

# 日志分析入门

## 版本修改记录

版本号	修订时间	修订人	修改内容
V1.0	2021年6月	曾倩云	创建文档

## 目录

日志分析入门.....	1
一、 日志类型.....	2
二、 bi.log 常用关键字.....	11
三、 常用工具.....	20
四、 日志分析学习方法.....	20
五、 日志搜集.....	21
六、 常见场景/逻辑.....	21

# 一、日志类型

id	日志类型	目录	记录信息	获取方式
1	bi.log	Yonghong/log Yonghong/logpack	产品运行日志	1. 从底层目录下获取 2. 从 <a href="#">管理系统-日志管理</a> 下载
2	catalina.out	tomcat/logs	tomcat启动日志	1. 从底层目录下获取
3	jvm.log	Yonghong/log	jvm信息	1. 从底层目录下获取 2. 从 <a href="#">管理系统-日志管理</a> 下载
4	jstack		线程日志	1. 手动打(jstack pid) 2. 通过脚本自动搜集
5	jmap		内存日志	1. 手动打(jmap -histo) 2. 通过脚本自动搜集
6	localhost_access_log	tomcat/logs	tomcat请求日志	1. 从底层目录下获取
7	hs_err_pid.log	tomcat/logs	JVM出现错误时dump下来的	1. 从底层目录下获取

备注：  
1. jstack和jmap默认没有，不会自动生成，没有对应目录。如果是脚本自动搜集，则在脚本定义的目录

## 1. bi.log

### 1.1 文件大小和保存时间

- ◆ 每个日志文件的物理最大值 默认 50M。可通过 `file.log.limit=50` 修改
- ◆ 日志保存时间 默认 30 天 可通过 `file.log.days=30` 修改
- ◆ 日志自动删除，删除记录记载在 `catalina.out` 中，tomcat 启动完成后会自动检测超过 30 天的日志，进行自动删除。如果系统一直启动，会每天的这个时候自动检测删除。删除的日志是 `Yonghong/log` 和 `Yonghong/logpack` 两个目录下的。

```
Log file 'bi.log 2021.05.06 09.27.43' out of date, has been deleted.
Log file 'bi.log 2021.05.07 15.48.50' out of date, has been deleted.
Log file 'bi.log 2021.05.07 15.57.53' out of date, has been deleted.
Log file 'bi.log 2021.05.07 17.46.33' out of date, has been deleted.
Log file 'bi.log 2021.05.07 20.16.18' out of date, has been deleted.
Log file 'bi.log 2021.05.08 19.00.07' out of date, has been deleted.
Log file pack 'bi.log 2021.05.06 09.27.43.zip' out of date, has been deleted.
Log file pack 'bi.log 2021.05.07 15.48.50.zip' out of date, has been deleted.
Log file pack 'bi.log 2021.05.07 15.57.53.zip' out of date, has been deleted.
Log file pack 'bi.log 2021.05.07 17.46.33.zip' out of date, has been deleted.
Log file pack 'bi.log 2021.05.07 20.16.18.zip' out of date, has been deleted.
```

- ◆ 产品运行日志是往 Yonghong/log/bi.log 写入，当 bi.log 文件超过 50M，进行自动分割，生成带时间戳的日志文件 `bi.log yyyy.MM.dd HH.mm.ss` 和 `bi.log yyyy.MM.dd HH.mm.ss.zip`(Yonghong/logpack) 带的时间戳是这个文件生成的时间，也就是这个日志结束的时间。除了大小超过 50M 自动分割，在 管理系统->日志管理 点击【新建日志文件】也会进行分割。

```
root@LAPTOP-CCVF58KB: /opt/yh9031/Yonghong/Log# tail -5 'bi.log 2021.06.11 19.50.25'
[2021-06-11 19:50:25.271] |[INFO] |[system-cfa9cf9223044abc0dfcc707a215f95] |[system] |[g5.AbsSetting.getBoolean(AbsSetting.java:359)] |[boolean value doe
s not exist: db.memory.debug]
[2021-06-11 19:50:25.271] |[INFO] |[system-cfa9cf9223044abc0dfcc707a215f95] |[system] |[g5.mem.MemoryManagerReporter.run0(MemManager.java:429)] |[Memory inf
ormation: max jvm mem: 2048M, used jvm mem: 967M, free jvm mem: 1080M
--GLink Report, link count: 10, total link mem: 301M, used link mem: 1M
--GLink Report, link count: 6, total link mem: 682M, used link mem: 0M
--Direct memory report, max: 2048M, used: 0M]
root@LAPTOP-CCVF58KB: /opt/yh9031/Yonghong/Log# ls
bi.log      'bi.log 2021.05.18 11.54.03'  'bi.log 2021.05.27 22.39.15'  'bi.log 2021.06.01 14.31.53'  jvm.log.00  jvm.log.08
'bi.log 2021.05.17 15.53.42'  'bi.log 2021.05.18 14.16.44'  'bi.log 2021.05.28 14.03.18'  'bi.log 2021.06.01 18.48.40'  jvm.log.01  jvm.log.09
'bi.log 2021.05.18 10.19.39'  'bi.log 2021.05.19 09.36.05'  'bi.log 2021.05.28 14.05.29'  'bi.log 2021.06.10 13.10.32'  jvm.log.02  jvm.log.10
'bi.log 2021.05.18 10.30.06'  'bi.log 2021.05.21 21.50.27'  'bi.log 2021.05.28 14.26.05'  'bi.log 2021.06.10 17.21.57'  jvm.log.03  jvm.log.11
'bi.log 2021.05.18 10.57.02'  'bi.log 2021.05.24 16.40.26'  'bi.log 2021.05.31 13.43.04'  'bi.log 2021.06.10 18.28.55'  jvm.log.04  jvm.log.12
'bi.log 2021.05.18 11.05.44'  'bi.log 2021.05.24 16.44.06'  'bi.log 2021.05.31 10.55.52'  'bi.log 2021.06.10 19.27.28'  jvm.log.05  jvm.log.13
'bi.log 2021.05.18 11.25.17'  'bi.log 2021.05.24 17.10.19'  'bi.log 2021.06.01 10.09.24'  'bi.log 2021.06.11 19.50.25'  jvm.log.06  jvm.log.14
'bi.log 2021.05.18 11.43.04'  'bi.log 2021.05.27 11.30.13'  'bi.log 2021.06.01 10.29.33'  jvm.log      jvm.log.07  jvm.log.15
```



## 1.2 下载逻辑 (管理系统->日志管理)

- A. 下载日志可以选择某一个时间段进行 (起始时间 到 起始时间+时间跨度)，实际下载的日志时间并不是刚好是这个时间段，而是包含这个时间的所有 bi.log xxx.zip 和 bi.log 文件，没有截取的功能。
- B. Yonghong/logpack 下日志不要手动删除，如果下载某段时间的日志，这个日志在 Yonghong/log 下存在，但是在 Yonghong/logpack 下不存在，也是没有办法通过 管理

系统->日志管理 进行下载的, 会导致下载的日志 跟选择的时间对不上。

- C. 当选择的时间段的日志不存在, 下载下来的日志是这个时间后面存在的最近的一个 bi.log  
xxxx。
- D. 如果是一个集群, 可以在 C 节点下载所有节点(单 C 节点除外)的 bi.log。Eg: 集群  
(192.168.0.101 c 节点, 192.168.0.102 cr 节点, 192.168.0.103 nm 节点), 那么在  
102 节点只能下载节点 102 和 103 节点的 bi.log, 不能下载 101 节点的 bi.log;反之, 在  
101 节点可以下载 101,102,103 这 3 个节点的 bi.log

日志下载

起始时间: 2019-01-24 13:36:41

时间跨度: 30分钟

节点地址:

- 192.168.1.146+50 (CMR)
- 192.168.1.146+51 (CNMR)

- E. 管理系统 -> 日志管理 下载的完整的日志可能包含: bi.log ,bi.log yyyy.MM.dd  
HH.mm.ss.zip, jvm.log.xx, thread.log (下载的时候勾选【下载线程日志】才会生成,  
包含的线程日志是下载时刻的线程日志, 不能下载历史某个时间的, 相当于手动执行命令  
在下载时候打线程)

log-10.	98.zip	180,280,218	WinRAR ZIP 压缩文件	2021/6/1 10:27	一级目录, 文件名:log-日志下载节点.zip	
log-9	06.zip	78,675,944	78,568,926	WinRAR ZIP 压缩文件	2021/6/1 10:19	6E8F833E
log-9	09.zip	11,661,167	11,572,588	WinRAR ZIP 压缩文件	2021/6/1 10:19	30F34F35
log-9	7.zip	17,327,429	17,060,997	WinRAR ZIP 压缩文件	2021/6/1 10:19	556A06C3
log-9	23.zip	16,588,205	16,329,890	WinRAR ZIP 压缩文件	2021/6/1 10:19	7E4E9965
log-1	18.zip	28,665,057	27,943,862	WinRAR ZIP 压缩文件	2021/6/1 10:19	1FEE98A6
log-1	32.zip	15,803,594	15,744,510	WinRAR ZIP 压缩文件	2021/6/1 10:19	2E2D0E47
log-1	95.zip	13,234,768	13,158,487	WinRAR ZIP 压缩文件	2021/6/1 10:19	AF2D61E8
bi.log	2021.06.01 09.05.36.zip	4,476,632	3,974,242	WinRAR ZIP 压缩文件	2021/6/1 9:05	17701DAD
bi.log	2021.06.01 09.11.11.zip	4,654,749	4,158,420	WinRAR ZIP 压缩文件	2021/6/1 9:11	A70FC33D
bi.log	2021.06.01 09.15.56.zip	4,938,960	4,446,423	WinRAR ZIP 压缩文件	2021/6/1 9:15	890D8A6C
bi.log	2021.06.01 09.21.22.zip	5,192,238	4,224,383	WinRAR ZIP 压缩文件	2021/6/1 9:21	73955E44
bi.log	2021.06.01 09.24.22.zip	5,569,516	4,630,075	WinRAR ZIP 压缩文件	2021/6/1 9:24	7E92F3D0
bi.log	2021.06.01 09.30.29.zip	4,724,977	4,675,932	WinRAR ZIP 压缩文件	2021/6/1 9:30	1871F2D3
bi.log	2021.06.01 09.36.33.zip	4,939,181	4,844,811	WinRAR ZIP 压缩文件	2021/6/1 9:36	C3944272
bi.log	2021.06.01 09.43.15.zip	4,482,333	4,400,919	WinRAR ZIP 压缩文件	2021/6/1 9:43	3538B072
bi.log	2021.06.01 09.48.26.zip	4,534,138	4,468,500	WinRAR ZIP 压缩文件	2021/6/1 9:48	4780D0P9
bi.log	2021.06.01 09.52.29.zip	4,826,730	4,444,814	WinRAR ZIP 压缩文件	2021/6/1 9:52	50649D0E
bi.log	2021.06.01 09.56.50.zip	3,959,221	3,856,574	WinRAR ZIP 压缩文件	2021/6/1 9:56	3D3F0A5C
bi.log	2021.06.01 09.59.37.zip	5,087,796	4,580,901	WinRAR ZIP 压缩文件	2021/6/1 9:59	0D762918
bi.log	2021.06.01 10.04.52.zip	5,052,719	4,985,817	WinRAR ZIP 压缩文件	2021/6/1 10:04	8D593235
bi.log	2021.06.01 10.08.38.zip	4,946,838	4,900,200	WinRAR ZIP 压缩文件	2021/6/1 10:08	3272AF66
bi.log	2021.06.01 10.11.40.zip	4,737,913	4,652,481	WinRAR ZIP 压缩文件	2021/6/1 10:11	C8D867E3
bi.log	2021.06.01 10.19.01.zip	4,620,520	4,553,403	WinRAR ZIP 压缩文件	2021/6/1 10:19	FDC38C62
bi.log		11,070,797	998,261	文本文件	2021/6/1 10:19	81D38A4C
jvm.log		7,766,944	875,924	文本文件	2021/6/1 10:19	2642CF54
jvm.log.00		25,398	3,352	00 文件	2021/4/15 23:24	FCAD0FCE
jvm.log.01		35,359	4,827	01 文件	2021/4/15 23:09	04A1E5D0

