

文章

[Claire Zheng](#) · 一月 20, 2021



阅读大约需分钟

使用内置的REST API监控InterSystems IRIS

我们不必等待SAM发布才开始规划和试用该API来监控IRIS实例。在以后文章中，我将更深入地探讨可用的指标及其意义，并提供一些交互式仪表板的示例。就，我将介绍一些相关背景和一些问题及答案。

IRIS (和Cache)一直在收集自身及其运行平台的数十个指标。收集这些指标来监控Cache和IRIS的方法向来有很多。我发现，很少有安装软件使用IRIS和Cache的内置解决方案。譬如，History Monitor作为Cache系统使用指标的历史数据库，已经推出很长时间了，但它没有简便法可实时显示这些指标和仪表系统。

IRIS平台解决方案(以及整个业界)正在从仅在一些本地实例上运行的单体应用程序过渡到“随处”部署的分布式解决方案。在许多用例中，原有的IRIS监控方案并不适用于这些新的模式。InterSystems没有做重复工作，而是将目光投向当前流行的、经过验证的监控和告警开源解决方案。

Prometheus?

Prometheus是一个基础广泛、应用广泛的著名开源监控系统，具有各种各样的插件。Prometheus的设计目的是在云环境中很好地运行，但它同样适用于本地环境。Prometheus的插件包括操作系统、web服务器(如Apache)和许多其他的应用程序。Prometheus通常与前端客户端(如Grafana)一起使用，后者提供极具定制性一流UI/UX体验。

Grafana?

Grafana也是开源平台。随着本系列文章的深入，我将提供常见场景监控仪表板的示例模板。在这些示例模板的基础上，你可以进一步设计自己想要的仪表板。将IRIS指标与你的整个解决方案堆栈的指标适当结合，就可以发挥真正的威力；指标来自于平台组件、操作系统、IRIS，尤其是当你在应用程序中添加仪表功能的时候。

我以前见过这种方法吗？

使用Prometheus和Grafana监控IRIS与Cache并不是新出现的方法。几年来，我一直使用这些应用程序监控开发和测试系统。在开发者社区中搜索“Prometheus”，就可以找到一些其它文章，这些文章讲述了如何显示Cache指标，以供Prometheus使用。

跟过去不同的是，现在的/api/monitor API是内置的，而且默认启用。用户无需显示指标而自己编写类。

Prometheus入门

在这里简单介绍一些Prometheus和一些术语。阅读本文能够深入学习并打一定基础，这将有助于你思考如何运用IRIS或其他来源提供指标或将其可视化。

Prometheus的工作方式是抓取或提取由作为HTTP端点的应用程序(IRIS /api/monitor等API)所公开的时间序列数据。现有的导出器和客户端库适用于多种语言、框架、开源应用程序(例如Apache等web服务器)、操作系统、docker、Kubernetes、数据库和现在的IRIS。

导出器用于将应用程序和服务仪表板，并公开端点上的相关指标以进行抓取。核心导出器支持各种标准组件，例如w

eb服务器、数据库等。其多种开源导出器可从Prometheus社区获取。

Prometheus术语

有几个关键术语需要：

- **目标**是指你关心的服务所处位置，例如主机、应用程序或服务(如：Apache、IRIS或你自己的应用程序)。
- Prometheus通过HTTP抓取目标，以收集指标作为时间序列数据。
- **时间序列数据**是通过应用程序(如IRIS)或导出器公开的。
- **导出器**用于你无法控制的数据，譬如Linux内核指标。
- **生成时间序列数据** 存储在本地Prometheus服务器上的一个数据库中**。
- **这个时间序列数据库** 可以使用优化的查询语言(PromQL)进行查询，以便建警报或由客户端应用程序(如Grafana)在仪表板中显示指标等。

**剧透警告：

出于安全性、扩展性、高可用性、运营效率方面的考虑，在新的SAM解决方案中，Prometheus时间序列数据所使用的数据库是IRIS数据库！但对IRIS平台的Prometheus数据库的访问是透明的，而应用程序(如Grafana)对这些访问既不知道也不关心。

Prometheus数据模型

该API返回的指标是Prometheus格式的。Prometheus采用基本的简单指标格式，每行一个指标，格式如：

```
[ (timen, valuen), ....]
```

指标可将标签作为(键,值)对，而标签是将指标作为维度过滤的强大方法。例如，检验IRIS/api/monitor返回的一个指标 日志存储空间：

```
iris_jrn_free_space{id="WIJ",dir="/fast/wij/"} 401562.83
```

标识符反映了指标的含义和来源：

```
iris_jrn_free_space
```

每个标签可用于修饰指标，然后用来过滤和查询。在这个例子中，可以翻WIJ及其存储目录：

```
id="WIJ",dir="/fast/wij/"
```

此外，还有一个值：401562.83(MB)。

有哪些IRIS指标可供使用？

预览版文档资料中含有一个

指标列表。但要注意，其中的内容可能会有所调整。你

也可以 [查询/api/monitor/metrics](#)端点以查列表。我使用的是

[Postman](#)

,关于这一点,我将在另一篇文章中进行阐述。

我应该监控什么?

考虑怎样监控你的系统和应用程序时,请记住要点:

1. 尽可能将影响用户的关键指标仪表板

- o 用户并不关心你的某台机器是否缺少CPU。
- o 用户关心的服务是否缓慢或出错。
- o 你的主仪表板应显示那些直接影响用户的高级指标。

2. 仪表板应避免显示大量图表

- o 人无法一次性处理数据。
- o 例如:每项服务都有一个仪表板。

3. 关注服务,而不是机器

- o 将问题定位到某项服务之后,你就可以接着往找原因,是否是哪台机器出了问题。

参考资料

文档资料 and 载

Prometheus: <https://prometheus.io/>

Grafana: <https://grafana.com/>

在InterSystems 2019年全球峰会上,我简要介绍了预发布版SAM(包括Prometheus和Grafana);你可以在InterSystems学习服务网站上找到 [相关链接](#)。如果此链接无效,请访问 [InterSystems学习服务网站](#),并搜索:“System Alerting and Monitoring Made Easy”。

你也可以在社区中搜索“Prometheus”和“Grafana”。

注:本文为译文,欢迎 [点击阅读原文](#),原文由 [Murray Oldfield](#)撰写

[#API](#) [#InterSystems 业务解决方案和架构](#) [#监视](#) [#开源](#) [#仪表板](#) [#开发运维](#) [#最佳做法](#) [#系统管理](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

源 URL: <https://cn.community.intersystems.com/post/%E4%BD%BF%E7%94%A8%E5%86%85%E7%BD%AE%E7%9A%84rest-api%E7%9B%91%E6%8E%A7intersystems-iris>