文章
Claire Zheng · 一月 20, 2021 阅读大约需 15 分钟

Open Exchange

InterSystems IRIS数据平台：数据接收速度测试

跨行业用例大多要求具备每秒接收数千或数百万条记录的能力，同时能够支持实时同步查询，例如：股票交易处理、欺诈检测、物联网应用（包括异常检测和实时OEE监控）等。Gartner将这种能力称为“ HTAP ”（混合事务分析处理）。Forrester等其他公司将其称为Translytics。InterSystems IRIS
是功能强大、可扩展、高性能、资源高效的事务分析型数据平台，同时具备内存数据库的高性能以及传统数据库的一致性、可用性、可靠性以及低成本的特性。

混合事务分析处理（HTAP）示例

此示例展示了InterSystems IRIS如何实现每秒接收数千条记录，同时允许对同一集群上的数据进行同步查询，该平台不仅具有很高的接收和查询性能，而且保持了较低的资源利用率。此示例可在单个InterSystems
IRIS实例或云端InterSystems IRIS集群上运行。

大家也可以在SAP HANA、MySQL、SqlServer及Amazon
Aurora上运行这个示例，以便对性能和资源利用率进行公平、合理的对比。

大家可以在AWS上运行该测试！以下是部分结果：

- **在AWS上运行InterSystems IRIS和SAP HANA：**

o在接收记录量方面，InterSystems IRIS比SAP HANA多39%

o在查询速度方面，InterSystems IRIS比SAP HANA快3699%

- **在AWS上运行InterSystems IRIS和AWS Aurora（MySQL）：**

o在接收记录量方面，InterSystems IRIS比AWS Aurora多831%
o在查询速度方面，InterSystems IRIS比AWS Aurora快485%

大家可以在自己的PC上使用Docker（3个CPU和7GB RAM）运行该测试！以下是部分结果：

- 在个人PC上运行InterSystems IRIS和MySQL 8.0：

o在接收记录量方面，InterSystems IRIS比MySQL 8.0多3043%

o在查询速度方面，InterSystems IRIS比MySQL 8.0快643%

- 在Ubuntu系统中运行InterSystems IRIS和SQL Server 2019

o在接收记录量方面，InterSystems IRIS比SQL Server 2019多223%，速度也更快

o在查询速度方面，InterSystems IRIS比SQL Server 2019快134,632%（请注意，数字没有打错哦！）

o为公平起见，我们未来将在AWS和Azure上对SQL Server进行测试。敬请期待！

在测试任何数据库的运行速度时，请先将速度测试运行一段时间进行预热，然后再记录结果。这样可以对数据库进行预扩展并执行其他操作。每次启动速度测试时，我们都需要清空表格重新开始。

1-在AWS上运行速度测试

请点击链接，查看如何在AWS上运行速度测试以便将InterSystems IRIS和其他数据库（如SAP HANA和AWS Aurora）进行对比。

2- 如何在PC上运行速度测试

在PC上运行速度测试的前提条件是：

- Docker和Docker Compose
- Git（可以克隆源代码）

目前，可以使用InterSystems IRIS、MySQL、SqlServer及SAP HANA在PC上运行本示例。

2.1 -在InterSystems IRIS Community上运行速度测试

要想在PC上运行本示例，请确保PC已经安装了Docker。您可以使用以下命令在Mac或Linux系统的PC上快速启动并运行：

```
wget https://raw.githubusercontent.com/intersystems-community/irisdemo-demo-
htap/master/docker-compose.yml
docker-compose up
```

如果在Windows系统中运行速度测试，请将**docker-compose.yml**
文件下载到一个文件夹中。打开命令提示符，并切换到该文件夹，然后运行**docker-compose up**

```
c:\MyFolder\docker-compose up
```

您也可以将存储库克隆到本地计算机上，从而获得完整的源代码。这时需要安装git，并将其放在git文件夹中：

```
git clone https://github.com/intersystems-community/irisdemo-demo-htap
cd irisdemo-demo-htap
docker-compose up
```

