

Solr技术原理

www.huawei.com





目标

- 学完本课程后，您将能够：
 - 掌握**Solr**基本功能、基本概念；
 - 熟悉产品的应用场景；
 - 掌握产品的系统架构；
 - 掌握产品的关键特性；

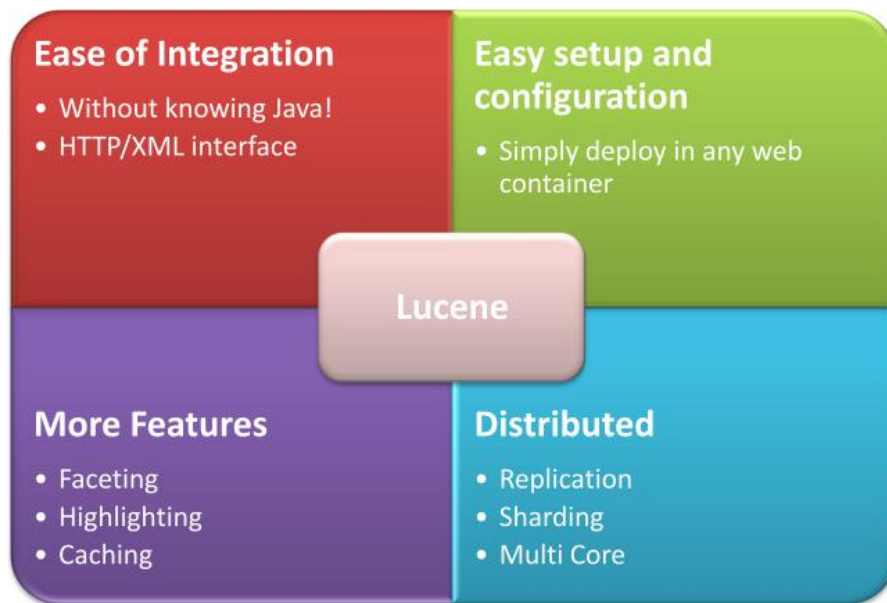


目录

1. 什么是Solr
2. 应用场景
3. 概念体系
4. 系统架构
5. 关键特性
6. 华为增强特性

什么是Solr

- **Solr**是一个高性能，基于**Lucene**的全文检索服务，也可以作为**NoSQL**数据库使用。
- **Solr**对**Lucene**进行了扩展，提供了比**Lucene**更为丰富的查询语言，同时实现了可配置、可扩展，并对查询性能进行了优化，还提供了一个完善的功能管理界面。
- **SolrCloud**从**Solr 4.0**版本基于**Solr**和**Zookeeper**进行开发，具有开创意义的分布式索引和搜索方案。





