.1h",

"disk.read.bytes.rate.2h",

"disk.read.bytes.rate.30min",

"disk.read.bytes.rate.3min",

"disk.read.bytes.rate.5min",

"disk.read.bytes.rate.detail",

"disk.write.bytes.rate.10min",

"disk.write.bytes.rate.12h",

"disk.write.bytes.rate.1d",

"disk.write.bytes.rate.1h",

"disk.write.bytes.rate.2h",

"disk.write.bytes.rate.30min",

"disk.write.bytes.rate.3min",

"disk.write.bytes.rate.5min",

"disk.write.bytes.rate.detail",

"memory.usage.10min",

"memory.usage.12h",

"memory.usage.1h",

"memory.usage.2h",

"memory.usage.30min",

"memory.usage.3min",

"memory.usage.5min",

"memory.usage.detail",

"network.incoming.bytes.rate.10min",

"network.incoming.bytes.rate.12h",

"network.incoming.bytes.rate.1d",

"network.incoming.bytes.rate.1h",

"network.incoming.bytes.rate.2h",

"network.incoming.bytes.rate.30min",

"network.incoming.bytes.rate.3min",

"network.incoming.bytes.rate.5min",

"network.incoming.bytes.rate.detail",

"network.outgoing.bytes.rate.10min",

"network.outgoing.bytes.rate.12h",

"network.outgoing.bytes.rate.1d",

"network.outgoing.bytes.rate.1h",

"network.outgoing.bytes.rate.2h",

"network.outgoing.bytes.rate.30min",

"network.outgoing.bytes.rate.3min",

"network.outgoing.bytes.rate.5min",

"network.outgoing.bytes.rate.detail",

"obs.storageUsed.1h",

"obs.storageUsed.1month",

"obs.storageUsed.24h",

"obs.used.1h",

"obs.used.1month",

"obs.used.24h",

"bandwidth.network.incoming.detail",

"bandwidth.network.outgoing.detail"];

doDeleteStaleData(col\_time, 'time', preDay);

doDeleteStaleData(col\_date, 'date', preDay);

doDeleteStaleData(col\_timestamp, 'timestamp', preDay);

print("\n----------DONE deleting stale data!----------\n");