二级目录, 文件名:  
log-勾选的节点.zip  
//每个节点一个压缩文件

三级目录: 单独的bi.log  
jvm.log.xx  
以及 每个bi.log xxxx.zip

## 1.3 相关设置

### ◆ 日志显示内容

默认的级别有 ERROR、WARN、INFO、DEBUG、TRACE，日志等级依次降低，用于筛选出符合对应日志级别的日志记录。

日志显示内容:  ERROR  WARN  INFO  DEBUG  TRACE 行数:   全部显示

### ◆ 日志管理配置，设置日志记录的等级，从上往下，日志等级依次降低，记录的更加详细。

跟上面一点的差别是：日志显示内容是查看的时候筛选，日志管理配置是日志文件中记录哪个级别。



### ◆ 系统参数配置

【调试信息参数配置】主要用于出现问题的时候，打开对应配置，输出更加详细的日志信息，用于定位问题，如果打开 bi.log 日志会记录的更加详细，生成更多日志，如果系统正常运行，这些配置可以不开启，默认不开启。这些配置的开启或关闭，不需要重启，直接生效。

如果是由于特定问题需要添加的参数，添加在 bi.properties 中，重启生效。也能会通过 product.jar 添加一些参数，用于输出更详细的信息。



## 2. catalina.out

### 2.1 文件大小和保存时间

- ◆ catalina.out 日志不会自动分割，也不会自动删除，会随着时间的推移越来越大，所以需要手动处理，rename 原来的，然后新建。
- ◆ catalina.yyyy-MM-dd.log 有进行启动的日期，每天生成一个，如果没有，表示这天没有进行启动。只记录启动相关的日志，其他的输出到 catalina.out 中的日志，不会记录到每天的 catalina.yyyy-MM-dd.log 中，所以拿 catalina.out 通常是拿 catalina.out 文件最后 10 万行，而不是拿对应日期的 catalina.yyyy-MM-dd.log。

### 3. jvm.log

搜索关键字“Real”，可以看到jvm内存回收的情况，一般关注时间大于1s的，然后看看回收的具体情况，短时间内频繁进行FULL GC，对系统是有影响的，感知到的一般是系统卡顿，慢，无响应（FULL GC的时候系统是没有响应的），有规律的长时间FULL GC，就需要看看关注一下了，特别是GC时间长，但是内存又并没有回收回来的情况。进一步分析还需结合jmap和jstack日志。

注意：脚本自动搜集jmap日志，需要注意，搜集的脚本需要是jmap -histo pid，而不能是jmap -histo:live pid（live需要去掉，有可能会引发规律的GC，导致性能问题）

```
270098 [2021-05-31T07:58:53.761+0800] GC(168599) Metaspace: 141554K->141545K(147456K)
270099 [2021-05-31T07:58:53.761+0800] GC(168599) Pause [Full] (GI Evacuation Pause) 63932M->63104M(64192M) 7514.172ms
270100 [2021-05-31T07:58:53.761+0800] GC(168599) User=73.24s Sys=0.05s Real=7.51s
270101 [2021-05-31T07:58:53.762+0800] GC(168600) Concurrent Mark From Roots 7514.630ms
270102 [2021-05-31T07:58:53.762+0800] GC(168600) Concurrent Mark Abort
270103 [2021-05-31T07:58:53.762+0800] GC(168600) Concurrent Cycle 7515.008ms
270104 [2021-05-31T07:58:55.369+0800] GC(168601) Pause Young (Normal) (GI Evacuation Pause)
270105 [2021-05-31T07:58:55.369+0800] GC(168601) Using 10 workers of 10 for evacuation
270106 [2021-05-31T07:58:55.431+0800] GC(168601) To-space exhausted
270107 [2021-05-31T07:58:55.431+0800] GC(168601) Pre Evacuate Collection Set: 0.1ms
270108 [2021-05-31T07:58:55.431+0800] GC(168601) Evacuate Collection Set: 11.4ms
270109 [2021-05-31T07:58:55.431+0800] GC(168601) Post Evacuate Collection Set: 49.9ms
270110 [2021-05-31T07:58:55.431+0800] GC(168601) Other: 1.0ms
270111 [2021-05-31T07:58:55.431+0800] GC(168601) Eden regions: 26->0(100)

Final result = (2182 hits)
Line 270025: [2021-05-31T07:58:35.484+0800] GC(168596) User=70.29s Sys=0.00s Real=7.44s
Line 270042: [2021-05-31T07:58:37.067+0800] GC(168596) User=0.43s Sys=0.01s Real=0.06s
Line 270060: [2021-05-31T07:58:44.656+0800] GC(168597) User=73.86s Sys=0.07s Real=7.55s
Line 270074: [2021-05-31T07:58:46.247+0800] GC(168598) User=0.44s Sys=0.01s Real=0.06s
Line 270100: [2021-05-31T07:58:53.761+0800] GC(168599) User=73.24s Sys=0.05s Real=7.51s
Line 270117: [2021-05-31T07:58:55.432+0800] GC(168601) User=0.44s Sys=0.00s Real=0.06s
Line 270135: [2021-05-31T07:58:58.871+0800] GC(168600) User=70.36s Sys=0.05s Real=7.44s
Line 270149: [2021-05-31T07:59:04.647+0800] GC(168603) User=0.52s Sys=0.02s Real=0.07s
Line 270175: [2021-05-31T07:59:12.041+0800] GC(168604) User=71.55s Sys=0.14s Real=7.39s
Line 270192: [2021-05-31T07:59:13.709+0800] GC(168606) User=0.44s Sys=0.01s Real=0.06s
Line 270210: [2021-05-31T07:59:12.109+0800] GC(168607) User=71.59s Sys=0.15s Real=7.40s
Line 270224: [2021-05-31T07:59:22.716+0800] GC(168608) User=0.47s Sys=0.00s Real=0.06s
Line 270250: [2021-05-31T07:59:27.596+0800] GC(168609) User=47.53s Sys=0.04s Real=4.88s
Line 270266: [2021-05-31T07:59:37.993+0800] GC(168611) User=0.49s Sys=0.01s Real=0.06s
Line 270278: [2021-05-31T08:00:16.899+0800] GC(168613) User=0.54s Sys=0.07s Real=0.10s
Line 270295: [2021-05-31T08:00:18.655+0800] GC(168613) User=0.35s Sys=0.01s Real=0.04s
Line 270300: [2021-05-31T08:00:20.008+0800] GC(168613) User=0.00s Sys=0.00s Real=0.00s
```

### 4. jstack&jmap

#### 4.1 搜集

jstack&jmap不会自动记录，但是分析问题的时候，尤其是性能问题又特别重要。搜集方式如下：

- A. 出现问题的时候手动搜集，步骤如下：
  - a. ps -ef|grep tomcat 找到 tomcat 对应的进程号(pid)



```
root@SC-201907131229: /opt/yh904/tomcat/bin# ps -ef|grep tomcat
root pid 85 1 0 May26 ? 02:28:57 /usr/lib/jvm/jdk-11.0.7/bin/java -Djava.util.logging
java.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDHKeySize=2048
es -Xmx8059m -Xms8059m -XX:MetaspaceSize=256m -XX:G1HeapRegionSize=32m -XX:InitiatingHeapOccupancy
:ParallelGCThreads=10 -XX:+ParallelRefProcEnabled -XX:SoftRefLRUPolicyMSPerMB=0 -Xlog:gc*:file=/o
java.awt.headless=true --add-exports java.base/jdk.internal.ref=ALL-UNNAMED --add-opens java.base
1.0.7/bin/jstack %p >/opt/yh9.2/tomcat/logs/thread_stack_%p.log -XX:ErrorFile=/opt/yh9.2/tomcat/l
.0.7/bin/jmap -dump:format=b,file=/opt/yh9.2/tomcat/logs/jmap_%p.log %p -XX:+ExitOnOutOfMemoryErr
lasspath :/opt/yh9.2/tomcat/webapps/bi/WEB-INF/classes:/opt/yh9.2/tomcat/./Yonghong/product/api-
0210305.jar:/opt/yh9.2/tomcat/./Yonghong/product/thirds.jar:/opt/yh9.2/tomcat/./Yonghong/lib/mo
.jar:/opt/yh9.2/tomcat/bin/bootstrap.jar:/opt/yh9.2/tomcat/bin/tomcat-juli.jar -Dcatalina.base=/o
tmpdir=/opt/yh9.2/tomcat/temp org.apache.catalina.startup.Bootstrap start
```

b. jstack pid >>jstack.txt //手动每 2-3 秒执行一次，执行多次

jmap -histo pid >> jmap.txt //手动执行多次

```
root@SC-201907131229:/yonghong# jstack 85 >>jstack.txt
root@SC-201907131229:/yonghong# jstack 85 >>jstack.txt
root@SC-201907131229:/yonghong# jstack 85 >>jstack.txt
root@SC-201907131229:/yonghong# jmap -histo 85 >>jmap.txt
root@SC-201907131229:/yonghong# jmap -histo 85 >>jmap.txt
root@SC-201907131229:/yonghong# jmap -histo 85 >>jmap.txt
root@SC-201907131229:/yonghong# ls
jmap.txt jstack.txt
```

生成的jstack&jmap 文件在当前目录下， 或者可以指定生成目录。在哪个目录下执行以上两个命令，没有要求。注意：需要出现问题的时候手动搜集。

### B. 下载 bi.log 的时候勾选【下载线程日志】

有局限性，很少这样用，这种方式只能在下载日志的时候生成 jstack 日志，出现问题的时候对应的日志拿不到。

### C. 部署监控自动搜集

优点：部署后可自动搜集 jstack 和 jmap,可设置搜集频率，日志保留时间等。



jstack&jmap日志  
收集脚本-linux-to

## 4.2 关键字

卡顿类的问题，首先检查线程，搜索“deadlock”，如果有说明是有问题的。



```

"Jmap_log" "bi_log" "jstack_log"
"Gang worker#3 (Parallel CMS Threads)" os_prio=0 tid=0x00007f94040fd000 nid=0x6a94 runnable
"VM Periodic Task Thread" os_prio=0 tid=0x00007f94041ba800 nid=0x6aa9 waiting on condition
JNI global references: 922

Found one Java-level deadlock ← 这意味着出现死锁
-----
http-bio-8080-exec-3052 : ←
waiting for ownable synchronizer 0x00000004aa9e62b8, (a java.util.concurrent.locks.ReentrantLock$NonfairSync),
which is held by "http-bio-8080-exec-3043" ← 互锁的两个线程
http-bio-8080-exec-3043 : ←
waiting for ownable synchronizer 0x00000004aafcb0f0, (a java.util.concurrent.locks.ReentrantLock$NonfairSync),
which is held by "http-bio-8080-exec-3052"

Java stack information for the threads listed above:
-----

```

## 5. localhost\_access\_log

### 5.1 日志含义

在 tomcat/conf/server.xml 中添加参数 %D ,localhost\_access\_log 日志中会记录上**请求的时延大小**, 单位是 ms。新版本中已经自动加上了这个配置, 老版本如果没有, 可以手动添加。

```

164 <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
165       prefix="localhost_access_log" suffix=".txt"
166       pattern="%h %l %u %t &quot;%r&quot; %s %b %D" />
167 <Valve className="org.apache.catalina.valves.ErrorReportValve"
168       showReport="false" showServerInfo="false" />
169 </Host>

