



图例：
“Yrsk”中的状态数据可以外部访问，这样就
可以方便地实现跨集群的数据迁移。

Architecture and Enabling Queryable State (可查询的架构和启用)

Yrsk的可查询状态服务包含三个过程：
1. 外部访问：通过QueryableStateClient接口，外部客户端可以访问状态数据。
2. 内部访问：通过QueryableStateClient接口，TaskManager可以访问状态数据。
3. 内部访问：通过QueryableStateClient接口，JobManager可以访问状态数据。

Queryable State (可查询的状态)

Queryable State (可查询的状态) 是指存储在QueryableStateClient中的状态数据。它可以通过QueryableStateClient接口进行访问。QueryableStateClient接口提供了以下方法：
- getQueryableState(String key, Serializable value): 获取指定key的状态数据。
- setQueryableState(String key, Serializable value): 设置指定key的状态数据。
- deleteQueryableState(String key): 删除指定key的状态数据。

5. Evolving Stateful Applications (升级有状态的应用)

升级有状态的应用是指在不中断服务的情况下，对应用的状态数据进行更新。这可以通过以下步骤实现：
1. 创建新的状态数据。
2. 将新的状态数据写入QueryableStateClient。
3. 从QueryableStateClient中读取新的状态数据。
4. 将新的状态数据写入应用。
5. 删除旧的状态数据。

4. Performance and Robustness of Stateful Applications (有状态的应用性能和健壮性)

有状态的应用性能和健壮性是指应用在处理大量数据时的稳定性和效率。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

3. Ensuring the Maintainability of Stateful Applications (确保有状态应用的可持续性)

确保有状态应用的可持续性是指应用能够长期稳定运行。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

2. Enabling Failure Recovery for Stateful Applications (为有状态的应用程序启用故障恢复)

为有状态的应用程序启用故障恢复是指应用能够在发生故障后自动恢复。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

1. Implementing Stateful Functions (实现有状态函数)

实现有状态函数是指应用能够处理有状态的数据。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

Queryable State

5. Evolving Stateful Applications (升级有状态的应用)

4. Performance and Robustness of Stateful Applications (有状态的应用性能和健壮性)

3. Ensuring the Maintainability of Stateful Applications (确保有状态应用的可持续性)

Chapter 7 Stateful Operators and Applications (有状态的算子和应用)

Implementing Stateful Functions (实现有状态函数)

2. Enabling Failure Recovery for Stateful Applications (为有状态的应用程序启用故障恢复)

1. Implementing Stateful Functions (实现有状态函数)

- 前言：
1. 重点介绍有状态用户自定义函数的实现
 2. 介绍有状态应用程序的可靠性和健壮性
 3. 解释了如何应用用户自定义的函数中定义不同类型的状态与交互
 4. 讨论有状态函数以及有状态函数function state的大小
 5. 展示了如何有状态函数配置为可靠以及从外部应用程序访问它

1.1 Declaring Keyed State as RuntimeContext (在RuntimeContext中声明“keyed state”)

1.1.1 keyed state的定义

keyed state是指存储在QueryableStateClient中的状态数据。它可以通过QueryableStateClient接口进行访问。QueryableStateClient接口提供了以下方法：
- getQueryableState(String key, Serializable value): 获取指定key的状态数据。
- setQueryableState(String key, Serializable value): 设置指定key的状态数据。
- deleteQueryableState(String key): 删除指定key的状态数据。

1.2 有状态的定义

有状态的定义是指应用能够处理有状态的数据。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

1.3 Implementing Stateful Functions (实现有状态函数)

实现有状态函数是指应用能够处理有状态的数据。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

2. Enabling Failure Recovery for Stateful Applications (为有状态的应用程序启用故障恢复)

为有状态的应用程序启用故障恢复是指应用能够在发生故障后自动恢复。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。

1. Implementing Stateful Functions (实现有状态函数)

实现有状态函数是指应用能够处理有状态的数据。这可以通过以下方法实现：
1. 使用分布式存储。
2. 使用分布式计算。
3. 使用分布式网络。
4. 使用分布式安全。