Aiops 使用手册

一. 简介

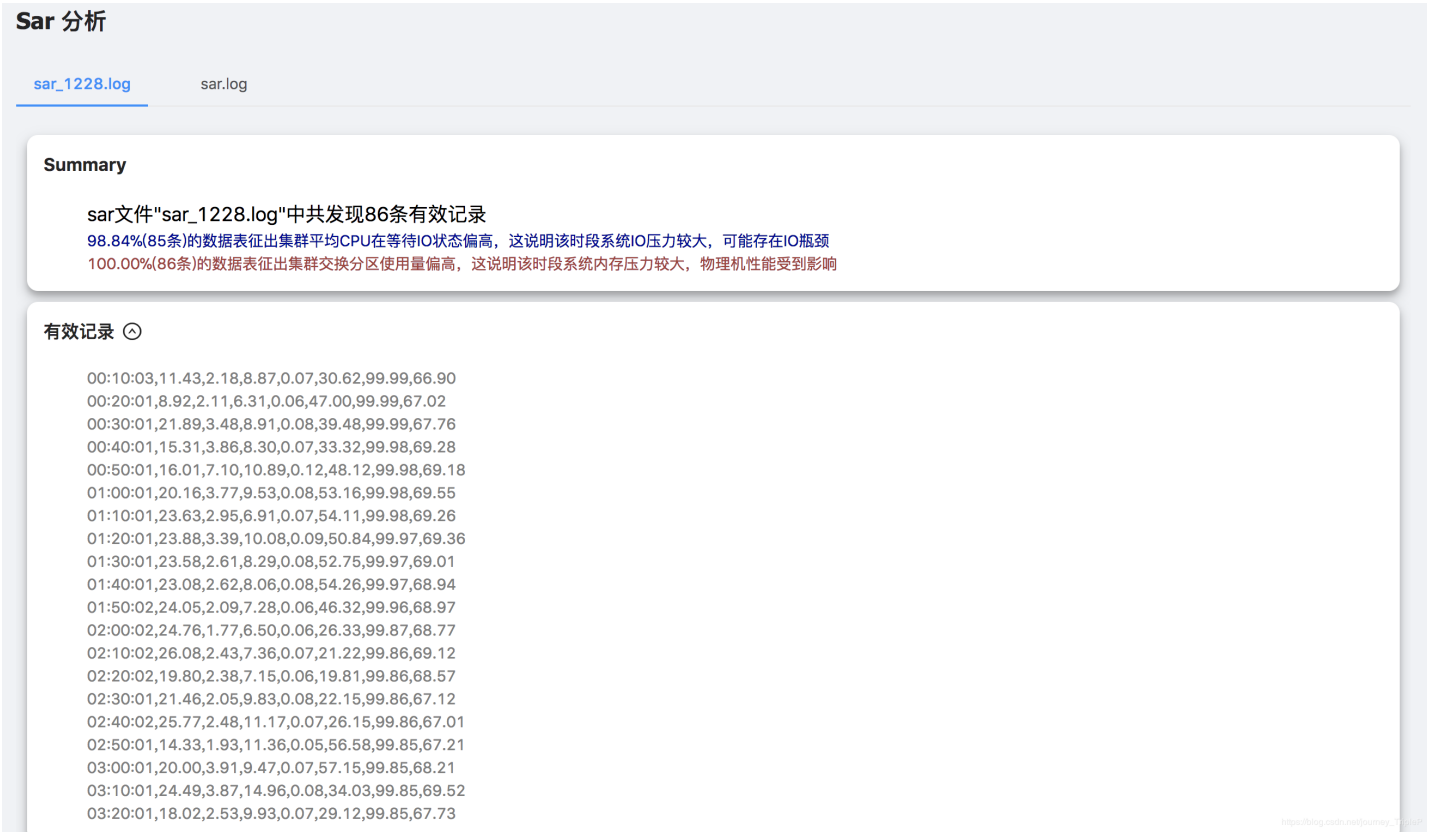
Aiops （智能运维） 是 ArgoDB （分布式闪存数据库） 的一个运维工具，致力于帮助客户更好地分析 Sar 、 Log 和 Jstack 等文件，快速定位问题，提高整个项目的健壮性。