```

```

127.0.0.1 - - [16/Jun/2021:10:12:14 +0800] "GET /bi/api?action=getRebootConfig HTTP/1.1" 200 41 1118
127.0.0.1 - - [16/Jun/2021:10:12:14 +0800] "GET /bi/api?action=testAccess HTTP/1.1" 200 133 31
192.168.0.251 - - [16/Jun/2021:10:14:34 +0800] "GET /bi/Viewer HTTP/1.1" 200 4577 96
192.168.0.251 - - [16/Jun/2021:10:14:34 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/uti1_d1c3a8a37942c98757a.js HTTP/1.1" 200 1292 7
192.168.0.251 - - [16/Jun/2021:10:14:34 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/index_9a351d2aa8e5d4f1c5e.css HTTP/1.1" 200 103894 43
192.168.0.251 - - [16/Jun/2021:10:14:34 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/index_aa876971f9bc17a44b16.js HTTP/1.1" 200 573041 229
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/AuditMtr_962bfe5f4bba9eb191e.js HTTP/1.1" 200 7541 7
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/common_d7f10c4bbcfac6e6b092.css HTTP/1.1" 200 4062 6
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/Login_1ce4f9e92ba6ca7f955c.css HTTP/1.1" 200 1716 7
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/vendor_be8173689254ad4b435b8.js HTTP/1.1" 200 67726 17
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/DialLogContainer_dce2d1f574b094e82f68.css HTTP/1.1" 200 1410 6
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/Login_b8aad50f13a29e098.js HTTP/1.1" 200 6402 7
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/DialLogContainer_f6bb3e39e0b20b984d1.js HTTP/1.1" 200 1102 6
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/common_b9e70721d180295bd5f5.js HTTP/1.1" 200 268740 56
192.168.0.251 - - [16/Jun/2021:10:14:35 +0800] "GET /bi/Viewer?proc=2&action=loadFavicon HTTP/1.1" 200 46326 6
192.168.0.251 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer HTTP/1.1" 200 1162 2
192.168.0.251 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer HTTP/1.1" 200 1985 10
192.168.0.224 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/pc/LoginPage_9d3cd204f7f0b1d1081bc.css HTTP/1.1" 200 3233 11
192.168.0.224 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/pc/uti1_c31189259a41572c432.js HTTP/1.1" 200 32589 38
192.168.0.224 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/pc/LoginPage_a72089444a75edd5d956.js HTTP/1.1" 200 32366 38
192.168.0.224 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/pc/common_37622b4d43265d122e9.css HTTP/1.1" 200 97942 86
192.168.0.224 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/pc/vendor_d4b251d5d66edaf466f.js HTTP/1.1" 200 29235 107
192.168.0.224 - - [16/Jun/2021:10:14:46 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/pc/common_d637b871c96a1e8470cb.js HTTP/1.1" 200 25599 129
192.168.0.224 - - [16/Jun/2021:10:14:47 +0800] "GET /bi/Viewer?proc=2&action=loadImage&resource=/gs/h5/style/images/Login/Logo.png HTTP/1.1" 200 406596 9
192.168.0.224 - - [16/Jun/2021:10:14:47 +0800] "GET /bi/Viewer?proc=2&action=loadImage&resource=/gs/h5/style/images/Login/Logo.png HTTP/1.1" 200 11584 7
192.168.0.224 - - [16/Jun/2021:10:14:47 +0800] "GET /bi/Viewer?proc=2&action=loadFavicon HTTP/1.1" 200 3162 4
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/Common.worker_a4f1d7ab52bcca7d20d7.worker.js HTTP/1.1" 200 14779 8
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "POST /bi/Viewer?isAjax=true&proc=1&isAjax=true&platform=WAP&deviceType=31&wap_act=login&admin=admin&pass=Bella123 HTTP/1.1" 200 518 58
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&action=loadFavicon HTTP/1.1" 200 1329 4
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/Home_6b8df872998589f6e3.js HTTP/1.1" 200 3931 6
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "POST /bi/Viewer?isAjax=true&proc=1&isAjax=true&platform=WAP&deviceType=31&action=reqNeedLogin=false HTTP/1.1" 200 531 46
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/Home_714352c3963b0d8e866.css HTTP/1.1" 200 951 53
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&action=loadViewerImage&resource=/gs/h5/style/images/wap/Logo.png HTTP/1.1" 200 3558 3
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "POST /bi/Viewer?isAjax=true&proc=1&isAjax=true&platform=WAP&deviceType=31&action=req HTTP/1.1" 200 3558 3
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "POST /bi/Viewer?isAjax=true&proc=1&isAjax=true&platform=WAP&deviceType=31&action=req HTTP/1.1" 200 3891 73
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/FullScreenHelper_82621fa728c98bb575.js HTTP/1.1" 200 1352 3
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/FullScreenHelper_5f8a67306f6d8892c68e.js HTTP/1.1" 200 1316 6
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/MobilePreviewToolBarUI_9ea9f6e0c19092609d.css HTTP/1.1" 200 829 5
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/MobilePreviewToolBarUI_9998f15ef8f383136a.js HTTP/1.1" 200 3819 5
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/ExtraElemPane_073f1513dcd412e156.js HTTP/1.1" 200 381 3
192.168.0.251 - - [16/Jun/2021:10:15:04 +0800] "POST /bi/Viewer?isAjax=true&proc=1&isAjax=true&platform=WAP&deviceType=31&action=req HTTP/1.1" 200 3851 60
192.168.0.251 - - [16/Jun/2021:10:15:05 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/ChartModule-ElementFullScreenModule-ListModule-MListUI_032d58326f29
00d0453.js HTTP/1.1" 200 4025 9
192.168.0.251 - - [16/Jun/2021:10:15:05 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/css/wap/ElemModule_252853e933afede9e16c.css HTTP/1.1" 200 4148 8
192.168.0.251 - - [16/Jun/2021:10:15:05 +0800] "GET /bi/Viewer?proc=2&locale=zh_CN&resource=/gs/h5/dist/js/wap/ElemModule_27f21d09c42ef112dc.js HTTP/1.1" 200 2748 11

```

格式	含义
%h	远端访问的客户端 IP
%l	远端请求用来认证的用户名 (一直是 '-')
%u	远端访问的已认证的用户 (如果有的话), 没有的话显示 '-'
%t	日期和时间
%r	请求的第一行 (方法和请求地址)
%s	返回结果对应的 http code
%b	发送的字节数, 不包含 httpHeader, 如果是 0 的话显示为 -
%D	处理这个请求的时间, 毫秒

## 5.2 分析场景

app 连接以及系统慢, 白屏等问题需要分析该日志。

## 6. hs\_err\_pid.log

主要是 OOM 的时候可能会生成。

## 7. 操作系统日志

宕机的时候 (进程不在) 可在服务器上执行这两个命令, 确认是否是系统 kill 了进程。

```
grep "Out of memory" /var/log/messages
```

```
egrep -i -r 'killed process' /var/log
```

## 二、bi.log 常用关键字

### 1. 直连相关

#### 1.1 start to init

标志查询开始执行，记录 **执行时间**，**查询数据库**，**查询 ID**，**对应参数**。

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log# cat bi.log |grep 'start to init'
[2021-06-20 23:42:10.193] |[INFO] |[779c2e1c06c14021aa96a3ca80801bd7] |[admin] |[g5.qry.jdbc.JDBCExecutor.init(JDBCExecutor.java:109)] |[DAS
HBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][下拉列表过滤3] g5.qry.sql.impl.MySQLExecutor@3b0777d9[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] start to init with params: {_MAX_ROWS=500000, fromVooltdbJoinNode=false, _DISCARDED_GRID=false, _SEL_STRATEGY=0, _Running_QQue
ry_Already_Added=true, _REFRESH=true, _MAX_PERMIT_ROWS=5000000, _isMobile=false, _CONN_TIME_OFFSET=0, __DATA_CACHE_KEY__=QRY[bella.coffeetes
t.JDBCConn: 4, jdbc:mysql://127.0.0.1:3306, com.mysql.jdbc.Driver,30,true,true,root,false,_mi002d0048002400480029,bella,,-1961null,null,coffee/coff
eeTest;省份00,产品名称00,产品种类00,咖啡因类型00;QCT:<4,,_MAX_ROWS:500000>省份00产品名称00产品种类00咖啡因类型00((UnaryNode[a:[BCol<省份>], ne
g]ORUnaryNode[a:[BCol<产品名称>], neg])ORUnaryNode[a:[BCol<咖啡类型>], neg])ORUnaryNode[a:[BCol<产品种类>], neg])}; by g5.qry.jdbc.JDBCExecutor
$SQLRunnable@63bd14c [ts=1624203730193, report=null, wts=0, cost=0, qryCost=0, grpCost=0, group=UNKNOWN, query=UNKNOWN, ready=true]
```

#### 1.2 start to get connection

开始获取数据库连接，记录 **时间**，**数据库类型**，**查询 ID**。

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log# cat bi.log |grep 'start to get connection'
[2021-06-20 23:42:10.208] |[INFO] |[779c2e1c06c14021aa96a3ca80801bd7] |[admin] |[g5.qry.jdbc.JDBCExecutor.init0(JDBCExecutor.java:990)] |[DA
SHBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][下拉列表过滤3] g5.qry.sql.impl.MySQLExecutor@3b0777d9[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] start to get connection
[2021-06-20 23:42:10.822] |[INFO] |[81f681c1ee3f40a58bf0b44d55579ed9] |[admin] |[g5.qry.jdbc.JDBCExecutor.init0(JDBCExecutor.java:990)] |[DA
SHBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@5574364b[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] start to get connection
[2021-06-20 23:42:10.822] |[INFO] |[81f681c1ee3f40a58bf0b44d55579ed9] |[admin] |[g5.qry.jdbc.JDBCExecutor.init0(JDBCExecutor.java:990)] |[DA
SHBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@5574364b[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] start to get connection
```

#### 1.3 success get connection

成功获取到连接，记录 **时间**，**数据库类型**，**查询 ID**，**连接 ID**。

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log# cat bi.log |grep 'success get connection'
[2021-06-20 23:42:10.208] |[INFO] |[779c2e1c06c14021aa96a3ca80801bd7] |[admin] |[g5.qry.jdbc.JDBCExecutor.init0(JDBCExecutor.java:998)] |[DA
SHBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][下拉列表过滤3] g5.qry.sql.impl.MySQLExecutor@3b0777d9[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] success get connection: g5.qry.sql.impl.UConn@5972c234
[2021-06-20 23:42:10.822] |[INFO] |[81f681c1ee3f40a58bf0b44d55579ed9] |[admin] |[g5.qry.jdbc.JDBCExecutor.init0(JDBCExecutor.java:998)] |[DA
SHBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@5574364b[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] success get connection: g5.qry.sql.impl.UConn@5972c234
```

#### 1.4 start to execute SQL:

打印执行的 sql，记录 **时间**，**查询 ID**，**sql 语句**。

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log# grep -A 10 'start to execute SQL:' bi.log
[2021-06-20 23:42:10.288] |[INFO] |[779c2e1c06c14021aa96a3ca80801bd7] |[admin] |[g5.qry.jdbc.JDBCExecutor.netRS(JDBCExecutor.java:1432)] |[D
ASHBOARD: [coffee直连20210620234210-320<test/咖啡直连.db>][下拉列表过滤3] g5.qry.sql.impl.MySQLExecutor@3b0777d9[child-g5.qry.sql.impl.MySQLExe
cutor@26aeb117] start to execute SQL:
SELECT '省份', '产品名称', '产品种类', '咖啡因类型'
FROM bella.coffeetest
WHERE (( ('省份' IS NOT NULL) OR ( '产品名称' IS NOT NULL)) OR ( '咖啡因类型' IS NOT NULL)) OR ( '产品种类' IS NOT NULL)
GROUP BY '省份', '产品名称', '产品种类', '咖啡因类型'
ORDER BY '省份' ASC, '产品名称' ASC, '产品种类' ASC, '咖啡因类型' ASC
LIMIT 500000
[2021-06-20 23:42:10.696] |[INFO] |[128ab7b67ef648a792a2e4c6880356d3] |[SYSTEM] |[g5.mem.LinkMgr.adjustMem(LinkMgr.java:185)] |[OTHER: Not
need adjust mem, free: 1657920856, total: 2147483648]
```