这两种技术都可行，并会触发示例中用于演示的镜像文件下载，之后将立刻启动所有的容器。

容器启动过程中，将出现与启动中容器相关的大量消息。这是正常的，不用担心！

启
动完
成后，它
会一直挂在那里，不会把控制权交还给你。这也是正常的。将窗口开着就可以。如果在此窗口上按**CTRL+C**，docker compose将停止所有容器并停止示例演示。

在所有容器启动之后，在浏览器上打开*http://localhost:10000*可查看示例的界面。点击**“ Run Test”**按钮即可运行HTAP Demo！

完成Demo演示后，返回到该终端并按**CTRL+C**。也可以输入以下命令，停止并删除仍在运行的容器：

```
docker-compose stop
docker-compose rm
```

这点很重要，特别是要在一个数据库（如InterSystems IRIS）和另一个数据库（如MySQL）之间反复运行速度测试时。

2.2 -PC上的MySQL

基于MySQL运行此示例，可以输入以下命令：

```
wget https://raw.githubusercontent.com/intersystems-community/irisdemo-demo-htap/master/docker-compose-mysql.yml
docker-compose -f ./docker-compose-mysql.yml up
```

现在，我们将下载一个不同的**docker-compose yml**文件：一个带有mysql后缀的文件。我们必须在docker-compose命令中使用**-f选项**来使用此文件。如前所述，将该终端窗口保持打开状态，并在浏览器上打开http://localhost:10000。

示例运行完成后，请返回终端并按**CTRL+C**。也可以输入以下命令，停止并删除仍在运行的容器：

```
docker-compose -f ./docker-compose-mysql.yml stop
docker-compose -f ./docker-compose-mysql.yml rm
```

这点很重要，特别是要在一个数据库（如InterSystems IRIS）和其他数据库之间反复运行速度测试时。

我们在测试中发现，InterSystems IRIS的数据接收速度**比MySQL和Amazon Aurora快25倍**。

2.3 -PC上的SQL Server 2019-GA-ubuntu-16.04

基于SQL Server运行此示例，可以输入以下命令：

```
wget https://raw.githubusercontent.com/intersystems-community/irisdemo-demo-htap/master/docker-compose-sqlserver.yml
docker-compose -f ./docker-compose-sqlserver.yml up
```

与前面一样，将该终端窗口保持打开状态，并在浏览器上打开http://localhost:10000。

我们在本地PC上运行速度测试后发现，InterSystems IRIS的数据接收速度**比SQL Server快2.5倍，查询速率则快400倍！**我们将与AWS RDS SQL Server相比较，进行速度测试并生成报告。

2.4 -PC上的SAP Hana

要在PC上基于SAP HANA运行速度测试，需要满足以下条件：

- 包含了Ubuntu 18 VM、docker和docker-compose的虚拟机——因为SAP HANA要求对Linux内核参数进行一些更改，否则将无法支持Mac或Windows上的Docker。另外，SAP HANA需要Linux内核4或更高版本。
- 该虚拟机至少配置9GB RAM，否则将无法启动！虚拟机崩溃后将显示无用的错误消息。

基于SAP HANA运行此Demo，可以**输入**以下命令：

```
git clone https://github.com/intersystems-community/irisdemo-demo-htap
cd ./irisdemo-demo-htap
./run.sh hana
```

等待下载镜像和启动容器。当docker-compose停止向屏幕写入时，一切已准备就绪。但是请耐心等待——SAP HANA大约需要6分钟才能启动！因此，屏幕会冻结一分钟左右，然后你会看到SAP HANA写入更多文本。这个重复写入过程大约持续6分钟。看到**“ 启动完成！（Startup finished!）”**后，就可以开始下一步了。如果在此过程中因为错误而发生崩溃，则可能需要配置更多的内存。

如您所知，与在InterSystems IRIS和MySQL中运行速度测试一样，使用SAP HANA测试不仅仅是运行docker-compose，还需要对Linux内核进行一些配置。大家可以通过**run.sh**文件来完成这些配置。
我们在虚拟机上运行速度测试后发现，InterSystems IRIS的数据接收速度**比SAP HANA快1.3倍，查询数据的速度快20倍，并且使用了更少的内存。**

