

系统监控-线程

- 操作说明
 - 自动打印
 - 手动打印
 - 线程堆栈文件目录及获取
 - 清空线程堆栈

线程页面用于定时将JVM所有的线程堆栈打印到临时目录中，方便定位服务器运行缓慢、假死等情况。

服务器：

设置

打印间隔：

600

秒

停止打印

导出

清空

当前服务器

导出所有

最新

当前线程堆栈

概述

监视

线程

性能

网络

日志

会话

缓存

堆打印

对象池

类查找

扩展包

所有字体

wsdl地址

log4j信息

SQL监控

实验监控

计算节点

2019-10-12 09:21:05

Full thread dump OpenJDK 64-Bit Server VM (25.202-b05 mixed mode):

"smartbi_valid_thread_pool-117" #1210 prio=7 os_prio=1 tid=0x00000000354b1000 nid=0x25040 waiting on condition [0x00000000459ea000]
java.lang.Thread.State: TIMED_WAITING (parking)
at sun.misc.Unsafe.park(Native Method)
- parking to wait for <0x00000004d4004f8> (a java.util.concurrent.SynchronousQueue\$TransferStack)
at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
at java.util.concurrent.SynchronousQueue\$TransferStack.awaitFulfill(SynchronousQueue.java:460)
at java.util.concurrent.SynchronousQueue\$TransferStack.transfer(SynchronousQueue.java:362)
at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1073)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134)
at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)

Locked ownable synchronizers:
- None

"smartbi_valid_thread_pool-116" #1207 prio=7 os_prio=1 tid=0x00000000354af800 nid=0x24d3c waiting for monitor entry [0x00000000458a000]
java.lang.Thread.State: BLOCKED (on object monitor)
at com.sap.db.jdbc.ConnectionSapDB.tryReconnect(ConnectionSapDB.java:1298)
- waiting to lock <0x00000004cf019290> (a java.lang.String)
at com.sap.db.jdbc.ConnectionSapDB.execute(ConnectionSapDB.java:838)
at com.sap.db.jdbc.ConnectionSapDB.execute(ConnectionSapDB.java:602)
at com.sap.db.jdbc.StatementSapDB.sendCommand(StatementSapDB.java:696)
at com.sap.db.jdbc.StatementSapDB.sendSQL(StatementSapDB.java:945)
at com.sap.db.jdbc.StatementSapDB.execute(StatementSapDB.java:255)
at com.sap.db.jdbc.StatementSapDB.executeQuery(StatementSapDB.java:391)
at com.sap.db.jdbc.trace.Statement.executeQuery(Statement.java:131)
at org.apache.commons.dbcp.DelegatingStatement.executeQuery(DelegatingStatement.java:205)
at smartbi.connectionpool.ConnectionPool\$S\$2.call(ConnectionPool.java:778)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:624)
at java.lang.Thread.run(Thread.java:748)

Locked ownable synchronizers:
- <0x000000007b2c4f98> (a java.util.concurrent.ThreadPoolExecutor\$Worker)

操作说明

自动打印

Smartbi系统在启动时会自动打印线程堆栈，默认打印间隔为10分钟，也就每隔10分钟就会在线程打印目录下生成一个线程信息文件。生成一个新线程信息文件后，会自动将生成时间距现在已超过3天的旧线程信息文件删除掉，以免占用磁盘空间过多。

自动打印设置的启动位于 **系统运维 > 系统选项 > 公共设置** 中，可设置系统自动打印。

如需再次启动自动打印，请在设置为是后重启服务器，并在线程界面确认是否已开始自动打印。

系统选项

公共设置 用户管理设置 查询设置 灵活分析即席查询设置 多维分析设置 页面设置 移动端 缓存设置 电子表格设置 透视分析设置 分析报告设置 机器学习配置

公共设置

起始模块设置：

导航页

系统默认

自定义

数据门户首页设置：

隐藏

显示

标题：首页

内容：openresource.jsp?resid=140280981016aa032a

选择内部资源

隐藏发布目录中“我的空间”模块：

是

否

初始值(否)

恢复初始值

页面、灵活分析功能：

启用

禁用

初始值(禁用)

恢复初始值

业务主题表关系：

全局

局部

初始值(局部)

恢复初始值

操作日志设置：

设置

添加表时使用注释作为别名：

是

否

初始值(是)

恢复初始值

单个跳转目标是否弹出菜单：

是

否

初始值(否)

恢复初始值

系统启动后自动定时导出线程信息：

是

否

初始值(是)