## 1.5 init finished

标志某个 sql 执行完成，记录 **执行完成时间，报告，执行 sql 花费时间。**

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/log# cat bi.log |grep 'init finished'
[2021-06-20 23:42:10.728] |-[INFO] |-[779c2e1c06c14021aa96a3ca80801bd7] |-[admin] |-[g5.qry.jdbc.JDBCExecutor.init(JDBCExecutor.java:150)] |-[DAS
HBOARD: [coffee直连20210620234210-320<test/coffee直连.db>][下拉列表过滤3] g5.qry.sql.impl.MySQLExecutor@3b0777d9[child-g5.qry.sql.impl.MySQLExec
utor@26aeb117] init finished, time occupied: 535ms, grid: g5.grid.impl.QGrid@34a03cfc-[QColumn[省份]<r-1,type-DynamicStringSeg@810062373<false,0>
[0]>, QColumn[产品名称]<r-1,type-DynamicStringSeg@1235387477<false,0>[0]>, QColumn[产品类型]<r-1,type-DynamicStringSeg@2082056055<false,0>[0]>, Q
Column[咖啡因类型]<r-1,type-DynamicStringSeg@2082056055<false,0>[0]>]>.]
[2021-06-20 23:42:11.538] |-[INFO] |-[81f681c1ee3f40a58bf0b44d55579ed9] |-[admin] |-[g5.qry.jdbc.JDBCExecutor.init(JDBCExecutor.java:150)] |-[DAS
HBOARD: [coffee直连20210620234210-320<test/coffee直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@696bf48e[child-g5.qry.sql.impl.MySQLExecutor@68a
23d08] init finished, time occupied: 731ms, grid: g5.grid.impl.QGrid@5332920f-[QColumn[DistinctCount_预算_边际利润]<r-1,type-DynamicBILongSeg@143
588039<false,0>[0]>, QColumn[DistinctCount_订单ID]<r-1,type-DynamicBILongSeg@830682159<false,0>[0]>, QColumn[DistinctCount_市场开销]<r-1,type-Dyn
amicBILongSeg@1817870028<false,0>[0]>, QColumn[DistinctCount_利润]<r-1,type-DynamicBILongSeg@1790592641<false,0>[0]>, QColumn[DistinctCount_总成
本]<r-1,type-DynamicBILongSeg@214122091<false,0>[0]>, QColumn[DistinctCount_销售额]<r-1,type-DynamicBILongSeg@617173392<false,0>[0]>, QColumn[Dis
tinctCount_边际利润]<r-1,type-DynamicBILongSeg@458942427<false,0>[0]>, QColumn[DistinctCount_预算_销售]<r-1,type-DynamicBILongSeg@602678399<false
,0>[0]>, QColumn[DistinctCount_区域代码]<r-1,type-DynamicBILongSeg@919172623<false,0>[0]>, QColumn[DistinctCount_预算_销售成本]<r-1,type-DynamicB
ILongSeg@1491264663<false,0>[0]>]>.]
```

## 1.6 init slow

记录慢查询情况，**执行完成时间，数据库，查询唯一 ID，花费时间。**

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/log# cat bi.log |grep -a 'init slow'
2021年5月31日 08:01:52.365[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@3597a513 init slow, cost: 52220ms.
2021年5月31日 08:02:28.303[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@43b0c6d0 init slow, cost: 35836ms.
2021年5月31日 08:04:49.790[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.SparkHiveExecutor@29311da9 init slow, cost: 141417ms.
2021年5月31日 08:05:04.598[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.SparkHiveExecutor@5cb31e95 init slow, cost: 294809ms.
2021年5月31日 08:05:19.856[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@601a8b41 init slow, cost: 19839ms.
2021年5月31日 08:05:51.665[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.SparkHiveExecutor@5cd9b116 init slow, cost: 61870ms.
2021年5月31日 08:06:40.389[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.SparkHiveExecutor@7f2531c6 init slow, cost: 95181ms.
2021年5月31日 08:08:02.834[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.SparkHiveExecutor@169a7be0 init slow, cost: 130343ms.
2021年5月31日 08:08:12.465[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.SparkHiveExecutor@243b4594 init slow, cost: 91742ms.
2021年5月31日 08:08:23.656[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@f729b4d init slow, cost: 14348ms.
2021年5月31日 08:09:17.902[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@75e79ea8 init slow, cost: 17890ms.
2021年5月31日 08:09:55.581[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@3657198b init slow, cost: 81140ms.
2021年5月31日 08:09:55.757[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@50f8f873 init slow, cost: 81267ms.
2021年5月31日 08:09:55.774[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@20bc2fae init slow, cost: 81333ms.
2021年5月31日 08:09:55.817[INFO]QueryPool$QueryRunnable.run: g5.qry.sql.impl.ImpalaExecutor@6eb105d9 init slow, cost: 81382ms.
```

## 1.7 load data finished

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/log# cat bi.log |grep 'load data finished'
[2021-06-20 23:42:10.744] |-[INFO] |-[779c2e1c06c14021aa96a3ca80801bd7] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:304)] |-[DASHBO
ARD: [coffee直连20210620234210-320<test/coffee直连.db>][下拉列表过滤3] g5.qry.sql.impl.MySQLExecutor@3b0777d9[child-g5.qry.sql.impl.MySQLExecu
tor@26aeb117] load data finished, time occupied: 0, load rows: 177, cancel:false, grid: g5.grid.impl.QGrid@34a03cfc.]
[2021-06-20 23:42:11.542] |-[INFO] |-[81f681c1ee3f40a58bf0b44d55579ed9] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:304)] |-[DASHBO
ARD: [coffee直连20210620234210-320<test/coffee直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@696bf48e[child-g5.qry.sql.impl.MySQLExecutor@68a23d
08] load data finished, time occupied: 0, load rows: 1, cancel:false, grid: g5.grid.impl.QGrid@5332920f.]
[2021-06-20 23:42:12.009] |-[INFO] |-[81f681c1ee3f40a58bf0b44d55579ed9] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:304)] |-[DASHBO
ARD: [coffee直连20210620234210-320<test/coffee直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@5574364b[child-g5.qry.sql.impl.MySQLExecutor@5e91b1
72] load data finished, time occupied: 1, load rows: 2, cancel:false, grid: g5.grid.impl.QGrid@7ad54e12.]
[2021-06-20 23:42:18.931] |-[INFO] |-[81f681c1ee3f40a58bf0b44d55579ed9] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:304)] |-[DASHBO
ARD: [coffee直连20210620234210-320<test/coffee直连.db>][图表1] g5.qry.sql.impl.MySQLExecutor@420489e4[child-g5.qry.sql.impl.MySQLExecutor@545ba2
d] load data finished, time occupied: 5311, load rows: 4248, cancel:false, grid: g5.grid.impl.QGrid@7ca5a04b.]
```



## 2.集市相关

### 2.1 Assign MAP tasks for job key: 'Job@IP\_ID' with plan

记录的是一个集市计算的 job，查询的所有 zb 的情况，如果特别多，是有问题的，建议使用文件过滤。

```
[2021-06-22 23:04:02.730] |-[INFO] |-[dc320f3bcb74cbfb510afa8dbe39c44] |-[admin] |-[g5.dc.mr.impl.MRUtil.assignM(MRUtil.java:186)] |-[DASHBOARD:
|coffeeTest-集市20210622230402-137<coffeeTest-集市_db>|[下拉列表过滤3]
assign MAP tasks for job key: Job@127.0.1.1_1422' with plan
MRFilePlan(
MRFilePlan for GCFile: qry_33bb7a75.coffeeTest/blk.33bb7a730.0.zb<127.0.1.1,load-3,cpu-10,rPoolPendingCnt-0>
MRFilePlan for GCFile: qry_33bb7a75.coffeeTest/blk.33bb7a730.1.zb<127.0.1.1,load-3,cpu-10,rPoolPendingCnt-0>
MRFilePlan for GCFile: qry_33bb7a75.coffeeTest/blk.33bb7a730.2.zb<127.0.1.1,load-3,cpu-10,rPoolPendingCnt-0>
)
```

### 2.2 executed one MAP task

记录一个 zb 文件单次执行的情况，执行完成时间，对应 job,行数，计算花费的时间 (ms)。

```
root@LAPTOP-CCVF584B:/mnt/e/01-log/log/M# cat bi_log* |grep -a 'one MAP task'
2021-05-24 18:10:53.418 |p |3e468f82ae51586131884c |-[INFO] |g5.dc.mr.GMapTask.exec(GMapTask.java:385) |Host |47' executed one M
AP task on '组/qry_9ad4c0d1 |数据/blk.665ece511575.1.zb', for job key: Job@206_2416409', with rows 0', time elapsed 402ms.
2021-05-24 18:10:54.481 |p |01946e4bba62ecd3e639c5f |-[INFO] |g5.dc.mr.GMapTask.exec(GMapTask.java:385) |Host |47' executed one M
AP task on '组/qry_9ad4c0d1 |数据/blk.665ece511575.1.zb', for job key: Job@206_2416413', with rows 0', time elapsed 1777ms.
2021-05-24 18:10:54.591 |p |48bb401946e4bba62ecd3e639c5f |-[INFO] |g5.dc.mr.GMapTask.exec(GMapTask.java:385) |Host |47' executed one M
AP task on '组/qry_9ad4c0d1 |数据/blk.665ece511575.1.zb', for job key: Job@206_2416517', with rows 0', time elapsed 1440ms.
2021-05-24 18:10:54.835 |p |8bb401946e4bba62ecd3e639c5f |-[INFO] |g5.dc.mr.GMapTask.exec(GMapTask.java:385) |Host |47' executed one M
AP task on '组/qry_9ad4c0d1 |数据/blk.665ece511575.1.zb', for job key: Job@206_2416404', with rows 0', time elapsed 2121ms.
2021-05-24 18:10:54.860 |p |8bb401946e4bba62ecd3e639c5f |-[INFO] |g5.dc.mr.GMapTask.exec(GMapTask.java:385) |Host |47' executed one M
AP task on '组/qry_9ad4c0d1 |数据/blk.665ece511575.1.zb', for job key: Job@206_2416424', with rows 0', time elapsed 1785ms.
2021-05-24 18:10:54.875 |p |8bb401946e4bba62ecd3e639c5f |-[INFO] |g5.dc.mr.GMapTask.exec(GMapTask.java:385) |Host |47' executed one M
AP task on '组/qry_9ad4c0d1 |数据/blk.665ece511575.1.zb', for job key: Job@206_2416540', with rows 0', time elapsed 1699ms.
```

备注：

- ✓ zb 文件是指集市文件，存储在 M 节点 Yonghong/cloud 下，所以这个执行记录是在 M 节点日志中。
- ✓ 不是所有的版本都会记录行数，比较新的版本才会记录。
- ✓ 一个 zb 的 map 执行时间和 zb 的大小以及具体的运算关系很大，map 任务是并行的，并行程度取决于线程数。

### 2.3 executed REDUCE task

记录 reduce 任务执行的情况，执行完成时间，报告名称，组件名，zb 文件，Job ID，花费时间。

