

# Skywalking-04：扩展Metric监控信息

 Switch

发布于 7 月 28 日

## 扩展 Metric 监控信息

### 官方文档

[Source and Scope extension for new metrics](#)

### 案例：JVM Thread 增加 Metrics

#### 修改 Thread 的定义

在 `apm-protocol/apm-network/src/main/proto/language-agent/JVMMetric.proto` 协议文件中覆盖 `message Thread` 的定义

```
message Thread {
  int64 liveCount = 1;
  int64 daemonCount = 2;
  int64 peakCount = 3;
  int64 deadlocked = 4;
  int64 monitorDeadlocked = 5;
  int64 newThreadCount = 7;
  int64 runnableThreadCount = 8;
  int64 blockedThreadCount = 9;
  int64 waitThreadCount = 10;
  int64 timeWaitThreadCount = 11;
  int64 terminatedThreadCount = 12;
}
```

重新构建 `apm-network` 项目

```
cd apm-protocol/apm-network
mvn clean package -DskipTests=true
```

PS：可以安装 `Protocol Buffer Editor` 插件，支持 `Protocol Buffer` 语法

#### 修改 agent core 中 Thread Metrics 的提供类

直接使用如下代码覆盖 `org.apache.skywalking.apm.agent.core.jvm.thread.ThreadProvider` 类

```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

```
package org.apache.skywalking.apm.agent.core.jvm.thread;

import java.lang.management.ManagementFactory;
import java.lang.management.ThreadInfo;
import java.lang.management.ThreadMXBean;
import java.util.Optional;

import org.apache.skywalking.apm.network.language.agent.v3.Thread;
```

## 修改 ServiceInstanceJVMThread

直接使用如下代码覆盖 `org.apache.skywalking.oap.server.core.source.ServiceInstanceJVMThread` 类，`ServiceInstanceJVMThread`继承了 `Source` 抽象类，`Source` 类是 Skywalking 中 [oal](#) 体系，资源及范围的定义。

```
/*
 * Licensed to the Apache Software Foundation (ASF) under one or more
 * contributor license agreements. See the NOTICE file distributed with
 * this work for additional information regarding copyright ownership.
 * The ASF licenses this file to You under the Apache License, Version 2.0
 * (the "License"); you may not use this file except in compliance with
 * the License. You may obtain a copy of the License at
 *
 *      http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package org.apache.skywalking.oap.server.core.source;

import lombok.Getter;
import lombok.Setter;

import static org.apache.skywalking.oap.server.core.source.DefaultScopeDefine.SERVICE_INSTANCE_CATALOG_NAME;
import static org.apache.skywalking.oap.server.core.source.DefaultScopeDefine.SERVICE_INSTANCE_JVM_THREAD;
```

## 修改 JVMSourceDispatcher

`org.apache.skywalking.oap.server.analyzer.provider.jvm.JVMSourceDispatcher` 是一个 `Source` 分发类，将从 `agent` 接收到的 `JVM` 相关 `Metrics` 拆分成对应的 `Source`。例如：`ServiceInstanceJVMMemory`、`ServiceInstanceJVMThread`。

修改方法`org.apache.skywalking.oap.server.analyzer.provider.jvm.JVMSourceDispatcher#sendToThreadMetricProcess` 为如下代码：