恢复初始值

系统启动后自动定时导出性能监控信息：

是

否

初始值(是)

恢复初始值

ZIP编码设置：

GBK

初始值(GBK)

恢复初始值

单个一级TAB页是否显示：

是

否

初始值(否)

恢复初始值

Excel2007最大导出行数：

20000

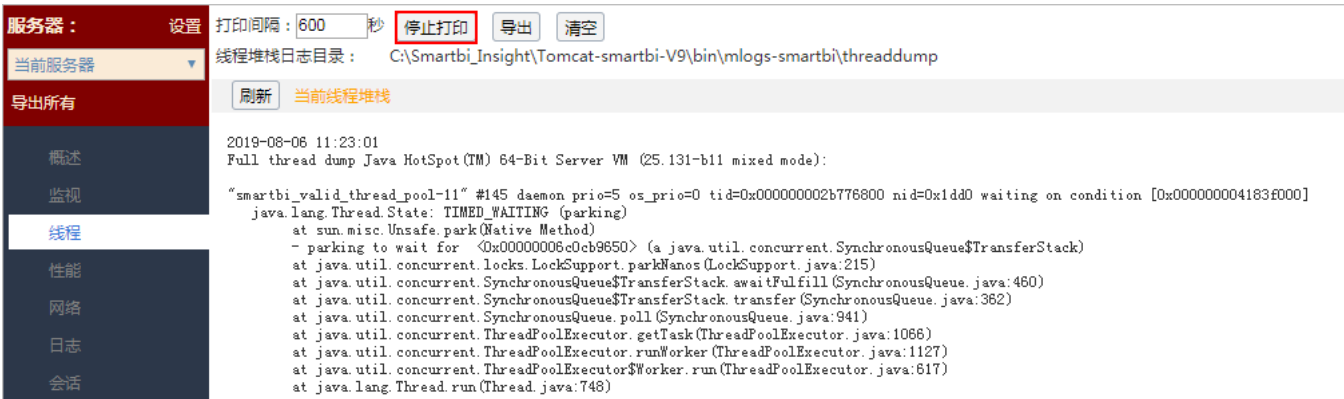
初始值(20000)