二. Sar 分析

Sar（System Activity Reporter）：是监控Linux系统各个性能的优秀工具， 包括文件的读写情况、系统调用的使用情况、磁盘I/O、CPU效率、内存使用状况、进程活动及IPC有关的活动等。 在系统级诊断中经常发挥主导作用，在大数据平台级诊断中也起到重要的辅助作用。本系统会对Sar文件进行基本诊断。

获取Sar文件的方式："sar -A > sar.log"
获取sar历史的方式："sar -A -f /var/log/sa/sa? > sar.log"

Sar 分析内容如下图所示：



1. Summary

Summary 部分是对 sar 文件的一个分析总结：

- sar 文件中的有效记录数
- 系统可能存在的问题

2. 有效记录

有效记录显示 sar 文件中所有的有效记录

三. Log 分析

```
对日志文件进行分析，将一条 SQL 语句的执行的相关信息视为一个 Goal，并根据执行的不同阶段，
将一个 Goal 分成多个 Task，分析这些 Goal 和 Task 的运行状态和错误情况。
```

1. 统计信息

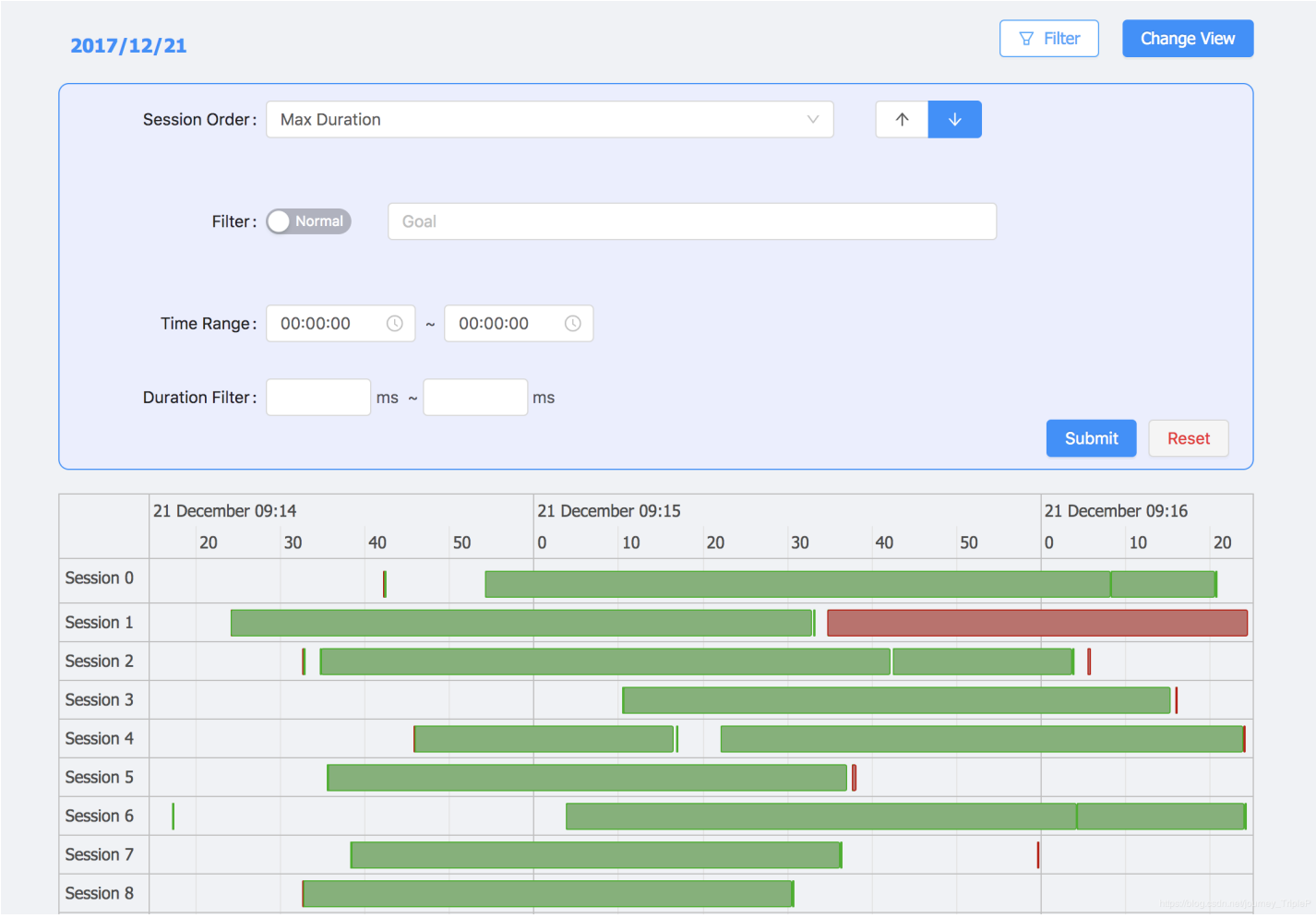
按照日期对日志文件的分析结果进行分类，并进行相关信息统计：

- Error SQL: 一共有多少条执行出错的 SQL 语句；
- Long Duration SQL: 一共有多少条执行时间过长的 SQL 语句；
- Normal SQL: 一共有多少执行正常的 SQL 语句。