目录

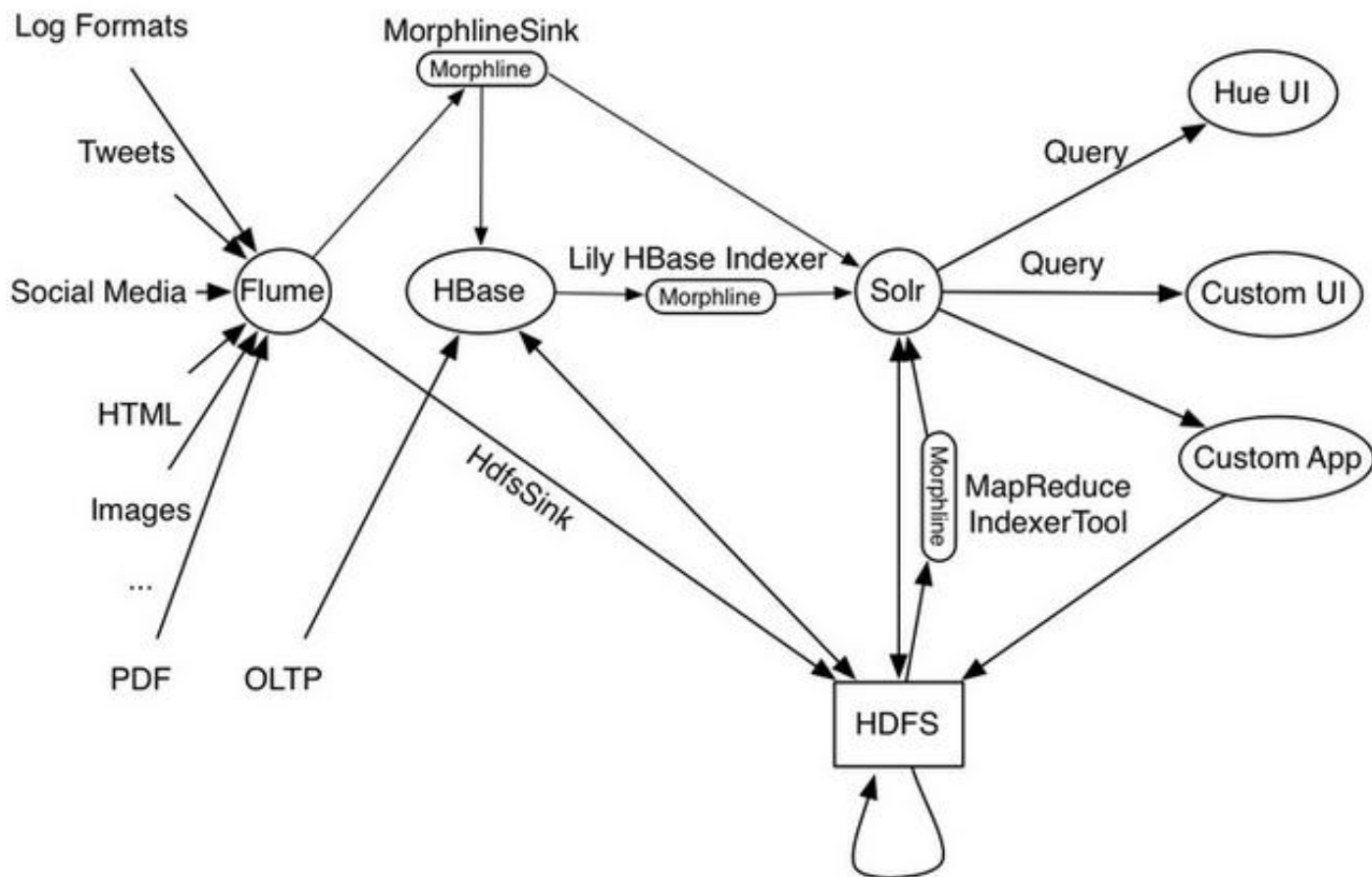
1. 什么是Solr
2. 应用场景
3. 概念体系
4. 系统架构
5. 关键特性
6. 华为增强特性