恢复初始值

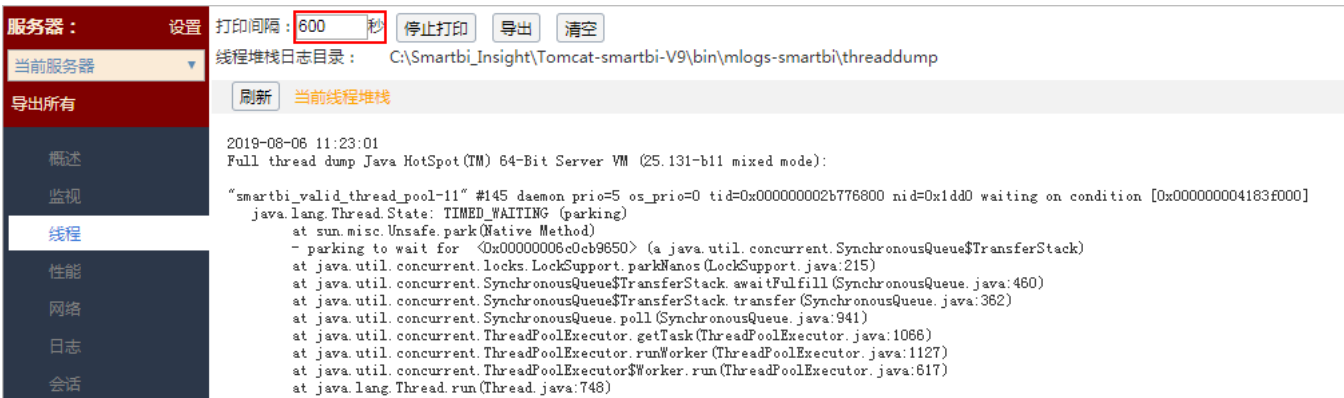
手动打印

可在需要时手工执行打印线程堆栈。

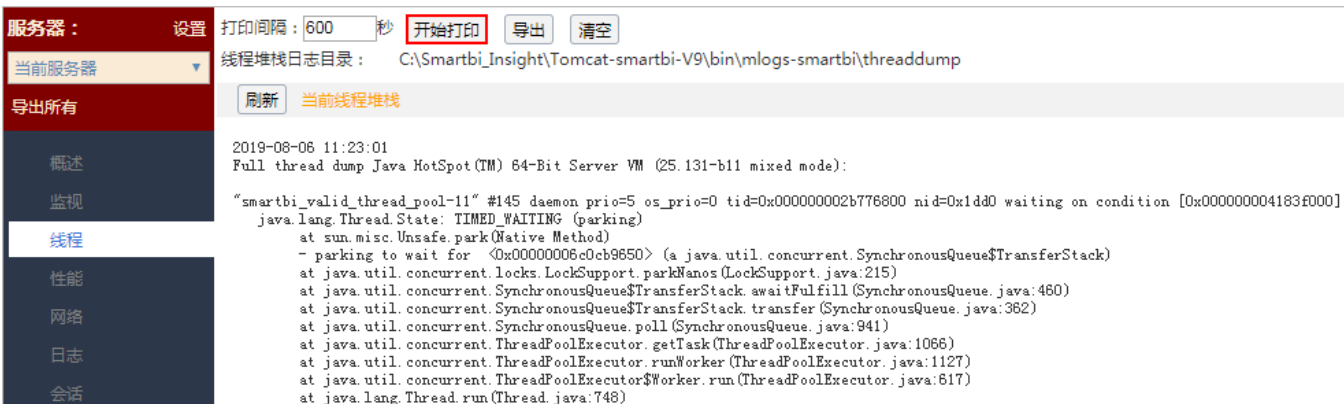
(1) 进入线程界面，先检查是否存在“停止打印”按钮，若存在，则点击“停止打印”按钮停止系统的自动打印；若不存在“停止打印”按钮，只有“开始打印”按钮，才可以进行下一步。



(2) 在“打印间隔”中，输入需要的打印间隔时间。



(3) 点击“开始打印”按钮，系统会按照设定的打印间隔时间自动打印线程堆栈。



线程堆栈文件目录及获取

打印生成的线程堆栈文件存放于“线程”界面显示的目录中。

服务器：设置

打印间隔：600秒

开始打印

导出

清空

当前服务器

导出所有

概述

监视

线程

性能

网络

日志

会话

线程堆栈日志目录：C:\Smartbi_Insight\Tomcat-smartbi-V9\bin\mlogs-smartbi\threaddump

刷新

当前线程堆栈

2019-08-06 11:23:01

Full thread dump Java HotSpot(TM) 64-Bit Server VM (25.131-b11 mixed mode):

"smartbi_valid_thread_pool-11" #145 daemon prio=5 os_prio=0 tid=0x000000002b776800 nid=0x1dd0 waiting on condition [0x000000004183f000]
java.lang.Thread.State: TIMED_WAITING (parking)
at sun.misc.Unsafe.park(Native Method)
- parking to wait for <0x000000006c0cb9650> (a java.util.concurrent.SynchronousQueue\$TransferStack)
at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
at java.util.concurrent.SynchronousQueue\$TransferStack.awaitFulfill(SynchronousQueue.java:460)
at java.util.concurrent.SynchronousQueue\$TransferStack.transfer(SynchronousQueue.java:362)
at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1066)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1127)
at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:748)

获取线程堆栈文件的方式有以下几种：

(1) 点击“线程”界面面上的“导出”按钮，会将线程堆栈目录下的所有线程堆栈文件以压缩包的形式导出。一般推荐这种方式。

服务器：设置

打印间隔：600秒

开始打印

导出

清空

当前服务器

导出所有

概述

监视

线程

性能

网络

日志

会话

线程堆栈日志目录：C:\Smartbi_Insight\Tomcat-smartbi-V9\bin\mlogs-smartbi\threaddump

刷新

当前线程堆栈

2019-08-06 11:23:01

Full thread dump Java HotSpot(TM) 64-Bit Server VM (25.131-b11 mixed mode):

"smartbi_valid_thread_pool-11" #145 daemon prio=5 os_prio=0 tid=0x000000002b776800 nid=0x1dd0 waiting on condition [0x000000004183f000]
java.lang.Thread.State: TIMED_WAITING (parking)
at sun.misc.Unsafe.park(Native Method)
- parking to wait for <0x000000006c0cb9650> (a java.util.concurrent.SynchronousQueue\$TransferStack)
at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
at java.util.concurrent.SynchronousQueue\$TransferStack.awaitFulfill(SynchronousQueue.java:460)
at java.util.concurrent.SynchronousQueue\$TransferStack.transfer(SynchronousQueue.java:362)
at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1066)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1127)
at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:748)