2. SQL 展示

提供两种 SQL 展示视图： 时间轴视图和列表视图，用户可以点击 "Change View" 按钮在这两种视图间随意切换。

2.1 时间轴视图



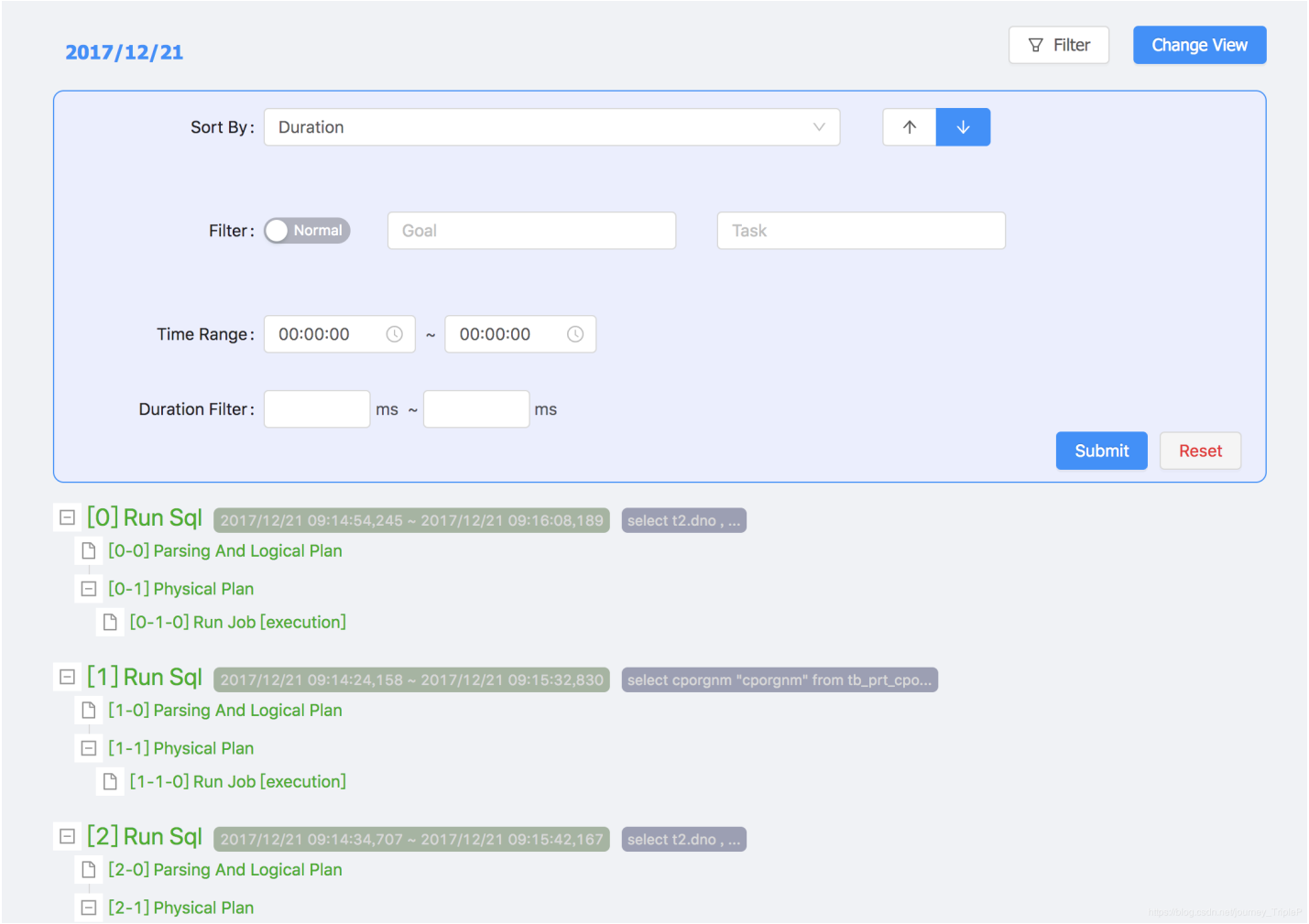
(1) 横轴表示执行 SQL 语句的时间， 纵轴是 Session，按照 Session 对 Goal（一条 SQL 语句的执行过程）进行分类。

(2) 点击时间轴中的任意一个 Goal 可以查看对应 SQL 语句的执行详情，包括执行时间，状态描述和前后 Goal 等信息。

(3) 过滤器 Filter:

- Session Order: 可以选择按照 Max Duration(最长执行时间）、Avg Duration(平均执行时间）、Exception Number(异常个数）对Sesiion 进行排序;
- Filter： 选择 Smart: 显示 Compile Error(编译错误)、Complete Error(完成但有错误)、Incomplete(未完成)的 Goal， Normal 情况下可以对 Compile Error(编译错误)、Complete Success(成功完成)、Complete Error(完成但有错误)、Incomplete(未完成) 中的一种或者多种 Goal 进行筛选；
- Time Range：设置起始时间和结束时间；
- Duration Filter： 设置最小和最大的 Goal 执行时间。