Solr应用场景

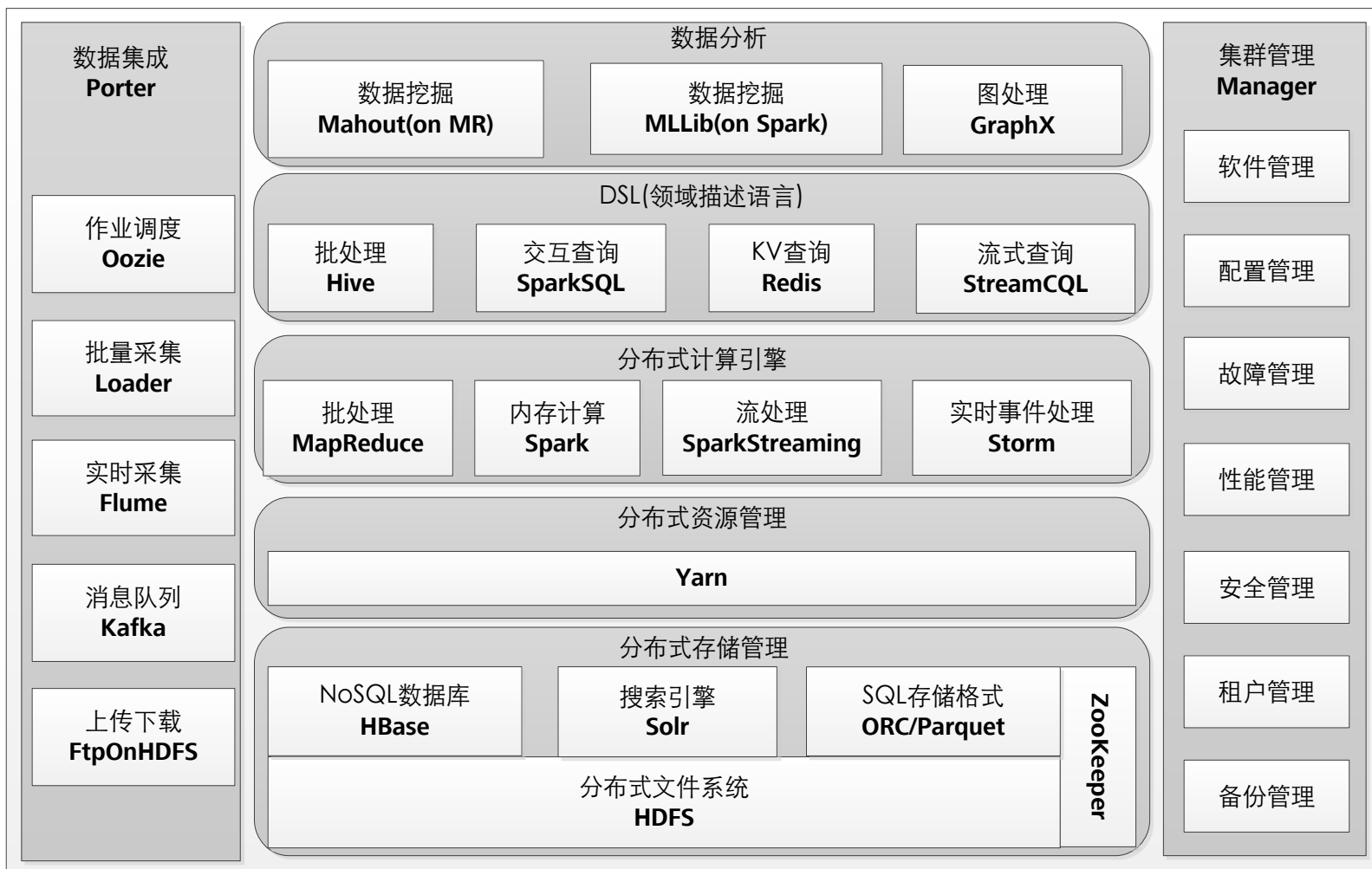
- 待检索的数据类型复杂：如需要查询的数据有结构化数据（关系型数据库等）、半结构化数据（网页、**XML**等）、非结构化数据（日志、图片、图像等）等，**Solr**可以对以上数据类型进行清洗、分词、建立倒排索引等一系列操作（建立索引），然后提供全文检索（查询）的能力。
- 检索条件多样化（如涉及字段太多），常规查询无法满足：全文检索（查询）可以包括词和短语，或者词或短语的多种形式。
- 读数据取远多于写入数据的应用场景。