```
private void sendToThreadMetricProcess(String service,
    String serviceId,
    String serviceInstance,
    String serviceInstanceId,
    long timeBucket,
    Thread thread) {
    ServiceInstanceJVMThread serviceInstanceJVMThread = new ServiceInstanceJVMThread();
    serviceInstanceJVMThread.setId(serviceInstanceId);
    serviceInstanceJVMThread.setName(serviceInstance);
    serviceInstanceJVMThread.setServiceId(serviceId);
    serviceInstanceJVMThread.setServiceName(service);
    serviceInstanceJVMThread.setLiveCount(thread.getLiveCount());
    serviceInstanceJVMThread.setDaemonCount(thread.getDaemonCount());
    serviceInstanceJVMThread.setPeakCount(thread.getPeakCount());
    serviceInstanceJVMThread.setTimeBucket(timeBucket);
    serviceInstanceJVMThread.setDeadlocked(thread.getDeadlocked());
    serviceInstanceJVMThread.setMonitorDeadlocked(thread.getMonitorDeadlocked());
    serviceInstanceJVMThread.setNewThreadCount(thread.getNewThreadCount());
    serviceInstanceJVMThread.setRunnableThreadCount(thread.getRunnableThreadCount());
    serviceInstanceJVMThread.setBlockedThreadCount(thread.getBlockedThreadCount());
    serviceInstanceJVMThread.setWaitThreadCount(thread.getWaitThreadCount());
    serviceInstanceJVMThread.setTimeWaitThreadCount(thread.getTimeWaitThreadCount());
    serviceInstanceJVMThread.setTerminatedThreadCount(thread.getTerminatedThreadCount());
    sourceReceiver.receive(serviceInstanceJVMThread);
}
```

## 在 java-agent.oal 增加相关指标

在oap-server/server-bootstrap/src/main/resources/oal/java-agent.oal 添加如下语句

```
// 参考oal语法
instance_jvm_thread_deadlocked = from(ServiceInstanceJVMThread.deadlocked).longAvg();
instance_jvm_thread_monitor_deadlocked = from(ServiceInstanceJVMThread.monitorDeadlocked).longAvg();
instance_jvm_thread_new_thread_count = from(ServiceInstanceJVMThread.newThreadCount).longAvg();
instance_jvm_thread_runnable_thread_count = from(ServiceInstanceJVMThread.runnableThreadCount).longAvg();
instance_jvm_thread_blocked_thread_count = from(ServiceInstanceJVMThread.blockedThreadCount).longAvg();
instance_jvm_thread_wait_thread_count = from(ServiceInstanceJVMThread.waitThreadCount).longAvg();
instance_jvm_thread_time_wait_thread_count = from(ServiceInstanceJVMThread.timeWaitThreadCount).longAvg();
instance_jvm_thread_terminated_thread_count = from(ServiceInstanceJVMThread.terminatedThreadCount).longAvg();
```

## 修改 apm.yml

在 oap-server/server-bootstrap/src/main/resources/ui-initialized-templates/apm.yml 文件的 APM 面板下的 Instance 项增加如下配置

```
{
  "width": 3,
  "title": "JVM Thread Count (Java Service)",
  "height": "250",
  "entityType": "ServiceInstance",
  "independentSelector": false,
  "metricType": "REGULAR_VALUE",
  "queryMetricType": "readMetricsValues",
  "chartType": "ChartLine",
  "metricName": "instance_jvm_thread_live_count, instance_jvm_thread_daemon_count,
instance_jvm_thread_peak_count,instance_jvm_thread_deadlocked,instance_jvm_thread_monitor_deadlocked"
},
{
  "width": 3,
  "title": "JVM Thread State Count (Java Service)",
  "height": "250",
  "entityType": "ServiceInstance",
  "independentSelector": false,
  "metricType": "REGULAR_VALUE",
  "metricName":
"instance_jvm_thread_new_thread_count,instance_jvm_thread_runnable_thread_count,instance_jvm_thread_blocked_threa
d_count,instance_jvm_thread_wait_thread_count,instance_jvm_thread_time_wait_thread_count,instance_jvm_thread_term
inated_thread_count",
  "queryMetricType": "readMetricsValues",
  "chartType": "ChartBar"
}
```

如果不清楚添加位置，可以直接使用如下配置，覆盖 `oap-server/server-bootstrap/src/main/resources/ui-initialized-templates/apm.yml`

```
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# UI templates initialized file includes the default template when the SkyWalking OAP starts up at the first
# time.
#
# Also, SkyWalking would detect the existing templates in the database, once they are missing, all templates in
# this file
# could be added automatically.

