文章
Nicky Zhu · 九月 9, 2021 阅读大约需 13 分钟

# IRIS 2021 技术文档 First Look 22 -- 技术概要：部署 InterSystems 分片集群

本文档将您介绍InterSystems IRIS®数据平台的分片（sharding）功能，以及它在分片集群中的使用，以水平扩展 InterSystems IRIS 的数据量。

作为本指南的一部分，您将使用 ICM 在公共云中提供的分片集群，并了解分片表（sharding a table）如何在集群中的分片之间分布其行。

# 分片（Sharding）如何帮助您？

您感受到大数据（Big Data）的热度了吗？

无论是否准备好了，我们都在管理比以往任何时候都多的数据，并被要求用这些数据做更多的事情——所需的响应时间也越来越短。无论您是照顾一千万名患者、每天处理数十亿的金融订单，追踪一个星系的恒星，还是监控一千个工厂的引擎，数据平台不仅要支持您目前的数据工作量，而且还必须在保持性能的同时进行*扩展（Scale）*，以满足不断增长的需求，避免业务中断。每个特定业务的工作量对其运行的数据平台提出了不同的挑战 — 而随着工作量的增加，这些挑战将变得更加严峻。

InterSystems IRIS 包含一套全面的功能来扩展（Scale）您的应用程序，这些功能可以单独或组合应用，这取决于您的工作量的性质和它所面临的特定性能挑战。其中之一是分片（sharding），它在多个服务器上对数据及其相关缓存进行分区，为查询和数据摄取提供灵活、价优的性能扩展，同时通过高效的资源利用使基础设施价值最大化。InterSystems IRIS *分片集群（sharded cluster）* 可以为各种应用程序提供显著的性能优势，特别是对于那些工作量包括以下一项或多项的应用程序：

- 高容量或高速数据摄入，或组合。
- 相对较大的数据集，返回大量数据的查询，或两者兼有。
- 执行大量数据处理的复杂查询，例如扫描磁盘上的大量数据或涉及大量计算工作的查询。

这些因素各自都会影响分片（sharding）的潜在收益，但如果将它们结合起来，收益可能会得到增强。例如，所有这三个因素的组合——快速摄入的大量数据、大型数据集以及检索和处理大量数据的复杂查询——使得如今的许多分析工作量非常适合进行分片（sharding）。

请注意，这些特征都与数据有关；InterSystems IRIS 分片（sharding）的主要功能是*扩展数据量（sharded cluster）*。但是，当涉及部分或所有这些数据相关因素的工作量也经历了来自大量用户的非常高的查询量时，分片集群也可以包括*扩展用户量（scale for user volume）*功能。分片（sharding）也可以与垂直扩展相结合。通过 InterSystems IRIS，您可以为工作量的性能挑战创建恰到好处的整体扩展解决方案。

# 分片（Sharding）是如何工作的？

分片架构的核心是跨多个系统对数据及其相关缓存进行分区。分片集群将大型数据库表水平（即按行）划分为多个 InterSystems IRIS 实例，称为*数据节点（data node）*，同时允许应用程序通过这些实例中的任何一个访问这些表。每个数据节点在集群分片数据中的份额被称为*分片（shard）*。这种架构有三个优势：