Solr功能简介

- 生态圈



Solr在平台中的位置





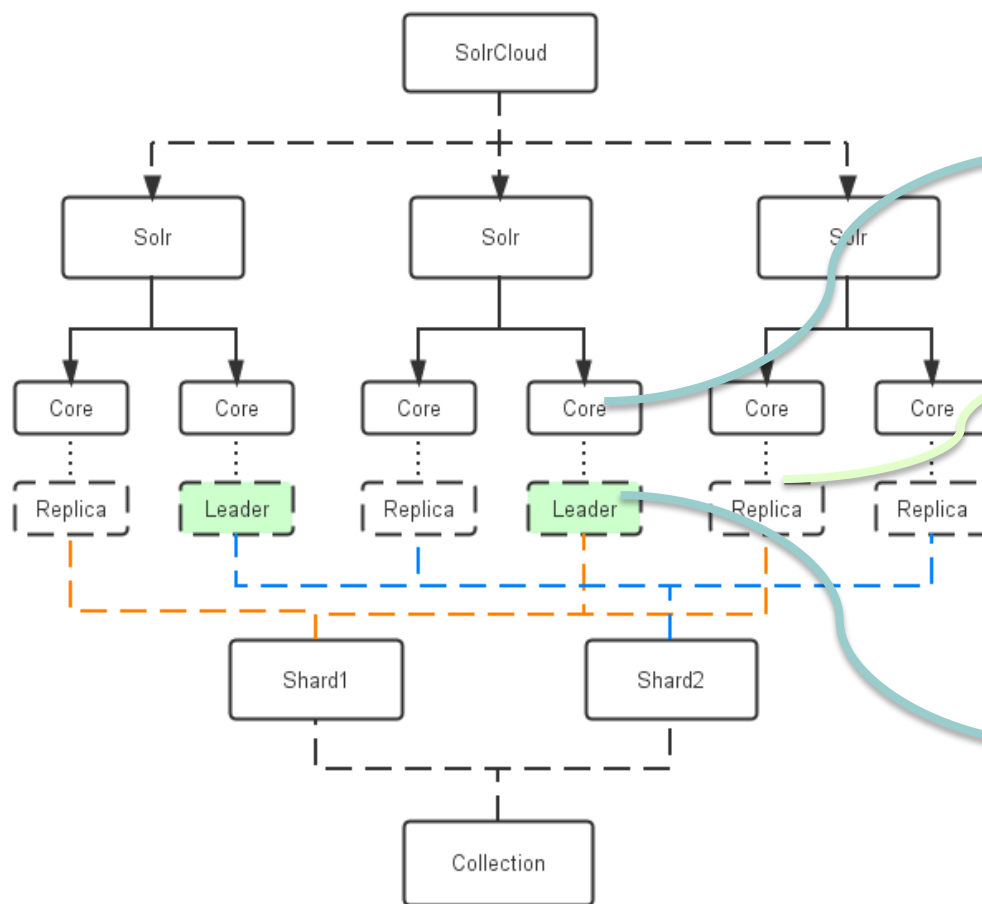
目录

1. 什么是Solr
2. 应用场景
3. 概念体系
4. 系统架构
5. 关键特性
6. 华为增强特性

Solr概念体系—总述

常见术语	描述
Config Set	Solr Core 提供服务必须的一组配置文件。包括 solrconfig.xml(SolrConfigXml) 和 schema.xml (SchemaXml) 等。
Core	即 Solr Core ，一个 Solr 实例中包含一个或者多个 Solr Core ，每个 Solr Core 可以独立提供索引和查询功能，每个 Solr Core 对应一个索引或者 Collection 的 Shard 的副本(replica)。
Shard	Collection 的逻辑分片。每个 Shard 都包含一个或者多个 replicas ，通过选举确定哪个是 Leader 。
Collection	在 SolrCloud 集群中逻辑意义上的完整的索引。它可以被划分为一个或者多个 Shard ，它们使用相同的 Config Set 。
Replica	Shard 下的实际存储索引的一个副本，与 Core 对应。
Leader	赢得选举的 Shard replicas 。当索引 documents 时， SolrCloud 会传递它们到此 Shard 对应的 leader ， leader 再分发它们到全部 Shard 的 replicas 。
ZooKeeper	它在 SolrCloud 是必须的，提供分布式锁、处理 Leader 选举、管理配置等功能。

Solr概念体系—逻辑视图



Core: 也就是Solr Core，一个Solr中包含一个或者多个Solr Core，每个Solr Core可以独立提供索引和查询功能，每个Solr Core对应一个索引或者Collection的Shard，