templates:
  - name: "APM"
    # The type includes DASHBOARD, TOPOLOGY_INSTANCE, TOPOLOGY_ENDPOINT.
```

## 效果展示

## 代码贡献

- [Add some new thread metric and class metric to JVMMetric #7230](#)

- [add some new thread metric and class metric to JVMMetric #52](#)
- [Remove Terminated State and New State in JVMMetric \(#7230\) #53](#)
- [Add some new thread metric and class metric to JVMMetric \(#7230\) #7243](#)

## 总结

Metric 如何扩展，网上基本上没案例，都是看官方文档和源码了解。对应这种非常热门的开源项目，还是看官方文档和源码更稳。

## 参考文档

- [Java ManagementFactory解析](#)
- [编程中使用ThreadMXBean类来检测死锁](#)
- [Source and Scope extension for new metrics](#)
- [Observability Analysis Language](#)

分享并记录所学所见



阅读 169 • 发布于 7 月 28 日

👍 赞

🔖 收藏

🔗 分享

本作品系原创，采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



Switch

9 声望   0 粉丝

关注作者

0 条评论

得票数 | 最新



撰写评论 ...

提交评论

你知道吗？

不要站着调试程序，那会使得你的耐心减半，你需要的是全神贯注。

注册登录

继续阅读

### JVM监控命令详解

JVM监控命令基本就是 jps、jstack、jmap、jhat、jstat 几个命令的使用就可以了 JDK本身提供了很多方便的JVM性能调优监控工具...  
[alterem](#) · 阅读 1.9k · 5 赞

### dubbo扩展点机制

spring是如何启动容器的 常见的一种在本地使用main方法启动spring的方法 {代码...} dubbo是如何启动容器的 这个大家应该都知道...  
[包月星](#) · 阅读 2.7k · 1 赞

### Django搭建个人博客：扩展用户信息

但是自带的User毕竟可用的字段较少。比方说非常重要的电话号码、头像等都没有。解决的方法有很多，你可以不使用User，自己...  
[杜赛](#) · 阅读 1.8k · 1 赞 · 10 评论

### Prometheus + Granafa 构建高大上的MySQL监控平台

对于MySQL的监控平台，相信大家实现起来有很多了：基于天兔的监控，还有基于zabbix相关的二次开发。相信很多同行都应该已...  
[架构文摘](#) · 阅读 2.1k · 1 赞

### 使用腾讯云“自定义监控”监控GPU使用率

随着人工智能以及比特币的火热，GPU云服务的使用场景是越来越广，在很多场景下我们也需要获取GPU服务器的性能参数来优化...  
[腾讯云加社区](#) · 阅读 3.2k

### Grafana 采集阿里云SLB监控信息

阿里云监控为云上用户提供常用云产品的监控数据和用户自定义上报的监控数据。在可视化展示层面，除了在云监控控制台查看监...  
[深蓝的海](#) · 阅读 2.2k · 2 评论

### 监控招标信息

\_\_\_\_为第一时间获取招标信息，需要不断访问各个招标网站，然而网络招标信息分布在公共资源交易网站、行业交易平台网站和专...  
[木头](#) · 阅读 177

### mysqld\_exporter监控mysql信息

一、背景使用 mysqld\_exporter 来抓取 mysql的一些指标信息。二、prometheus接入mysqld\_exporter1、安装mysqld\_exporter {代...  
[huan1993](#) · 阅读 757

产品

热门问答

热门专栏

热门课程

最新活动

技术圈

酷工作

课程

Java 开发课程

PHP 开发课程

Python 开发课程

前端开发课程

移动开发课程

资源

每周精选

用户排行榜

勋章

帮助中心

声望与权限

社区服务中心

建议反馈

合作

关于我们

广告投放

职位发布

讲师招募

联系我们

合作伙伴

关注

产品技术日志

社区运营日志

市场运营日志

团队日志

社区访谈

条款

服务协议

隐私政策

下载 App