- 并行处理（Parallel processing）

查询在数据节点上并行运行，合并查询结果，并作为完整的查询结果返回给应用程序，在许多情况下显著提高了执行速度。

- 分区缓存（Partitioned caching）

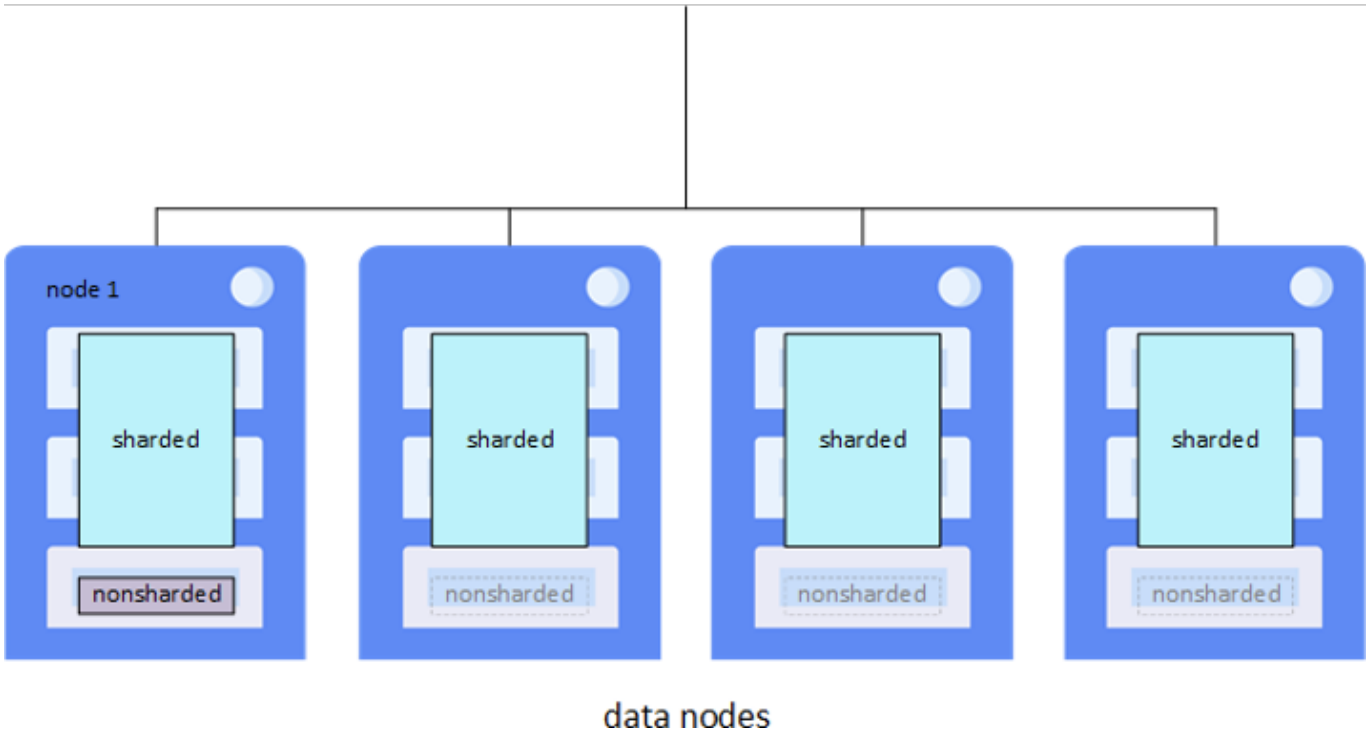
每个数据节点都有自己的专用缓存，而不是由单个实例的缓存为整个数据集服务，这大大降低了缓存溢出和强制降低性能的磁盘读取的风险。

- 并行加载（Parallel loading）

数据可以并行加载到数据节点上，从而减少摄取工作量和查询工作量之间的缓存和磁盘争夺，提高两者的性能。

一个被称为*分片管理器（sharding manager）*的联合软件组件会跟踪哪些数据位于哪些数据节点上，并相应地指导查询。非分片数据存储在配置的第一个数据节点上，称为*数据节点 1*（它也存储代码和元数据）。从应用程序 SQL 的角度来看，分片表（sharded table）和非分片表（nonsharded table）之间的区别是完全透明的。

***图 1: 一个基本的分片集群***




# 试一试！部署和演示 InterSystems IRIS 分片集群

在这次探索中，您将

- 使用 InterSystems 云管理器（InterSystems Cloud Manager，ICM）在公共云中部署基本的分片集群。

ICM 是一个自动化的命令行工具，可以轻松地在云中配置基础设施，并在提供的节点上部署 InterSystems IRIS 和其他服务。ICM 负责所有需要的 InterSystems IRIS 配置，所以如果您指定了一个分片集群，该集群在部署完成后就可以使用了。