3-资源

我们正在制作有关本示例的视频。在此期间，您可以**点击链接**查看一篇有意思的文章，该文章介绍了InterSystems IRIS的体系结构，并解释了为什么它能以更快的速度接收和查询数据。

4-该基准测试与YCSB或TPC-H等标准基准测试相比如何？

Yahoo Cloud Serving Benchmark（YCSB）是一项开源项目，其目的是开发一个框架和一组通用的工作负载，来评估不同的“ 键-值存储 ”和“ 云 ”服务的性能。

尽管YCSB上有一些工作负载可以描述成HTAP，但YCSB不一定要依靠SQL来完成。但是该基准必须依靠SQL。

TPC-H侧重于决策支持系统（DSS），而这并不是我们正在研究的用例。

此基准测试针对的是接收速度和查询响应时间之间的关系。我们有一个表格，其中包含许多不同数据类型的列。我们想衡量一个数据库在允许响应式查询的同时，其接收记录的速度能有多快。

这是一个复杂的问题。金融服务和物联网等许多行业都要求每秒必须接收数千条记录。在如此高的接收速率下，内存消耗得非常快。传统数据库需要写入磁盘，而内存数据库也将被迫不断写入磁盘（**更改log/journal**），甚至在某些情况下部分数据会写入内存，就像传统数据库一样）。**问题是：**如果InterSystems IRIS不仅要将事务日志写入磁盘（像内存数据库一样），还要异步保持数据库的最新状态，那么InterSystems IRIS是如何做到比内存数据库更快的呢？

一切都与效率有关。接收工作负载会使数据库非常繁忙。CPU和内存都将努力运转。一些内存数据库将尝试压缩内存中的数据。其他内存数据库在内存已满时会将数据持久化到磁盘。所有这些在我们试图实时查询数据库时都有发生。

我们想要证明，在某些工作负载上（例如股票交易、高接收吞吐量[物联网]等），内存数据库的性能不及InterSystems IRIS。这就是我们设计本测试的原因。**这意味着该测试比一般用途的测试要简单得多：**

（1）它只有一个表格，包含19个列和3种差别很大的数据类型

（2）表格上声明了主键（Primary Key）。

（3）我们执行的查询将通过主键（账户ID）获取记录，并使用固定的8个键随机查询：W1A1、W1A10、W1A100、

W1A1000、W1A10000、W1A100000、W1A1000000和W1A10000000。**这样做的原因如下：**

1. 我们知道，在生产系统的内存中保存所有数据是不可能的。内存数据库虽然具有复杂的体系结构，但当内存满了之后，就会将数据移出内存。为了简化测试并使其具有可比性，我们通过主键获取固定记录集，以避免对数据库中可能存在的其他类型的索引进行比较。
2. 通过账号（主键）获取客户账户数据记录是我们许多客户的真实工作负载。数据库在高速接收数据的同时，也需要对查询做出响应。
3. 由于账户ID是主键，因此数据库将使用它的首选（即最优）索引对其进行索引。这样在比较数据库时能够保持公平、简单。
4. 当我们不断请求相同的账号时，数据库有可能将该数据缓存在内存中。这对于内存数据库来说是一项轻松的任务。

InterSystems IRIS是一个混合型数据库。与传统数据库一样，它也尝试将数据保存在内存中。但由于每秒需要接收成千上万的记录，因此内存清理得非常快。通过这个测试可以看到，与其他传统数据库和内存数据库相比，InterSystems IRIS在缓存方面更加智能。你会看到：

（1）传统数据库在同时处理接收和查询时表现不佳

（2）内存数据库：

1. 在测试的最初几分钟内表现良好，随着内存填满，数据压缩变得更加困难，不可避免地要写入磁盘
2. 由于系统忙于接收、压缩数据，以及将数据移出内存等，因此查询性能表现不佳。