(2) 访问系统中的系统运维，点击“导出日志”按钮，将日志文件导出。在导出的压缩包中，有如下以“ThreadDumps”开头的压缩包

名称	修改日期	类型	大小
extensionsInfo.xml	2017/7/20 9:56	XML 文档	7 KB
HeapDump.log	2017/7/20 9:56	文本文档	410 KB
pools.txt	2017/7/20 9:56	文本文档	2 KB
sessionsInfo.xml	2017/7/20 9:56	XML 文档	8 KB
smartbi.log	2017/7/20 9:56	文本文档	954 KB
smartbi.log.1	2017/7/19 18:46	1 文件	10,248 KB
smartbi.log.2	2017/7/18 10:44	2 文件	10,630 KB
smartbi.log.3	2017/7/18 10:20	3 文件	10,506 KB
smartbi.log.4	2017/7/14 23:40	4 文件	10,245 KB
Smartbi.SystemProperties.log	2017/7/20 9:56	文本文档	5 KB
smartbi-config.xml	2017/7/20 9:56	XML 文档	3 KB
ThreadDump.txt	2017/7/20 9:56	文本文档	83 KB
ThreadDumps3820344039306787962.zip	2017/7/20 9:56	WinRAR ZIP 压缩...	1,939 KB
trace.xml	2017/7/20 9:56	XML 文档	1 KB
versions.txt	2017/7/20 9:56	文本文档	2 KB

该压缩包中包含系统保存的所有打印的线程堆栈文件。

(3) 若系统已经无法访问，并且需要线程堆栈文件，则可以直接进入上图中的线程堆栈目录下，即可拿到打印的线程堆栈文件。

计算机 > 本地磁盘 (D:) > Smartbi > tomcat_v7_smartbi_17000 > bin > mlogs-smartbi > threadump				
包含到库中 ▾ 共享 ▾ 新建文件夹				
名称	修改日期 ▾	类型	大小	
 2017-07-20 09_42_52.txt	2017/7/20 9:42	文本文档	55 KB	
 2017-07-20 09_40_29.txt	2017/7/20 9:40	文本文档	55 KB	
 2017-07-20 09_30_29.txt	2017/7/20 9:30	文本文档	53 KB	
 2017-07-20 09_20_29.txt	2017/7/20 9:20	文本文档	21 KB	
 2017-07-20 05_38_19.txt	2017/7/20 5:38	文本文档	56 KB	
 2017-07-20 05_28_19.txt	2017/7/20 5:28	文本文档	56 KB	
 2017-07-20 05_18_19.txt	2017/7/20 5:18	文本文档	56 KB	
 2017-07-20 05_08_19.txt	2017/7/20 5:08	文本文档	56 KB	
 2017-07-20 04_58_19.txt	2017/7/20 4:58	文本文档	55 KB	
 2017-07-20 04_48_19.txt	2017/7/20 4:48	文本文档	55 KB	
 2017-07-20 04_38_19.txt	2017/7/20 4:38	文本文档	55 KB	
 2017-07-20 04_28_19.txt	2017/7/20 4:28	文本文档	55 KB	
 2017-07-20 04_18_19.txt	2017/7/20 4:18	文本文档	55 KB	

清空线程堆栈

点击“线程”界面上的“清空”按钮，可清空线程堆栈目录下的所有线程堆栈文件。

服务器： 设置

当前服务器 ▾

导出所有

概述

监视

线程

性能

网络

日志

会话

打印间隔：600 秒 开始打印 导出 清空

线程堆栈日志目录： C:\Smartbi_Insight\Tomcat-smartbi-V9\bin\mlogs-smartbi\threadump

刷新 当前线程堆栈

2019-08-06 11:23:01

Full thread dump Java HotSpot(TM) 64-Bit Server VM (25.131-b11 mixed mode):

"smartbi_valid_thread_pool-11" #145 daemon prio=5 os_prio=0 tid=0x000000002b776800 nid=0x1dd0 waiting on condition [0x000000004183f000]
java.lang.Thread.State: TIMED_WAITING (parking)
at sun.misc.Unsafe.park(Native Method)
- parking to wait for <0x000000006c0cb9650> (a java.util.concurrent.SynchronousQueue\$TransferStack)
at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
at java.util.concurrent.SynchronousQueue\$TransferStack.awaitFulfill(SynchronousQueue.java:460)
at java.util.concurrent.SynchronousQueue\$TransferStack.transfer(SynchronousQueue.java:362)
at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1066)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1127)
at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:748)