- 使用不同的分片键（shard key）从相同的数据中创建三个分片表（sharded table），并观察数据节点上跨分片的行的不同分布。

分片键是用来确定分片表的哪些行存储在哪些分片上的一个或多个字段。默认情况下，RowID 被用作分片键，这种分布对于大多数分片表来说是最有效的方法，因为它提供了数据均匀分布的最佳保证，并允许最有效的并行数据加载。然而，在一种情况下，用户定义的键可能是有益的。当您对在字段或用于连接它们的字段上的查询中经常连接的两个大表进行分片时，要连接的行被存储在相同的分片上，并且连接可以在每个分片上本地执行，而不是跨分片执行（across the shard），从而提供显著的性能提升。这被称为 *cosharded 连接（cosharded join）*。

- 在所有三个分片表上运行相同的查询，以证明跨分片的行分布对查询是完全透明。

为了向您介绍分片（sharding），而又不至于让您在细节上陷入困境，我们保持了简单的探索。生产中的分片集群需要计划和许多决策，所以确保不要把对分片（sharding）的探索和真实情况相混淆！例如，在设计一个生产集群时，您将检查您的架构和表，以决定哪些表将被分片，然后根据分片数据的预期工作集和服务器上数据库缓存的可用内存量，决定部署多少个数据节点（通常为 4 到 16 个）。然而，在本次探索中，您将部署一个由两个数据节点组成的基本集群。本文档末尾列出的参考资料将使您对生

产中使用分片（sharding ）的情况有一个很好的了解。*Scalability Guide*（《*可扩展性指南*》）中的 "Horizontally Scaling for Data Volume Sharding（《水平扩展数据卷分片[Sharding]》)"一章提供了有关分片（sharding）和分片集群的详细信息。

这些说明假定您有 InterSystems IRIS 分片（sharding）许可证并可以访问 InterSystems 软件下载。

# 使用 ICM 部署集群

您可以在 *First Look：InterSystems Cloud Manager（《技术概要：InterSystems 云管理器》）*中找到使用 ICM 在亚马逊网络服务（Amazon Web Services）公共云平台上部署分片集群的程序；特别是在 Try It!Deploy InterSystems IRIS in the Cloud with ICM（试一试！使用 ICM 在云中部署 InterSystems IRIS）部分。您可以使用此处显示的整个程序。当您进入自定义定义.json（Customize definitions.json）时，验证每个 LicenseKey 字段的值是否是 defaults.json 文件中 LicenseDir 字段指定的暂存位置中的 InterSystems IRIS 分片（sharding）许可证的文件名。当您完成了部署阶段（部署 InterSystems IRIS[Deploy InterSystems IRIS]）后，留在 ICM 容器中。

**注意：** 正如 *Scalability Guide*(《*可扩展性指南*》)中的 Deploying the Sharded Cluster(部署分片集群) 所描述的那样，您也可以在现有的物理、虚拟或云机器上安装 InterSystems IRIS 实例，并使用分片 API（Sharding API） 或管理门户（Management Portal）部署分片集群，或使用多种方法之一自动化部署分片集群。

# 使用不同的分片键以不同的方式分布行，然后查询分片表

为了了解分片集群如何根据您使用的分片键（shard key）在数据节点上的分片上对分片表（sharded table）进行分区，您将连接到集群并执行以下步骤：

- 创建并填充一个小型的非分片表。
- 使用三个不同的分片键，创建三个与非分片表具有相同字段的分片表 。
- 从非分片表中填充分片表。

- 从分片表中选择行，以查看在每种情况下，行在分片（数据节点）上的分布情况。

当一个表被分片时，RowID 值被从每个分片的不同范围分配，从相隔甚远的范围开始；相同范围的行在同一个分片上。当处理一个小表时，就像本例一样，分片的 RowID 很容易区分，清楚地告诉您哪些行在一个分片上，哪些在另一个分片上（尽管您无法分辨哪个分片是哪个）。在 RowID 上分片的表的行的分布方式与行中的数据没有关系，如果您清空并重新加载一个表，可能会有所不同。另一方面，用户定义的分片键，根据键中一个或多个字段的值来分配行，因此当您清空并重新加载表（假设值和分片数没有改变）时将是相同的。