5-表是怎样的？

以下是我们发送到所支持的全部数据库的建表语句：

```
CREATE TABLE SpeedTest.Account
(
    account_id VARCHAR(36) PRIMARY KEY,
    brokerageaccountnum VARCHAR(16),
    org VARCHAR(50),
    status VARCHAR(10),
    tradingflag VARCHAR(10),
    entityaccountnum VARCHAR(16),
    clientaccountnum VARCHAR(16),
    active_date DATETIME,
    topaccountnum VARCHAR(10),
    repteamno VARCHAR(8),
    repteamname VARCHAR(50),
    office_name VARCHAR(50),
    region VARCHAR(50),
    basecurr VARCHAR(50),
    createdby VARCHAR(50),
    createdts DATETIME,
    group_id VARCHAR(50),
    load_version_no BIGINT
)
```

插入程序Ingestion Worker会尽可能多地发送INSERT数据，以测量每秒插入的记录数据量以及每秒的兆字节数。

查询程序Query worker
将通过accountid从此表中进行选择，并尝试选择尽可能多的记录来测量性能（即每秒选择的记录以及每秒选择的兆字节），以测试端到端性能，并提供工作量证明（Proof of Work）。

端到端性能与一些JDBC驱动程序最优化有关。如果仅执行查询操作，JDBC驱动程序可能不会从服务器获取记录，只有当实际请求列值后，JDBC驱动程序才会从服务器获取记录。

为了证明实际读取的正是我们选取的列，我们将返回的所有fild的字节加起来作为工作量证明。

6-如何实现接收和查询的最大吞吐量？

为了实现最大吞吐量，每个ingestion worker将启动多个线程，**每个线程将：**

（1）为上述表格的每一列准备1000个随机值。这样做是为了让每一列具有不同的数据类型和大小。所以我们希望生成可相应变化的记录

（2）对于要插入的每个新记录，ingestion worker将在每列的1000个值中随机选择一个值，准备好之后，该记录将被添加到批处理中

（3）使用批量插入，默认批量大小为每批1000条记录

Ingestion worker的默认线程数量是15，但是可以在测试过程中**单击“设置”**进行更改。

另一方面，query worker也启动多个线程来查询尽可能多的记录。如上所述，我们也将提供工作量证明。我们将读取返回的列，并汇总读取的字节数，以确保数据是从数据库通过连接传输进入query worker的，从而避免某些JDBC驱动程序实现优化后，仅在实际使用数据时才通过连接传输数据。我们实际使用返回的数据，并提供每秒读取数据的兆字节总和以及读取的总兆字节数作为工作量证明。

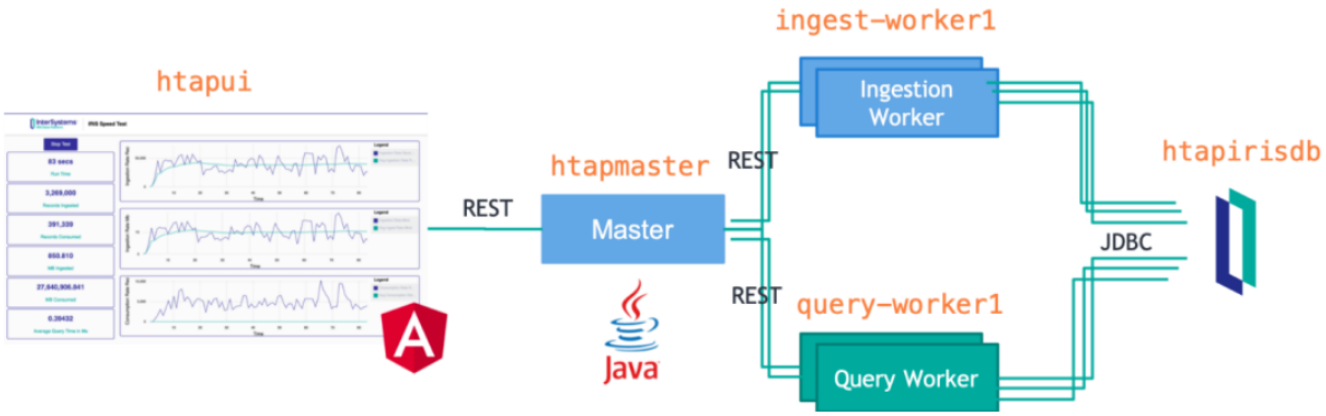
7-占用多少磁盘空间?