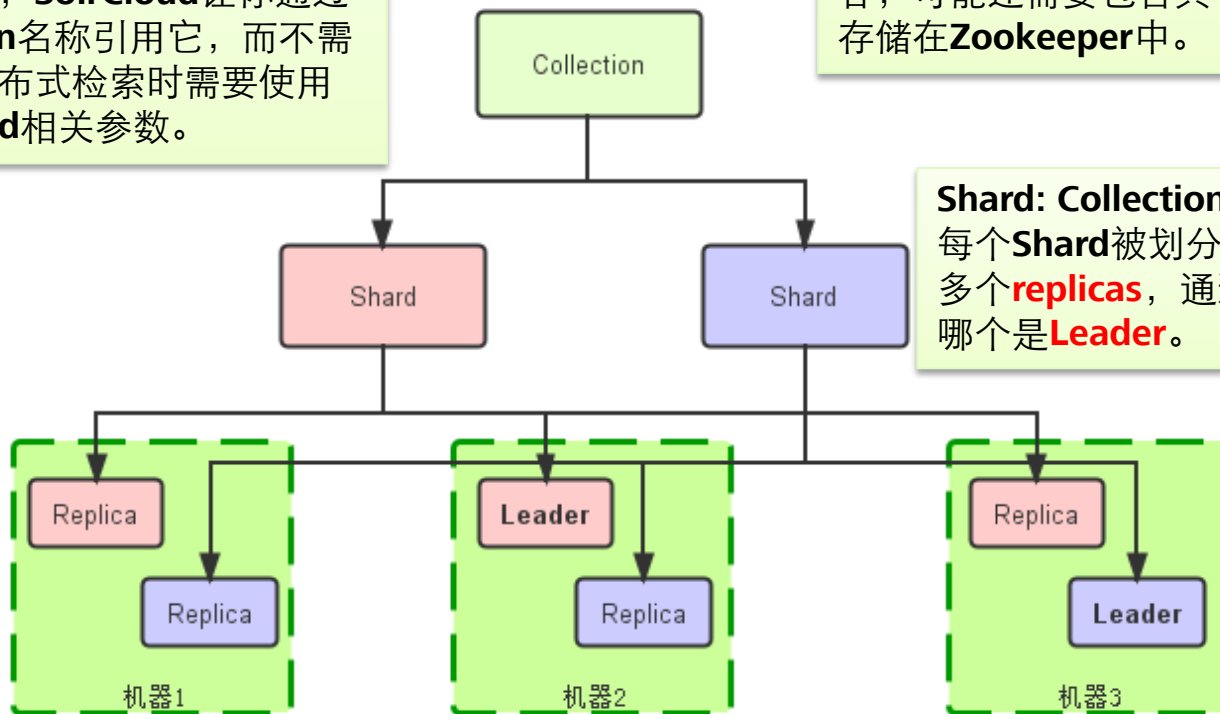
Replica: Shard的一个拷贝。

Leader: 赢得选举的Shard replicas。每个Shard有多个Replicas，这几个Replicas需要选举来确定一个Leader。

Solr概念体系—逻辑视图

Collection: 在SolrCloud中逻辑意义上的完整的索引。它常常被划分为一个或多个**Shard**，它们使用相同的**Config Set**。如果**Shard**数超过一个，它就是分布式索引，**SolrCloud**让你通过**Collection**名称引用它，而不需要关心分布式检索时需要使用的和**Shard**相关参数。

Config Set: Solr Core提供服务必须的一组配置文件。每个**config set**有一个名字。最小需要包括**solrconfig.xml** 和**schema.xml**，除此之外，依据这两个文件的配置内容，可能还需要包含其它文件。它存储在**Zookeeper**中。



Shard: Collection的逻辑分片。每个**Shard**被划分成一个或者多个**replicas**，通过选举确定哪个是**Leader**。

Solr概念体系—Core

coll_example_s...

Overview

Analysis

Dataimport

Documents

Files

Ping

Plugins / Stats

Query

Replication

Schema Browser

Segments info

Statistics

Last Modified: about an hour ago

Num Docs: 44142

Max Doc: 44142

Heap Memory Usage: -1

Deleted Docs: 0

Version: 58

Segment Count: 5

Optimized:

Current:

Replication (Master)

	Version	Gen	Size
Master (Searching)	1462349895171	4	8.93 MB
Master (Replicable)	1462349910205	5	-