- 查询所有三个表以获得其中一个字段的最大长度，来显示跨分片（across shards）的行的分布对查询是透明的。

对于这部分探索，请使用以下程序：

1. 为任意一个数据节点实例打开终端（Terminal）窗口。
    - 在 ICM 命令行上， 发出命令 **icm session** 命令，使用 **-machine** 选项指定一个数据节点的名称，使用 **- namespace** 选项指定集群命名空间 **IRISCLUSTER** (或者使用默认文件中的 Namespace 字段指定的命名空间，如果不同的话)。 如果您不确定节点的名称，可以使用 **icm inventory** 命令来显示节点名称：

```
$ icm inventory
```

Machine IP Address DNS Name Region Zone

Acme-DATA-TEST-0001 00.53.183.209 ec2-00-53-183-209.us-west-1.compute.amazonaws.com us-west-1 c

Acme-DATA-TEST-0002 00.53.183.185 ec2-00-53-183-185.us-west-1.compute.amazonaws.com us-west-1 c

选择一个节点（哪个节点并不重要）并发出命令：

```
icm session -machine Acme-DATA-TEST-0002 -namespace IRISCLUSTER
```

**icm session** 命令为指定节点上的 InterSystems IRIS 实例打开一个终端（Terminal）窗口。

> 如果您使用 %SYSTEM.Cluster.Sharding API 手动部署您的集群，使用 *InterSystems IRIS Basics：Connecting an IDE*（《*技术概要：连接一个 IDE*》）中的procedure described for your instance（为您的实例描述的程序），在任一实例上打开一个终端（Terminal）窗口，然后切换到 **IRISCLUSTER** 命名空间 (或者初始化集群时指定的命名空间，如果不同的话)。

```
Node: Acme-DATA-TEST-0002, Instance: IRIS USER>set $namespace="IRISCLUSTER" IRISCLUSTER>
```

2. 打开终端（Terminal）SQL shell：IRISCLUSTER>do $SYSTEM.SQL.Shell() [SQL]IRISCLUSTER>>
3. 使用以下 SQL 语句，创建并填充非分片表 test.nonsharded。

```
CREATE TABLE test.nonsharded (field1 CHAR(5), field2 CHAR(5)) INSERT INTO test.nonsharded (field1,field2) VALUES ('one','one') INSERT INTO test.nonsharded (field1,field2) VALUES ('one','two') INSERT INTO test.nonsharded (field1,field2) VALUES ('one','three') INSERT INTO test.nonsharded (field1,field2) VALUES ('two','one') INSERT INTO test.nonsharded (field1,field2) VALUES ('two','two') INSERT INTO test.nonsharded (field1,field2) VALUES ('two','three') INSERT INTO test.nonsharded (field1,field2) VALUES ('three','one') INSERT INTO test.nonsharded (field1,field2) VALUES ('three','two')
```

```
INSERT INTO test.nonsharded (field1,field2) VALUES ('three','three')
```

使用 SELECT 来查看表的内容：

```
SELECT * FROM test.nonsharded field1 field2

one one

one two

one three

two one
```

two two

two three

three one

three two

three three

4. 使用 test.nonsharded 中的相同字段创建三个分片表，在第一个表中使用默认的分片键（RowID），在第二个表中使用 **field1** 字段，在第三个表中使用 **field2** 字段，并使用从 test.nonsharded 中选择的 INSERT INTO 语句填充每一个：

```
CREATE TABLE test.rowid (field1 CHAR(5), field2 CHAR(5), SHARD)

INSERT INTO test.rowid (field1,field2) SELECT field1,field2 FROM test.nonsharded

CREATE TABLE test.field1 (field1 CHAR(5), field2 CHAR(5), SHARD KEY (field1)) INSERT INTO test.field1 (field1,field2) SELECT field1,field2 FROM test.nonsharded

CREATE TABLE test.field2 (field1 CHAR(5), field2 CHAR(5), SHARD KEY (field2)) INSERT INTO test.field2 (field1,field2) SELECT field1,field2 FROM test.nonsharded
```

5. 使用 SELECT *,%ID 来显示每个分片表的内容和它在分片上的 RowID。

```
SELECT *,%ID FROM test.rowid ORDER BY %ID

field1 field2 ID one one 1

one two 2

two one 3

two two 4

three three 5

one three 256000

two three 256001

three one 256002
```

three two 256003

这个分布并不反映 field1 或 field2
的值（具有各自全部三个值的行位于两个分片上）。如果您删除、重新创建并重新加载
test.rowid，分布可能会有所不同。