```

root@LAPTOP-CCVFS8KB:/mnt/e/01-Log/Log# cat bi2.log |grep "executed REDUCE task"
[2021-06-22 23:04:02.833] |-[INFO] |-[dc320f3bcb74cbfb518af8db39c44] |-[admin] |-[g5.dc.mr.GredTask.add(GredTask.java:323)] |-[DASHBOARD: [coffeeTest-集市
20210622230402-137<coffeeTest-集市.db>][下拉列表过滤3] Host '127.0.1.1' executed REDUCE task on 'qry_33bb7a75.coffeeTest/blk.33bb7a730.2.zb]', for job key:
'Job@127.0.1.1.1422', in 5ms, prepare: 3ms, lock: 0ms, execute: 5ms.
[2021-06-22 23:04:03.028] |-[INFO] |-[dc320f3bcb74cbfb518af8db39c44] |-[admin] |-[g5.dc.mr.GredTask.add(GredTask.java:323)] |-[DASHBOARD: [coffeeTest-集市
20210622230402-137<coffeeTest-集市.db>][下拉列表过滤3] Host '127.0.1.1' executed REDUCE task on 'qry_33bb7a75.coffeeTest/blk.33bb7a730.0.zb]', for job key:
'Job@127.0.1.1.1422', in 2ms, prepare: 1ms, lock: 0ms, execute: 1ms.
[2021-06-22 23:04:03.030] |-[INFO] |-[dc320f3bcb74cbfb518af8db39c44] |-[admin] |-[g5.dc.mr.GredTask.add(GredTask.java:323)] |-[DASHBOARD: [coffeeTest-集市
20210622230402-137<coffeeTest-集市.db>][下拉列表过滤3] Host '127.0.1.1' executed REDUCE task on 'qry_33bb7a75.coffeeTest/blk.33bb7a730.1.zb]', for job key:
'Job@127.0.1.1.1422', in 5ms, prepare: 1ms, lock: 1ms, execute: 2ms.
[2021-06-22 23:04:03.543] |-[INFO] |-[b3617de93138471586f6d9dcafa08c752] |-[admin] |-[g5.dc.mr.GredTask.add(GredTask.java:323)] |-[DASHBOARD: [coffeeTest-集市
20210622230402-137<coffeeTest-集市.db>][图表1] Host '127.0.1.1' executed REDUCE task on 'qry_33bb7a75.coffeeTest/blk.33bb7a730.2.zb]', for job key: 'Job@12
7.0.1.1.1434', in 51ms, prepare: 3ms, lock: 0ms, execute: 48ms.
[2021-06-22 23:04:04.693] |-[INFO] |-[b3617de93138471586f6d9dcafa08c752] |-[admin] |-[g5.dc.mr.GredTask.add(GredTask.java:323)] |-[DASHBOARD: [coffeeTest-集市
20210622230402-137<coffeeTest-集市.db>][图表1] Host '127.0.1.1' executed REDUCE task on 'qry_33bb7a75.coffeeTest/blk.33bb7a730.0.zb]', for job key: 'Job@12
7.0.1.1.1434', in 43ms, prepare: 4ms, lock: 0ms, execute: 39ms.
[2021-06-22 23:04:04.930] |-[INFO] |-[b3617de93138471586f6d9dcafa08c752] |-[admin] |-[g5.dc.mr.GredTask.add(GredTask.java:323)] |-[DASHBOARD: [coffeeTest-集市
20210622230402-137<coffeeTest-集市.db>][图表1] Host '127.0.1.1' executed REDUCE task on 'qry_33bb7a75.coffeeTest/blk.33bb7a730.1.zb]', for job key: 'Job@12
7.0.1.1.1434', in 89ms, prepare: 2ms, lock: 0ms, execute: 87ms.

```

### 3. 调度任务相关

#### 3.1 status:0/status:1/status:2

分别表示任务被触发，开始执行，执行完成。有多 C 节点的情况，可以通过这个判断任务

在哪个 C 节点执行。

```

root@LAPTOP-CCVFS8KB:/mnt/f# cat bi.log |grep 'status:0' |grep '同步test'
[2021-06-21 23:41:58.794] |-[INFO] |-[d3ebaf7f925c4527b5cc7775d92ba782] |-[admin] |-[g5.sched.SchedulerUtil.updateStatus(SchedulerUtil.java:1093)] |-[OTHER:
Update the job status. the job path:同步test, status:0]
[2021-06-21 23:42:05.944] |-[INFO] |-[d3ebaf7f925c4527b5cc7775d92ba782] |-[admin] |-[g5.sched.SchedulerUtil.updateStatus(SchedulerUtil.java:1093)] |-[OTHER:
Update the job status. the job path:同步test, status:0]
root@LAPTOP-CCVFS8KB:/mnt/f# cat bi.log |grep 'status:1' |grep '同步test'
[2021-06-21 23:42:06.002] |-[INFO] |-[d3ebaf7f925c4527b5cc7775d92ba782] |-[admin] |-[g5.sched.SchedulerUtil.updateStatus(SchedulerUtil.java:1093)] |-[SCHEDUL
ER: [同步test20210621234205_2312_0] Update the job status. the job path:同步test, status:1]
root@LAPTOP-CCVFS8KB:/mnt/f# cat bi.log |grep 'status:2' |grep '同步test'
[2021-06-21 23:42:12.187] |-[INFO] |-[d3ebaf7f925c4527b5cc7775d92ba782] |-[admin] |-[g5.sched.SchedulerUtil.updateStatus(SchedulerUtil.java:1093)] |-[SCHEDUL
ER: [同步test20210621234205_2312_0<同步test>][同步test] Update the job status. the job path:同步test, status:2]
root@LAPTOP-CCVFS8KB:/mnt/f#

```

#### 3.2 waiting to run

可以查到某个任务，准备运行，记录时间，任务唯一 ID。

```

root@LAPTOP-CCVFS8KB:/mnt/e/01-Log/Log/入市# cat bi.log |grep -a 'waiting to run' |grep '同步test'
[2021-06-22 20:39:22.616] |-[WARN] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[g5.sched.impl.SchedPool.add(SchedPool.java:155)] |-[OTHER:
Add job '同步test' into pool, waiting to run, with user: FSPrincipal<admin, name=admin>[admin_role] clientIP:192.168.0.224, locale=zh_CN, areaCo
de=null, mobileNumber=null, auth=true, with executor: g5.sched.impl.SchedExecutor@6c53f0af, manually:true]
root@LAPTOP-CCVFS8KB:/mnt/e/01-Log/Log/入市#

```

#### 3.2 loaded 10000

入市任务的 sql 查询，返回 10000 条数据。记录了 时间，查询 ID，加载 10000 行花费的时间。

```

[2021-06-22 20:39:22.719] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:288)] |-[SCHEDULER: [同步test202106222039
22_6092_60<同步test>][同步test] g5.qry.sql.impl.MySQLExecutor@83986b366 loaded 10000 rows, time occupied: 45]
[2021-06-22 20:39:22.768] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:288)] |-[SCHEDULER: [同步test202106222039
22_6092_60<同步test>][同步test] g5.qry.sql.impl.MySQLExecutor@83986b366 loaded 10000 rows, time occupied: 49]
[2021-06-22 20:39:22.813] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:288)] |-[SCHEDULER: [同步test202106222039
22_6092_60<同步test>][同步test] g5.qry.sql.impl.MySQLExecutor@83986b366 loaded 10000 rows, time occupied: 45]
[2021-06-22 20:39:22.852] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[g5.qry.sql.ARSExecutor.load(ARSExecutor.java:288)] |-[SCHEDULER: [同步test202106222039
22_6092_60<同步test>][同步test] g5.qry.sql.impl.MySQLExecutor@83986b366 loaded 10000 rows, time occupied: 39]

```

#### 3.3 load data finished

记录了查询的数据全部加载完成花费的时间，加载的行数。

```

root@LAPTOP-CCVF58K8:/mnt/e/01-log/log/入市# cat bil.log |grep -a '3986b366' |grep 'load data finished'
[2021-06-22 20:39:36.291] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_ary.sql.ARSExecutor.load(ARSExecutor.java:286)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5.ary.sql.impl.MySQLExecutor@3986b366 load data finished, grid: g5.grid.impl.QGrid@41c15f4, time occupied: 13617, load rows: 2174976, can cel:false]

```

### 3.4 begin to execute g5.dc.split.Splitter

表示开始进行 split,split 开始不会等到数据全部加载完成再开始，加载一部分数据就会进行 split。记录**时间**，**QueryTask ID**，**splitter ID**。

```

root@LAPTOP-CCVF58K8:/mnt/e/01-log/log/入市# cat bil.log |grep -a '同步test' |grep 'begin to execute'
[2021-06-22 20:39:24.134] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_sched.jobs.AbstractCloudTask.runSplitter(AbstractCloudTask.java:91)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5_sched.jobs.QueryTask@6ec2be12 begin to execute g5.dc.split.Splitter@1e2cd258]

```

### 3.5 split grid to blocks over

split 完成，记录 **时间**，**splitter ID**，**花费的时间**。

```

root@LAPTOP-CCVF58K8:/mnt/e/01-log/log/入市# cat bil.log |grep -a '同步test' |grep 'split grid to blocks over'
[2021-06-22 20:39:38.442] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.split.Splitter.complete(Splitter.java:429)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5_dc.split.Splitter@1e2cd258 split grid to blocks over, time occupied 14388ms, canceled: false.]

```

### 3.6 create gsfile

生成集市文件 (zb 文件)，生成的个数跟数据量有关 (大约 100W 条数据一个 zb)，跟文件大小无关，文件大小跟字段数，字段长度有关。

```

root@LAPTOP-CCVF58K8:/mnt/e/01-log/log/入市# cat bil.log |grep -a '同步test' |grep 'create gsfile'
[2021-06-22 20:39:38.444] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.split.Splitter.createGSFiles(Splitter.java:155)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5_dc.split.Splitter@1e2cd258 create gsfile: qry_33bb7a75.coffeeTest/blk.33bb7a730.0.zb]
[2021-06-22 20:39:38.444] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.split.Splitter.createGSFiles(Splitter.java:155)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5_dc.split.Splitter@1e2cd258 create gsfile: qry_33bb7a75.coffeeTest/blk.33bb7a730.1.zb]
[2021-06-22 20:39:38.445] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.split.Splitter.createGSFiles(Splitter.java:155)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5_dc.split.Splitter@1e2cd258 create gsfile: qry_33bb7a75.coffeeTest/blk.33bb7a730.2.zb]

```

### 3.7 Succeed to modify file

将 zb 文件存储到 M 节点

```

root@LAPTOP-CCVF58K8:/mnt/e/01-log/log/入市# cat bil.log |grep -a '同步test' |grep 'Succeed to modify file'
[2021-06-22 20:39:38.091] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.fs.GMFiles.modify(GMFiles.java:75)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] Succeed to modify file of node 127.0.1.1, path: qry_33bb7a75.coffeeTest/blk.33bb7a730.0.zb]
[2021-06-22 20:39:38.725] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.fs.GMFiles.modify(GMFiles.java:75)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] Succeed to modify file of node 127.0.1.1, path: qry_33bb7a75.coffeeTest/blk.33bb7a730.1.zb]
[2021-06-22 20:39:38.753] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_dc.fs.GMFiles.modify(GMFiles.java:75)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] Succeed to modify file of node 127.0.1.1, path: qry_33bb7a75.coffeeTest/blk.33bb7a730.2.zb]

```

### 3.8 run task over

任务运行完成。