在接收171,421,000条记录后，我填满了一个70Gb的数据文件系统。这意味着，每条记录平均占用439个字节（向上取整）。

我还填写了第一个日志目录的100%和第二个日志目录的59%。这两个文件系统都有100Gb，这意味着171,421,000条记录将占用大约159Gb的日志空间，换言之，每条记录平均占用996个字节。

8-HTAP Demo体系架构

HTAP Demo的体系架构如下图所示：



本示例使用docker compose启动五项服务：

（1）htapui——这是用于运行示例的Angular UI。

（2）htapirisdb——由于本示例在InterSystems IRIS Community上运行，所以不需要InterSystems IRIS许可证即可运行。但请注意，InterSystems IRIS Community有两个重要限制条件：

1. 最多5个连接
2. 数据库最大为10Gb

（3）htapmaster——这是HTAP示例主程序。UI与主程序对话，主程序与worker对话，以及启动/停止速度测试，并收集指标。

（4）ingest-worker1——这是插入程序ingestion worker。实际上，大家可以拥有多个ingestion worker，只需给每个worker分配不同的服务名称即可。它们将尝试尽快地将记录插入数据库。

（5）query-worker1——这是查询程序query worker，大家也可以拥有多个query worker。它们将尝试尽快地从数据库中读取记录。

在PC上运行示例时，我们使用的是Docker和Docker Compose。Docker Compose需要一个**docker-compose.yml**来描述这些服务及其使用的Docker镜像。本示例实际上提供了许多docker-compose.yml文件，并且很快将添加更多此类文件：

（1）docker-compose.yml——这是针对InterSystems IRIS Community（上述项目及图片中有所描述）运行速度测试的默认演示程序。

（2）docker-compose-mysql.yml——这是针对MySQL的速度测试。大家应该注意到，该测试结果表明，InterSystems IRIS比MySQL快25倍。在Amazon Aurora MySQL（MySQL的微调版本）上运行此测试可得到相同的结果。

（3）docker-compose-sqlserver.yml——这是针对使用Docker部署的SqlServer的速度测试。

（4）docker-compose-enterprise-iris.yml——如果要在标准版本的InterSystems IRIS上运行速度测试示例，这是一个docker-compose.yml的文件例子。

9. 可以在没有容器的情况下在InterSystems IRIS集群上运行本Demo吗？

可以！完成此示例最简便的方法是将此存储库克隆到即将运行master（主程序）和（在同一服务器上运行的）UI的每台服务器上以及每种worker类型（接收和查询worker）上。你可以根据自己的需要，拥有任意数量的接收worker和查询worker！

对于InterSystems IRIS，请查看文件夹*./standalonescripts/iris-jdbc.* 中的文件。每个服务器都有一个脚本：

（1）在主程序上：startmasterandui.sh——此脚本将启动主程序和UI。

（2）在Ingestion Worker上：startingestionworker.sh——此脚本将启动Ingestion Worker，后者随后将与主程序连接并进行注册。

（3）在Query Worker上：startqueryworker.sh——此脚本将启动query worker，然后query worker将与主程序连接并进行注册。

对于InterSystems IRIS，大家有两种选择：

（1）可以使用startiris.sh脚本在Docker容器上启动InterSystems IRIS服务器进行快速测试。

（2）可以手动或使用ICM设置InterSystems IRIS集群。然后做一些有趣的事情，比如：

1. 使接收和查询worker都指向同一InterSystems IRIS

2. 使用ECP配置InterSystems IRIS，让ingestion worker指向数据库服务器，同时让query worker指向ECP服务器
3. 配置分片的InterSystems IRIS集群
4. 等等