2.2 列表视图



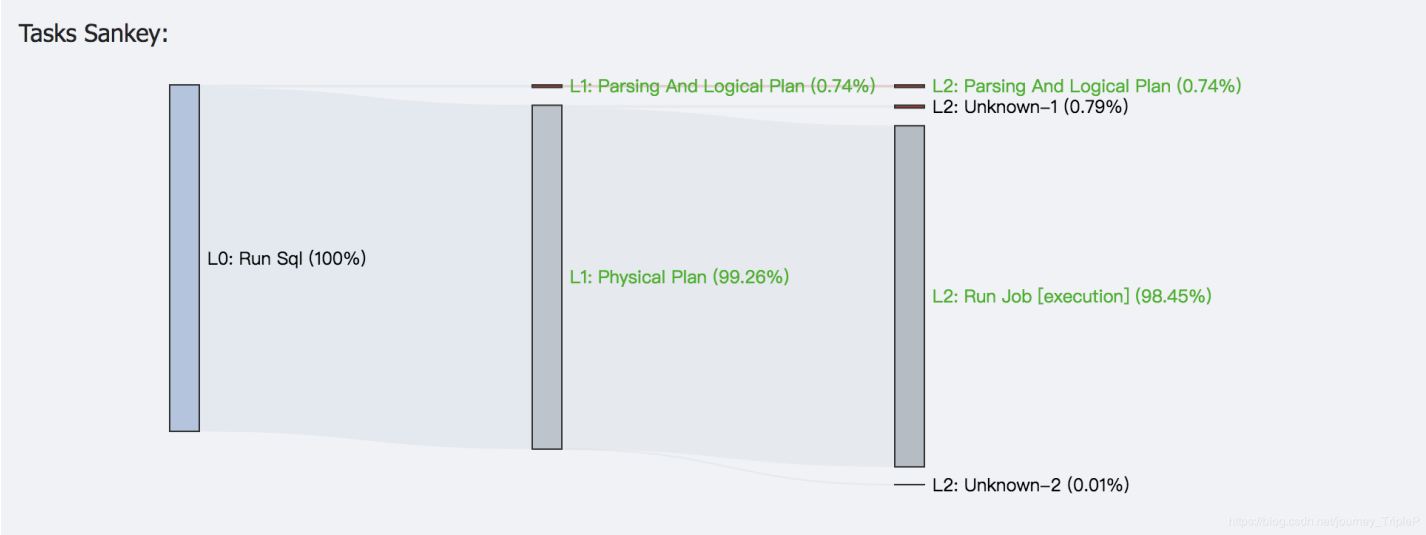
（1）按照层级显示 Goal 和 Task(根据 SQL 的执行过程将 Goal 分成多个 task)，点击可查看对应的 Goal/Task 的详细信息；

（2）过滤器 Filter:

- Sort By: 可以选择按照 Start Time(开始时间)、End Time(结束时间)、Duratio(执行时长)对 Goal 进行排序；
- Filter： 选择 Smart: 显示 Compile Error、Complete Error、Incomplete 的 Goal； Normal 情况下可以对不同执行状态的 Goal 和 Task 进行筛选
- Time Range：设置起始时间和结束时间；
- Duration Filter： 设置最小和最大的 Goal 执行时间。

3. Tasks Sankey

当点击某个 Goal/Task 可以查看详情，详情中的 Tasks Sankey 展示一个 Goal/Task 执行过程中各个部分的耗时以及中间未知的时间，如下图所示：