```

root@LAPTOP-CCVF58K8:/mnt/e/01-log/log/入市# cat bil.log |grep -a '同步test' |grep 'run task over'
[2021-06-22 20:39:38.763] |-[INFO] |-[add075e1ba6a49dc8bac8c7f0f0c08b5] |-[admin] |-[gs_sched_impl.SchedExecutor$TaskRunnable.run(SchedExecutor.java:671)] |-[SCHEDULER: [同步test20210622203922_6092_60<同步test>][同步test] g5_sched.jobs.QueryTask@6ec2be12 run task over, job: 同步test]

```



## 4.其他

### 4.1 GRLink Report

结果缓冲区占用情况。

max jvm mem: 给 jvm 分配的最大内存

used jvm mem: 当前使用内存

free jvm mem: 空闲内存

total link mem: 结果缓冲区, 结果缓冲区是大部分查询结果的逻辑存储空间, 一旦超出缓冲区的大小, 结果数据将被交换到磁盘, 当再次使用时再从磁盘交换回内存, 缓冲区不宜过小, 但太大容易导致计算中间过程出现内存溢出。

Total of in mem GResults: 表示当前在缓冲区中的对象数以及总共注册的个数, 除了一些正在注册的对象, 说明很多有一些对象交换到了磁盘。交换到磁盘会受磁盘 IO 瓶颈的影响, 导致运算速度下降。

备注: mem.serial.debug 需打开

```
bl.log 2021.05.31 01.49.37:2021年5月31日 01:41:16.003[INFO]GRLink$Reporter.run: --GRLink Report, link count: 0, max jvm mem: 114688M, used jvm mem: 80958M, free jvm mem: 3729M, total link mem: 19114M
bl.log 2021.05.31 01.49.37- GroupLinkMgr0[191M of 2389M]
bl.log 2021.05.31 01.49.37- In mem GResults: 40619
bl.log 2021.05.31 01.49.37- contains 2551 GroupLink
bl.log 2021.05.31 01.49.37- GroupLinkMgr1[207M of 2389M]
bl.log 2021.05.31 01.49.37- In mem GResults: 40547
bl.log 2021.05.31 01.49.37- contains 2546 GroupLink
bl.log 2021.05.31 01.49.37- GroupLinkMgr7[197M of 2389M]
bl.log 2021.05.31 01.49.37- In mem GResults: 40688
bl.log 2021.05.31 01.49.37- contains 2546 GroupLink
bl.log 2021.05.31 01.49.37- Total of in mem GResults: 324767 of registered GResults: 330792
bl.log 2021.05.31 01.49.37- Some GResults might be in construction, which occupy mem too.
```

### 4.2 Create dashboard

记录打开报告的时间, 报告路径, 报告名称, 本次打开唯一 ID。

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log/集市# cat bi.log* |grep -a 'Create dashboard'
2021-04-27 09:03:02.423 |p_c 35677f0f347388464810f47aaa74e |-[INFO] |g5.sv.db.impl.DBServiceImpl.open(DBServiceImpl.java:345) | Create dashboard
: 团队 20210427090302-577, total=43
2021-04-27 09:03:08.384 |p 92197347db815a3aff2ba688c |-[INFO] |g5.sv.db.impl.DBServiceImpl.open(DBServiceImpl.java:345) | Create dashboard
: 未来 20210427090308-1500, total=45
2021-04-27 09:03:28.718 |v_s g-aadt d626c43d494c1198cc8bd6af7 |-[INFO] |g5.sv.db.impl.DBServiceImpl.open(DBServiceImpl.java:345) | Create dashboard:
: 明细 20210427090328-1848, total=45
2021-04-27 09:04:18.597 |p_c 74d1581f8ce57ad5c5ede |-[INFO] |g5.sv.db.impl.DBServiceImpl.open(DBServiceImpl.java:345) | Create dashboard:
: 未来 20210427090418-1011, total=47
```

### 4.3 first page area

报告首屏打开的情况, 打开的时间, 报告名, 本次打开的唯一 id, 打开花费的时间(ms)。

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log#
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log# cat bi.log* |grep -a 'first page area'
2021年5月31日 08:09:57.325[INFO]RTDashboard.removePending: Open db 首页_当月-310 first page area cost 846
2021年5月31日 08:09:57.522[INFO]RTDashboard.removePending: Open db 首页_当月-1513 first page area cost 821
2021年5月31日 08:10:44.677[INFO]RTDashboard.removePending: Open db 首页-641 first page area cost 1271
2021年5月31日 08:11:13.666[INFO]RTDashboard.removePending: Open db 周报-1812 first page area cost 30220
2021年5月31日 08:13:03.198[INFO]RTDashboard.removePending: Open db 门店诊断总览-291 first page area cost 167259
2021年5月31日 08:28:36.902[INFO]RTDashboard.removePending: Open db 首页-1755 first page area cost 844
2021年5月31日 08:28:38.444[INFO]RTDashboard.removePending: Open db 首页_当月-266 first page area cost 4531
2021年5月31日 08:29:18.738[INFO]RTDashboard.removePending: Open db 日报-78 first page area cost 42664
```

```
root@LAPTOP-CCVF58KB:/mnt/e/01-log/Log/集市# cat bi.log* |grep -a 'first page area'
2021-04-27 09:03:12.082 |p_c 35677f0f347388464810f47aaa74e |-[INFO] |g5.db.RTDashboard.checkPendings(RTDashboard.java:670) | Open db 指标
20210427090302-577 first page area cost 9659
2021-04-27 09:03:21.483 |p 83b9f99b41d3910997a83ce1a4ed |-[INFO] |g5.db.RTDashboard.checkPendings(RTDashboard.java:670) | Open db 未来
: 未来 20210427090308-1500 first page area cost 13099
2021-04-27 09:03:35.615 |v_s g-aadt 4c4a3bbc4ab275c |-[INFO] |g5.db.RTDashboard.checkPendings(RTDashboard.java:670) | Open db 明细
: 明细 20210427090328-1848 first page area cost 26397
2021-04-27 09:04:28.667 |p_c 2ef0c4935890f9307af1ecc8c |-[INFO] |g5.db.RTDashboard.checkPendings(RTDashboard.java:670) | Open db 未来
20210427090418-1011 first page area cost 18070
```







## 4.9 join result

复合查询执行时间和加载记录数。

```
root@LAPTOP-CCVF58K8:/mnt/e/01-日志分析/ # cat bi* | grep -a 'join result'
2021-05-24 18:55:47.633 [-] 3068f34918a6d22359936408c |-[INFO]| g5.grid.join.JExecutor.execute(JExecutor.java:57) | The join result for 'g5.grid.join.Def36
rid@63a8b240' has row count: '3', cost: 31ms, Related Query: '重点 5286-45' Related Query: '组合数据' Related Element:
'图表5' Related column: '时间_last' Run user: 'r' u'
```

## 4.10 Concurrent map tasks are limited to 1, so waiting

说明出现了明细查询排队的情况，明细查询加载大量数据会占用较多内存，降低系统执行效率，系统默认限制明细查询的执行线程数未1，较多的明细查询需要排队进行。

```
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@56da6e46 [ts=1624520050509, report=null, wts=7, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@76553e7 [ts=1624520050509, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@765ab42b [ts=1624520050509, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@83cfc41a [ts=1624520050510, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@78c0700b [ts=1624520050510, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@1fd994ac [ts=1624520050510, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@857a79c0 [ts=1624520050510, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@841e70f [ts=1624520050510, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@82eab06f9 [ts=1624520050510, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@10dbab38 [ts=1624520050511, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@83cfc41a [ts=1624520050511, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@86d186ee [ts=1624520050511, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@859516ac [ts=1624520050511, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@7014912c [ts=1624520050511, report=null, wts=8, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@87b7c331 [ts=1624520050512, report=null, wts=7, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@17124494 [ts=1624520050512, report=null, wts=7, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@2d1fb192 [ts=1624520050512, report=null, wts=7, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@70a8fa1e [ts=1624520050512, report=null, wts=7, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@83b9e4eb [ts=1624520050513, report=null, wts=6, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@6464f62 [ts=1624520050513, report=null, wts=6, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=5429]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@2d1fb192 [ts=1624520050512, report=null, wts=10, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=542]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@70a8fa1e [ts=1624520050512, report=null, wts=10, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=542]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@83b9e4eb [ts=1624520050513, report=null, wts=10, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=542]
| Concurrent map tasks are limited to 1, so waiting: g5.dc.mr.s.MapProcMapRunnable@65624f62 [ts=1624520050513, report=null, wts=10, cost=0, qryCost=0, grpCost=0, groupyh-度量系列-过滤列-1154, query=542]
```

## 4.11 Find the slow script expression

查询 执行慢的 不能下推的 js 表达式。

```
root@LAPTOP-CCVF58K8: # cat bi* | grep -a 'Find the slow script expression'
2021-05-24 18:47:30.749 [-] lvanwu-baf44a3b10fa4920b8852242f42a52e1 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script expr
ession: '
2021-05-24 18:47:30.751 [-] lvanwu-8806ae888d7249c98550d5879ca7cda4 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script expr
ession: '
2021-05-24 18:47:30.749 [-] lvanwu-38d178f37a747269801fe8878d1f81 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script expr
ession: 'if(col["所属区域"]='东区',1,0)'; Unsupported expression: if(col["所属区域"]='东区',1,0)
2021-05-24 18:47:30.752 [-] lvanwu-c6b4dd05a4444a42afa47feb8ff1429e |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script expr
ession: '
2021-05-24 18:55:35.958 [-] l_yapingli-036f89bcfa2540c88b32e713d47cf538 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script
expression: ' var a = 1 - ( col["Sum_技术能力弱"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100'; Unsupported expression: var a
= 1 - ( col["Sum_技术能力弱"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 0 , a ] ) * 100
2021-05-24 18:55:35.959 [-] l_yapingli-036f89bcfa2540c88b32e713d47cf538 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script
expression: ' var a = 1 - ( col["Sum_解决时效差2"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100'; Unsupported expression: var a
= 1 - ( col["Sum_解决时效差2"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100
2021-05-24 18:55:35.960 [-] l_yapingli-036f89bcfa2540c88b32e713d47cf538 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script
expression: ' var a = 1 - ( col["Sum_服务态度差"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100'; Unsupported expression: var a
= 1 - ( col["Sum_服务态度差"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100
2021-05-24 18:55:35.960 [-] l_yapingli-036f89bcfa2540c88b32e713d47cf538 |-[WARN]| g5.meta.exp.ExpExecutor.getNode(ExpExecutor.java:54) | Find the slow script
expression: ' var a = 1 - ( col["Sum_服务态度差"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100'; Unsupported expression: var a
= 1 - ( col["Sum_服务态度差"] / col["Sum_反馈量"] / 0.03 ) ; min ( [ 1 , max ( [ 0 , a ] ) ] ) * 100
```

## 4.12 waiting to get grid cost

打印 1 次是正常的，如果一个查询 ID，打印多次，说明该查询长时间未反馈执行结果。每隔 5 分钟打印 1 次。