SELECT *,%ID FROM test.field1 ORDER BY %ID

| field1 one one one | field2 | ID |
|---|---|---|
| | one | 1 |
| | two | 2 |
| | three | 3 |
| two | one | 256000 |
| two | two | 256001 |
| two | three | 256002 |
| three | one | 256003 |
| three | two | 256004 |
| three | three | 256005 |

在 **field1** 字段上分片（sharding）会分布行，使那些具有相同 **field1** 值的行被放在同一个分片上。在这种情况下，值为 **1** 的行在一个分片上，值为 **2** 和 **3**
的行在另一个分片上，但是哪个数值最终在哪个分片上取决于有多少个分片和多少个数值。

SELECT *,%ID FROM test.field2 ORDER BY %ID

| field1 | field2 | ID |
|---|---|---|
| one | one | 1 |
| two | one | 2 |
| three | one | 3 |

```
one     two     256000

one     three   256001

two     two     256002

two     three   256003

three   two     256004

three   three   256005
```

这里，分布是由 **field2** 字段的值决定的。

6. 最后，作为分片如何将工作分配到数据节点的示例，对所有三个分片表使用以下 SELECT 语句：

```
SELECT MAX(LENGTH(field2)) FROM <table>
```

在每种情况下，结果都是 **5**，即最长值的长度是 **3**，因为跨分片的行分布对查询来说是完全透明的。MAX(LENGTH(field2)) 表达式在每个分片上独立计算，分片管理器选择它们返回结果的 MAX()。例如，当查询在 test.field2 表上运行时，一个分片返回 **3**，因为它在 **field2** 字段中只有值 **1**，而另一个分片返回 **5**；然后分片管理器选择 **5** 作为两个结果中较大的那个。

如果您愿意，可以使用 **EXPLAIN** 来显示查询计划，明确指出如何将工作发送到分片上：

```
EXPLAIN SELECT MAX(LENGTH(field2)) FROM <table>
```

# 更多分片集群选项

分片集群的其他选项包括如下内容：

- 您可以随时添加数据节点，并跨扩展的数据节点集重新平衡现有的分片数据。重新平衡（Rebalancing）不能与查询和更新同时进行，因此只有在分片集群离线并且没有其他分片操作的情况下才能进行。*(Scalability Guide*[*《可扩展性指南》*]中的 Add Nodes and Rebalance Data[添加节点和重新平衡数据] )
- 要为集群上的数据添加高可用性，您可以将数据节点部署为镜像故障转移对。*( Scalability Guide*[*《可扩展性指南》*]中的 Mirror for High Availability[高可用性镜像])。
- 对于需要极低查询延迟的高级用例（可能与不断涌入的数据相冲突），可以添加*计算节点*，为查询服务提供一个透明的缓存层。当集群包含计算节点时，只读的查询会自动在计算节点上并行执行，而不是在数据节点上执行；所有写入操作（插入、更新、删除和 DDL 操作）继续在数据节点上执行。这种分工将查询和数据摄取工作量分开，同时保持并行处理和分布式缓存的优势，提高两者的性能。(*Scalability Guide*[*《可扩展性指南》*]中的 Deploy Compute Nodes[部署计算节点])

# 了解有关分片（Sharding）的更多信息

要了解有关分片的更多信息，请参见

- Data Platform Scalability Technology Overview（《数据平台可扩展性技术概要》）(视频)
- Introduction to Sharding（《分片简介》）(在线课程)
- Sharding Basics（《分片基础》）(在线课程)
- Experience InterSystems IRIS in the Cloud（在云中体验 InterSystems IRIS）(在线体验)
- Scalability Guide（《可扩展性指南》）
- First Look:InterSystems Cloud Manager（《技术概要：InterSystems 云管理器》）

#InterSystems IRIS

---

**源 URL:** https://cn.community.intersystems.com/post/iris-2021-%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3-first-look-22-%E6%8A%80%E6%9C%AF%E6%A6%82%E8%A6%81%EF%BC%9A%E9%83%A8%E7%BD%B2-intersystems-%E5%88%86%E7%89%87%E9%9B%86%E7%BE%A4