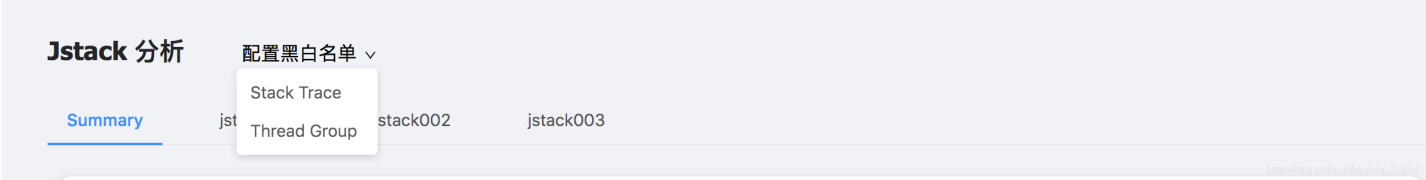


四. Jstack 分析

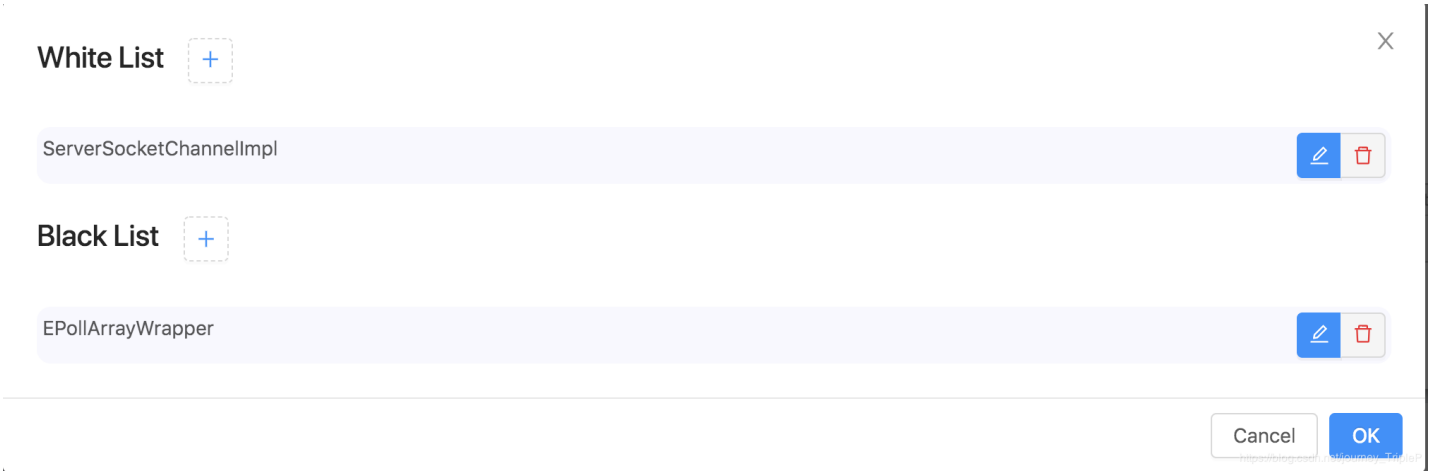
> "Jstack 分析" 是一个通用的 Java 线程堆栈分析器，为用户提供有用的信息报告。

1. 配置黑白名单

点击 “配置黑白名单”， 可以选择 “Stack Trace" 和 "Thread Group" 配置，如下图所示：



例如选择 “Stack Trace"，会弹出配置黑白名单的对话框如下，可以对黑白名单进行增删改等操作。



（1）Stack Trace 配置