```
root@LAPTOP-CCVE58KB: /mnt/e/01-log/ # cat bi* | grep -a 'waiting to get grid cost' | grep '53c29e9'
2021年5月31日 07:09:07.013[WARNING]QueryExecutor.grid0: g5.qry.sql.impl.ImpalaExecutor@53c29e9 waiting to get grid cost 388886ms. Related Sched Job: '20-
自助分 销量分布' Related Query: '自助分 销量分布' Run user: '自助分'
2021年5月31日 07:14:07.019[WARNING]QueryExecutor.grid0: g5.qry.sql.impl.ImpalaExecutor@53c29e9 waiting to get grid cost 608013ms. Related Sched Job: '20-
自助分 销量分布' Related Query: '自助分 销量分布' Run user: '自助分'
2021年5月31日 07:19:07.026[WARNING]QueryExecutor.grid0: g5.qry.sql.impl.ImpalaExecutor@53c29e9 waiting to get grid cost 908020ms. Related Sched Job: '20-
自助分 销量分布' Related Query: '自助分 销量分布' Run user: '自助分'
```

## 三、常用工具

1. Notepad++
2. Linux (cat/tail/grep/awk)
3. V9.3 版本自带日志分析
4. 动态性能分析报告

## 四、日志分析学习方法

- ◇ 结合现象分析日志。
- ◇ 拆分。如果一个报告慢，可以先进行拆分，单个组件是否慢，分析单个组件。
- ◇ 缩小时间范围，记录报告开始执行时间，任务开始执行时间，可以缩小查看日志的范围
- ◇ 记录关键信息（报告名，任务名）找到关键 id，查询 id 相关信息
- ◇ 关注时间，ms
- ◇ 关注时间戳，可以大致判断时间花在哪里

## 五、日志搜集

性能类日志收集										
	bi.log	jvm.log	jmap	jstack	catalina.out	access (%d)	hs_err	/var/log/messages	/var/log	F12
宕机	✓	✓	✓	✓	✓		✓	✓	✓	
白屏	✓	✓	✓	✓	✓	✓				✓
直连慢	✓			✓		✓				
集市慢	✓	✓	✓	✓		✓				
系统慢	✓	✓	✓	✓	✓	✓				✓
报错类	✓									✓

集成类信息收集					
	web.xml	server.xml	bi.log	bi.properties(db)	F12
API	✓	✓	✓	✓	✓
LDAP	✓	✓	✓	✓	✓
钉钉	✓	✓	✓	✓	✓
企业微信	✓	✓	✓	✓	✓
飞书	✓	✓	✓	✓	✓
单点	✓	✓	✓	✓	✓
welink	✓	✓	✓	✓	✓
URL	✓	✓	✓	✓	✓



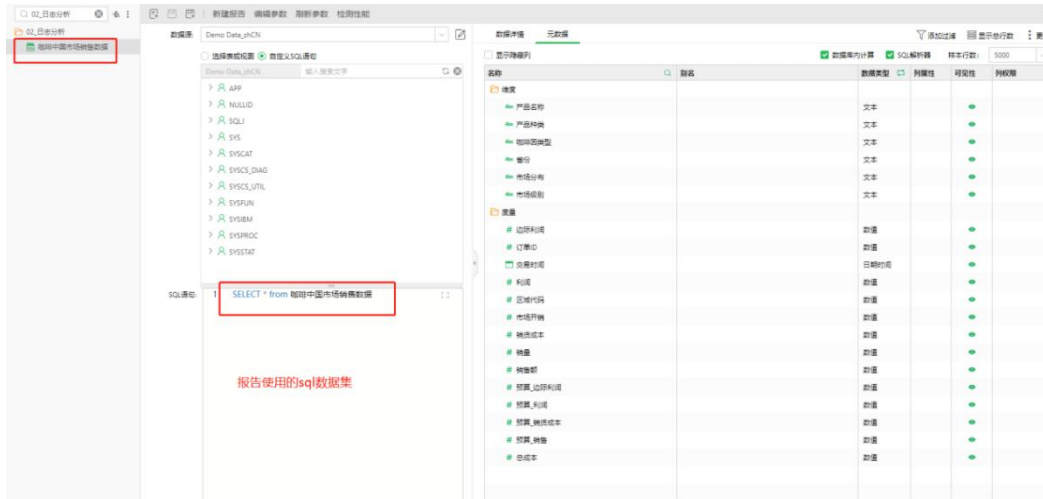
常见性能问题排查  
方案.xlsx

## 六、常见场景/逻辑

### 1. 直连数据库的报告执行的 sql 是什么

直连数据库的报告打开快慢，很大程度上取决于数据库执行 sql 的情况，同时执行的 sql 也并不是简单的 sql 数据集中咱们写的 sql，而是跟咱们制作报告进行的计算有关，比如，咱们进行了求和，平均等，这些正常情况下都会到数据库执行。

Eg:



实际这两个组件执行的 sql, 并不是数据集中的 SELECT \* from 咖啡中国市场营销数据 而是跟组件做的计算有关。所以当咱们直连的报告打开慢的时候, 不能说只是去看对应的数据集的 sql 执行是否快, 能否快速预览数据, 这样是不够的。

```

19 2021-05-17 15:53:47.002 |admin-61402d0c3a5464ab3deade12bf8bd |-[INFO] |g5.grid.impl.DynamicStringSeg.convert(DynamicStringSeg.java:137) | Slow DefStringSeg is used!
20 2021-05-17 15:53:47.019 |admin-61402d0c3a5464ab3deade12bf8bd |-[INFO] |g5.gryy.QQuery.getGrid(QQuery.java:1130) | Execute one query 'g5.gryy.QQuery$5c2a34b', result
  'g5.grid.MemRowGr148551b04b', occupied 100ms. Related Dashboard: '01直连报表的sql20210517155346-190' Related query: '咖啡中国市场营销数据' Related Element: '列表过滤!' Run
  user: 'admin' sql statement:
21 FROM "市场分布"
22 FROM 咖啡中国市场营销数据
23 WHERE "市场分布" IS NOT NULL
24 ORDER BY "市场分布" ASC
25 FETCH FIRST 1000 ROWS ONLY
26
27
28 2021-05-17 15:53:47.020 |admin-61402d0c3a5464ab3deade12bf8bd |-[INFO] |g5.agry.AQuery.getGrid(AQuery.java:560) | Execute one AQuery 'g5.agry.AQuery$4dab6c0', result
  'g5.grid.MemRowGr148551b04b', occupied 100ms.
29 2021-05-17 15:53:47.033 |admin-61402d0c3a5464ab3deade12bf8bd |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: db.memory.debug
30 2021-05-17 15:53:47.033 |admin-61402d0c3a5464ab3deade12bf8bd |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: db.memory.debug
31 2021-05-17 15:53:47.042 |admin-61402d0c3a5464ab3deade12bf8bd |-[INFO] |g5.grid.impl.DynamicStringSeg.convert(DynamicStringSeg.java:137) | Slow DefStringSeg is used!
32 2021-05-17 15:53:47.070 |admin-eb0bb7f93a24619b12b61a703e446 |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: sys.pngencoder.imageio
33 2021-05-17 15:53:47.191 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: db.memory.debug
34 2021-05-17 15:53:47.191 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.gryy.QQuery.getGrid(QQuery.java:1130) | Execute one query 'g5.gryy.QQuery$3d01c15f', result
  'g5.grid.MemRowGr148551b04b', occupied 0ms. Related Dashboard: '01直连报表的sql20210517155346-190' Related query: '咖啡中国市场营销数据' Related Element: '报表!' Run us
  'admin'
35 2021-05-17 15:53:47.191 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.agry.AQuery.getGrid(AQuery.java:560) | Execute one AQuery 'g5.agry.AQuery$73dbeb21', result
  'g5.grid.MemRowGr148551b04b', occupied 1ms.
36 2021-05-17 15:53:47.192 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.agry.HQuery.getGrid(HQuery.java:153) | Execute one HQuery 'g5.agry.HQuery$23982c1', result
  'g5.grid.MemRowGr148551b04b', occupied 1ms.
37 2021-05-17 15:53:47.200 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: db.memory.debug
38 2021-05-17 15:53:47.202 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.gryy.QueryPool.add(QueryPool.java:84) | executor QueryExecutor4 has been added into pool with grpCo
  [0] and grpCnt: [0]
39 2021-05-17 15:53:47.202 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.gryy.QQuery.getPreGrid(QQuery.java:1317) | g5.gryy.QQuery$7035278a waiting to get grid by
  g5.gryy.sql.impl.GenericExecutor$6d6ec714
40 2021-05-17 15:53:47.216 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.gryy.sql.SQLExecutor.init(SQLExecutor.java:426) | g5.gryy.sql.impl.GenericExecutor$6d6ec714 start to
  init, with param: {MAX_ROWS=500000, PARAN_0=[东部市场, 南部市场], fromVool1dbJoinNode=false, DISCARDED_GRID=false, SEL_STRATEGY_0=, DATA_CACHE_TYPE=6,
  _Running_QueryAlready_Added=true, MAX_FETCH_ROWS=500000, isMobile=false, _CONN_TIME_OFFSET=0}, by g5.gryy.sql.SQLExecutor$SQLRunnable$3f041906 [ts=162123027215, report=m
  wtsn, cost=0, grpCnt=0, grpCost=0, group=UNKNOWN, query=UNKNOWN, ready=true]
41 2021-05-17 15:53:47.230 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.gryy.sql.SQLExecutor.init(SQLExecutor.java:1372) | g5.gryy.sql.impl.GenericExecutor$6d6ec714 start t
  get connection
42 2021-05-17 15:53:47.231 |-[INFO] |g5.gryy.driver.ConnectionProvider.getConnection(ConnectionProvider.java:101) | Succeeded to get connection, classloader: Def Class loader.
43 2021-05-17 15:53:47.232 |admin-f080857f1b649ca909c23f04cf21af6 |-[INFO] |g5.gryy.sql.SQLExecutor.init(SQLExecutor.java:1390) | g5.gryy.sql.impl.GenericExecutor$6d6ec714 start t
  execute SQL
44 SELECT "市场分布", SUM("利润") as "利润_总和"
45 FROM 咖啡中国市场营销数据
46 WHERE "市场分布" IN ( ?, ? )
47 GROUP BY "市场分布"
48 ORDER BY "市场分布" ASC
49 FETCH FIRST 5000000 ROWS ONLY
50

```



## 2. 数据库内计算

默认保留勾选，执行数据集查询时将会对 SQL 进行封装，计算将会下推到数据库内执行，

取消勾选数据库内计算后，数据集的后续操作将不会在数据库内执行，计算将会交给内存处

理。