只要确保更改startmaster.sh脚本中对应使用正确的InterSystems IRIS端点、用户名和密码来配置环境变量。

10-自定义

10.1 -如何配置本Demo让其与更多worker、线程等一起工作？

Docker-compose.yml文件中的环境变量支持配置所有内容。docker-compose.yml文件只是个不错的起点：大家可以复制它们并对副本进行更改，从而得到更多的worker（如果在PC上运行，不会有太大区别），每个worker类型都可以得到更多线程数，还可以更改接收数据的批处理大小，以及各查询之间的等待时间（以毫秒为单位）等。

10.2 -可以更改表的名称或结构吗？

可以，但必须：

（1）在PC上将复制此存储库

（2）更改源代码

（3）使用shellscript build.sh在PC上重建demo。

更改表的结构也很简单。

复制了该存储库后，需要更改*/image-master/projects/master/src/main/resources*文件夹下的文件。

如果更改表的结构，请确保使用与现有表相同的数据类型，这些数据类型是受支持的。另外还可以更改表的名称。

其次，需要配合更改其他*** .sql**脚本，如INSERT脚本、SELECT脚本等。

最后，只需**运行build.sh**来重建demo就可以了！

11-其他示例应用程序

我们还有一些其他涉及不同主题的InterSystems IRIS
示例应用程序，例如NLP、ML、与AWS服务的集成、Twitter服务、性能基准测试等。以下是其中的部分内容：

（1） HTAP Demo——混合事务分析处理（HTAP）基准。可以测试InterSystems IRIS同时插入和查询数据的速度。你会发现它的速度比AWS Aurora快20倍！

（2） 欺诈预防——InterSystems IRIS通过机器学习和制定业务规则，防止金融服务交易中出现欺诈行为。

（3） Twitter情绪分析——演示InterSystems IRIS如何实时使用Tweet，并通过其NLP（自然语言处理）和业务规则功能来评估Tweet的情绪和元数据，从而决定何时与某人联系以提供支持。

（4） HL7协议和SMS（文本消息）应用程序——演示InterSystems IRIS医疗版如何解析HL7协议消息，从而给患者发送SMS（文本消息）提醒。它还演示了基于存储在标准化数据湖中预约数据的实时仪表板。

（5） Readmission Demo——患者再入院在医疗保健领域被称为"机器学习的Hello World"。针对这个问题，我们在本示例中演示了如何使用InterSystems IRIS安全地构建并运行用于实时预测的ML模型，以及如何将其集成到应用程序中。本InterSystems IRIS医疗版示例旨在展示如何构建针对再入院问题的完整解决方案。

12-支持的数据库

这是目前为止支持的数据库列表：

- **Runing on your PC with docker-compose (NO mirroring/replication)**
  - InterSystems IRIS 2020.2
  - MySQL 8.0
  - MariaDB 10.5.4-focal
  - MS SQL Server 2019-GA-ubuntu-16.04
  - SAP HANA Express 2.0 (on Linux VM only)
  - Postgres 12.3
- **Running on AWS:**
  - InterSystems IRIS (with or without mirroring)
  - AWS RDS Aurora (MySql) 5.6.10a (parallel query) with replication
  - AWS RDS SQL Server 2017 Enterprise Edition (production deployment) with replication
  - AWS RDS Postgres (production deployment) with replication
  - AWS RDS MariaDB (production deployment) with replication
  - SAP HANA Express Edition 2.0 without replication
  - SAP Sybase ASE 16.0 SP03 PL08, public cloud edition, premium version, without replication
  - AWS RDS Oracle (production deployment) with replication

注：本文为节选，欢迎点击原文链接，了解更多详情。

#IRIS Analytics Architect #JDBC #性能 #InterSystems IRIS #InterSystems IRIS for Health
在 InterSystems Open Exchange 上检查相关应用程序

源URL:https://cn.community.intersystems.com/post/intersystems-iris%E6%95%B0%E6%8D%AE%E5%B9%B3%E5%8F%B0%EF%BC%9A%E6%95%B0%E6%8D%AE%E6%8E%A5%E6%94%B6%E9%80%9F%E5%BA%A6%E6%B5%8B%E8%AF%95