“Stack Trace” 黑白名单配置功能和 “Identical Stack Trace" 模块结合使用，对线程 Stack Trace 进行过滤。

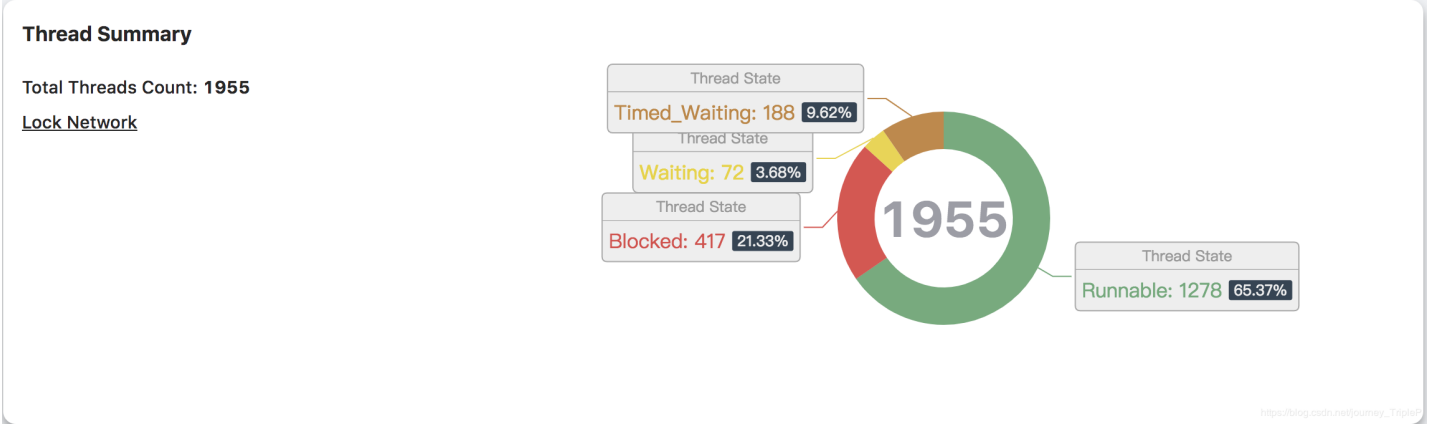
用户点击 “Stack Trace” 选项，可以设置白名单和黑名单。只要某个线程的 stack Trace 含有白名单中的关键字，这个线程就会被认为是很有用的，并且显示在 "Identical Stack Trace" 表格中的前面；只要某个线程的 stack Trace 含有黑名单中的关键字，这个线程就会被认为是无用的，并且不会出现在 “Identical Stack Trace" 的表格中。