```
116 2021-05-18 10:30:10.397 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gqry.GQuery.getGrid(GQuery.java:1130) | Execute one query 'g5.gqry.DefGQuery#857b471', result
'g5.grid.MaxRowGrid#84ef1e90f', occupied lms. Related Dashboard: '02数据库内计算20210518103010-1235' Related Query: '数据库内计算Case' Related Element: '图表1' Run user:
'admin'
117 2021-05-18 10:30:10.397 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.agry.AQuery.getGrid(AQuery.java:560) | Execute one AQuery 'g5.agry.AQuery#845f8e9f9', result
'g5.grid.MaxRowGrid#84ef1e90f', occupied lms.
118 2021-05-18 10:30:10.397 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.agry.HQuery.getGrid(HQuery.java:153) | Execute one HQuery 'g5.agry.HQuery#857f56217', result
'g5.grid.MaxRowGrid#84ef1e90f', occupied lms.
119 2021-05-18 10:30:10.400 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gry.QueryPool.add(QueryPool.java:84) | executor QueryExecutor22 has been added into pool with grpCost:
{0} and qryCost: {0}.
120 2021-05-18 10:30:10.401 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gqry.GQuery.getPreGrid(GQuery.java:1317) | g5.gqry.SQLGQuery#47639eb1 waiting to get grid by
g5.gry.sql.impl.MySQLExecutor#198a6756
121 2021-05-18 10:30:10.408 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gry.sql.SQLExecutor.init(SQLExecutor.java:426) | g5.gry.sql.impl.MySQLExecutor#198a6756 start to init,
with params: {_MAX_ROWS=5000000, PARAM_0=茶, 花茶, 咖啡, fromVooltdbJoinNode=false, _DISCARDED_GRID=false, _SEL_STRATEGY=0, _DATA_CACHE_TYPE=6,
_Running_GQuery_Already_Added=true, _MAX_PERMIT_ROWS=5000000, isMobile=false, _CONN_TIME_OFFSET=0}, by g5.gry.sql.SQLExecutor$SQLRunnable#8441f5550 [ts=1621305010408, report=null,
wts=0, cost=0, qryCost=0, grpCost=0, group=UNKNOWN, query=UNKNOWN, ready=true]
122 2021-05-18 10:30:10.408 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: db.memory.debug
123 2021-05-18 10:30:10.424 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gry.sql.SQLExecutor.init0(SQLExecutor.java:1372) | g5.gry.sql.impl.MySQLExecutor#198a6756 start to get
connection
124 2021-05-18 10:30:10.424 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gry.sql.SQLExecutor.init0(SQLExecutor.java:1390) | g5.gry.sql.impl.MySQLExecutor#198a6756 start to
execute SQL:
125 SELECT 产品种类 AS LONG_COL_0, SUM(边际利润) AS LONG_COL_1, SUM(边际利润*利润) AS LONG_COL_2
126 FROM 咖啡中国市场销售数据
127 WHERE 产品种类 IN ( ?, ?, ? )
128 GROUP BY 产品种类
129 ORDER BY 产品种类 ASC
130 LIMIT 5000000
131 2021-05-18 10:30:10.431 |admin-4c95149e2851451e9ad960e0688e0ea8 |-[INFO] |g5.gry.sql.SQLExecutor.init0(SQLExecutor.java:455) | g5.gry.sql.impl.MySQLExecutor#198a6756 init finished,
grid: g5.grid.impl.QGrid#8b84553<{QColumn[产品种类]<-1,type=DynamicStringSeg#1163274156&false,0>{0}, QColumn[Sum_边际利润]<-1,type=DynamicObjectSeg#108932553&false,0>{0}>,
391 2021-05-18 10:32:09.126 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gqry.GQuery.getGrid(GQuery.java:1130) | Execute one query 'g5.gqry.DefGQuery#857b471', result
'g5.grid.MaxRowGrid#85e54e69', occupied lms. Related Dashboard: '02数据库内计算20210518103208-1110' Related Query: '数据库内计算Case' Related Element: '图表1' Run user:
'admin'
392 2021-05-18 10:32:09.126 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.agry.AQuery.getGrid(AQuery.java:560) | Execute one AQuery 'g5.agry.AQuery#829a997c', result
'g5.grid.MaxRowGrid#85e54e69', occupied lms.
393 2021-05-18 10:32:09.126 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.agry.HQuery.getGrid(HQuery.java:153) | Execute one HQuery 'g5.agry.HQuery#85587bb00', result
'g5.grid.MaxRowGrid#85e54e69', occupied lms.
394 2021-05-18 10:32:09.129 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gry.QueryPool.add(QueryPool.java:84) | executor QueryExecutor25 has been added into pool with grpCost:
{0} and qryCost: {0}.
395 2021-05-18 10:32:09.129 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gqry.GQuery.getPreGrid(GQuery.java:1317) | g5.gqry.DefGQuery#817411e7 waiting to get grid by
g5.gry.sql.impl.MySQLExecutor#8087b87
396 2021-05-18 10:32:09.130 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.AbsSetting.getBoolean(AbsSetting.java:358) | boolean value does not exist: db.memory.debug
397 2021-05-18 10:32:09.130 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gry.sql.SQLExecutor.init(SQLExecutor.java:426) | g5.gry.sql.impl.MySQLExecutor#8087b87 start to init,
with params: {_MAX_ROWS=0, _DISCARDED_GRID=false, _SEL_STRATEGY=0, _DATA_CACHE_TYPE=6, _Running_GQuery_Already_Added=true, _MAX_PERMIT_ROWS=5000000, isMobile=false,
_CONN_TIME_OFFSET=0}, by g5.gry.sql.SQLExecutor$SQLRunnable#857e2b118 [ts=1621305129130, report=null, wts=0, cost=0, qryCost=0, grpCost=0, group=UNKNOWN, query=UNKNOWN, ready=true]
398 2021-05-18 10:32:09.146 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gry.sql.SQLExecutor.init0(SQLExecutor.java:1372) | g5.gry.sql.impl.MySQLExecutor#8087b87 start to get
connection
399 2021-05-18 10:32:09.147 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gry.sql.SQLExecutor.init0(SQLExecutor.java:1390) | g5.gry.sql.impl.MySQLExecutor#8087b87 start to
execute SQL:
400 SELECT 订单ID, 交易时间, 市场级别, 市场分布
401 , 产品种类, 产品名称, 省份, 咖啡因类型, 销售成本, 边际利润
402 , 市场开销, 销量, 利润, 销售额, 总成本
403 FROM 咖啡中国市场销售数据
404 2021-05-18 10:32:09.149 |admin-426964dc68f3454086aaa382ed6c7703 |-[INFO] |g5.gry.sql.SQLExecutor.init(SQLExecutor.java:459) | g5.gry.sql.impl.MySQLExecutor#8087b87 init finished,
grid: g5.grid.impl.QGrid#8b84553<{QColumn[订单ID]<-1,type=DynamicIntSeg#1401402913&false,0>{0}, QColumn[交易时间]<-1,type=DynamicDateSeg#8250184163&false,0>{0}>,
```

### 3. 筛选组件逻辑

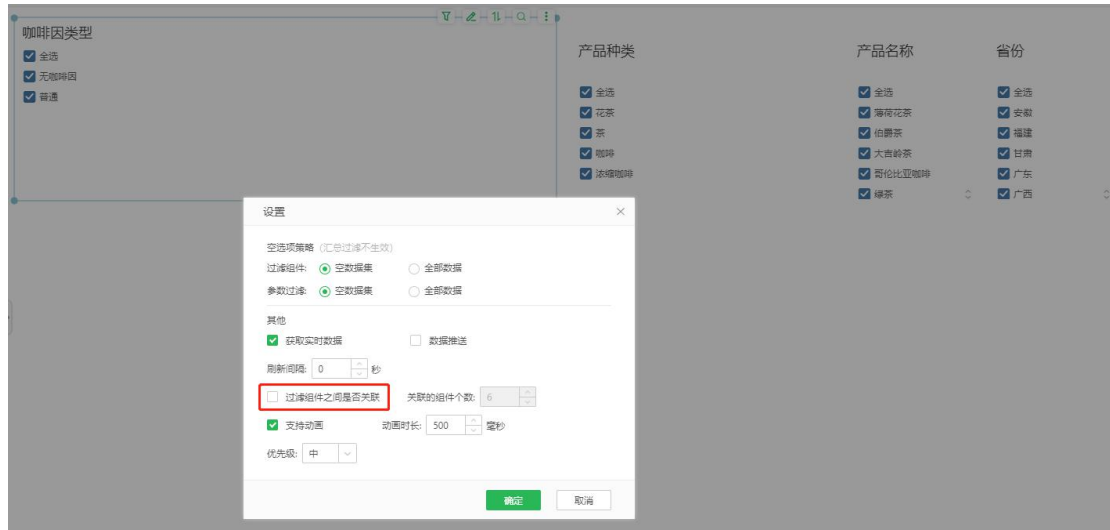
#### 5.1 过滤组件有依赖的情况，拼成一个大的 sql



实际执行的 sql:

```
SELECT `省份`, `产品名称`, `产品种类`, `咖啡因类型`  
FROM bella.coffeetest  
WHERE ((`省份` IS NOT NULL) OR (`产品名称` IS NOT NULL)) OR (`咖啡因类型` IS NOT NULL)  
OR (`产品种类` IS NOT NULL)  
GROUP BY `省份`, `产品名称`, `产品种类`, `咖啡因类型`  
ORDER BY `省份` ASC, `产品名称` ASC, `产品种类` ASC, `咖啡因类型` ASC  
LIMIT 500000  
//IS NOT NULL 是因为过滤组件默认忽略空值
```

## 5.2 过滤组件无依赖关系，每个组件单独执行一个 sql，并行



```
SELECT `省份`  
FROM bella.coffeetest  
WHERE `省份` IS NOT NULL  
GROUP BY `省份`  
ORDER BY `省份` ASC  
LIMIT 500000
```

```
SELECT `产品名称`  
FROM bella.coffeetest  
WHERE `产品名称` IS NOT NULL  
GROUP BY `产品名称`  
ORDER BY `产品名称` ASC  
LIMIT 500000
```

```
SELECT `咖啡因类型`  
FROM bella.coffeetest  
WHERE `咖啡因类型` IS NOT NULL  
GROUP BY `咖啡因类型`  
ORDER BY `咖啡因类型` ASC  
LIMIT 500000
```

```
SELECT `产品种类`  
FROM bella.coffeetest  
WHERE `产品种类` IS NOT NULL  
GROUP BY `产品种类`  
ORDER BY `产品种类` ASC
```

LIMIT 50000

## 5.3 参数组件之间本身无依赖，每个组件的 sql 是单独的，如果有依赖会单独更新依赖



```
[2021-06-21 22:54:47.840] -[INFO] -[56144231d6840c29808178eedb9747] [admin] [g5-ary.jdbc.DBCExecutor.g5RSU/DBCExecutor.java:1432] [DASHBOARD: {参数组件-2-20210621225447-336}[下拉列表参数1] g5-ary.sqlImpl.MySQLExecutor@31463016child-g5-ary.sqlImpl.MySQLExecutor@31463016] FROM baillia.coffeeest WHERE 市场分布 IS NOT NULL GROUP BY 市场分布 ORDER BY 市场分布 ASC LIMIT 50000
```

```
[2021-06-21 22:54:48.890] -[INFO] -[56144231d6840c29808178eedb9747] [admin] [g5-ary.GQuery.g5Grid/GQuery.java:1244] [DASHBOARD: {参数组件-2-20210621225447-336}[下拉列表参数1] Execute one query 'g5-ary.SQL.GQuery@4448684f; result 'g5.grid.SortGrid@786031a0', oc FROM baillia.coffeeest WHERE 市场分布 IS NOT NULL GROUP BY 市场分布 ORDER BY 市场分布 ASC LIMIT 50000]
```

```
[2021-06-21 22:54:48.940] -[INFO] -[56144231d6840c29808178eedb9747] [admin] [g5-ary.jdbc.DBCExecutor.g5RSU/DBCExecutor.java:1432] [DASHBOARD: {参数组件-2-20210621225447-336}[下拉列表参数2] g5-ary.sqlImpl.MySQLExecutor@2464e26bchild-g5-ary.sqlImpl.MySQLExecutor@2464e26b] SELECT 产品名称 FROM baillia.coffeeest WHERE ( 产品名称 IS NOT NULL) AND ( 市场分布 IN ( '东部市场', '南部市场' )) GROUP BY 产品名称 ASC LIMIT 50000
```

```
[2021-06-21 22:54:49.558] -[INFO] -[56144231d6840c29808178eedb9747] [admin] [g5-ary.GQuery.g5Grid/GQuery.java:1244] [DASHBOARD: {参数组件-2-20210621225447-336}[下拉列表参数2] Execute one query 'g5-ary.SQL.GQuery@348912ec; result 'g5.grid.SortGrid@1a5f1301', oc FROM baillia.coffeeest WHERE ( 产品名称 IS NOT NULL) AND ( 市场分布 IN ( ?, ? )) GROUP BY 产品名称 ASC LIMIT 50000]
```