（2）Thread Group 配置

“Thread Group” 黑白名单配置功能和 “Thread Group" 模块结合使用，对线程族进行过滤。

用户点击 “Thread Group” 选项，可以设置白名单和黑名单。只要某个线程族的线程名(Group Name)含有白名单中的关键字，这个线程族就会被认为是很有用的，并且显示在 "Thread Group" 表格中的前面；只要某个线程族的线程族名含有黑名单中的关键字，这个线程族就会被认为是无用的，并且不会出现在 “Thread Group" 的表格中。

2. Thread Summary



- Timestamp: Jstack 文件的显示时间；
- Total Threads Count：文件中一共有多少个线程；
- Lock Network: 点击显示线程间锁的占用情况；
- 饼状图：显示不同状态的线程的统计信息。

3. Identical Stack Trace

- 对具有相同调用栈的线程进行归类，点击链接可以查看详细信息；
- 可以通过页面上方的 “Stack Trace 配置” 按钮对 stack trace 进行黑白名单配置，在黑名单中的 "Identical Stack Trace" 将不再显示在表格中；
- 排序： 可以按照线程个数进行排序，白名单始终显示在前面；
- 过滤器：可以选择不同的线程状态。

4. Most Used Methods

Most Used Methods

Thread Count	Method
1023 threads	at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method)
187 threads	at org.apache.log4j.Category.callAppenders(Category.java:204)
173 threads	at org.apache.hadoop.hive.ql.exec.TransactionManagerHeartbeaterThread.stopHeartBeat(TransactionManagerHeartbeaterThread.java:66)
167 threads	at sun.misc.Unsafe.park(Native Method)
66 threads	at java.lang.Object.wait(Native Method)
57 threads	at sun.nio.ch.ServerSocketChannelImpl.accept(ServerSocketChannelImpl.java:225)
24 threads	at java.lang.Thread.sleep(Native Method)
9 threads	at java.net.PlainSocketImpl.socketAccept(Native Method)
3 threads	at sun.nio.ch.ServerSocketChannelImpl.accept0(Native Method)
3 threads	at java.net.SocketInputStream.socketRead0(Native Method)

< 1 2 > Goto

- 显示用的最多的方法，点击链接可以查看详情。

5. Thread Group

Thread Group

Group Name	Thread Count
shuffle-server-	480 threads
HiveServer2-Handler-Pool: Thread-	387 threads
I/O dispatcher	240 threads
elasticsearch[Indra][	204 threads
Gang worker#	201 threads
qtp1848848801-	173 threads
sparkDriver-akka.actor.default-dispatcher-	27 threads
qtp761879999-	24 threads
qtp404429505-	24 threads
New I/O worker #	12 threads

< 1 2 > Goto

- Thread Group(线程族): 按照线程名对线程进行分类；
- 可以通过页面上方的 “Thread Group 配置” 按钮对 "Group Name" 进行黑白名单配置，在黑名单中的 "Group Name" 将不再显示在表格中；
- 排序： 可以按照线程个数进行排序，白名单始终显示在前面。