Instance

CWD: /opt/huawei/Bigdata/nodeagent

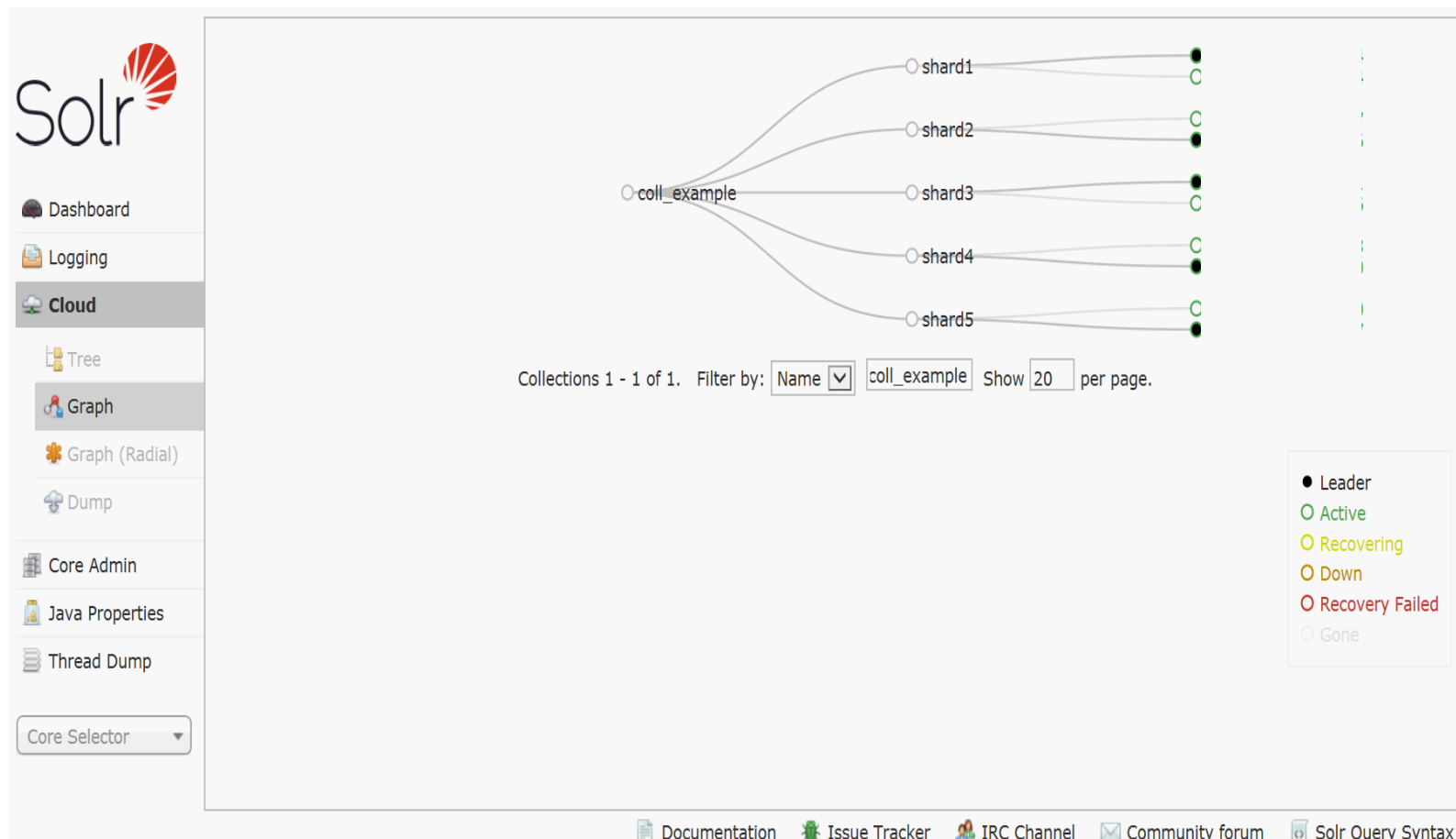
Instance: /srv/BigData/solr/solrserver5/coll_example_shard2_replica1

Data: /srv/BigData/solr/solrserver5/coll_example_shard2_replica1/data

Index: /srv/BigData/solr/solrserver5/coll_example_shard2_replica1/data/index

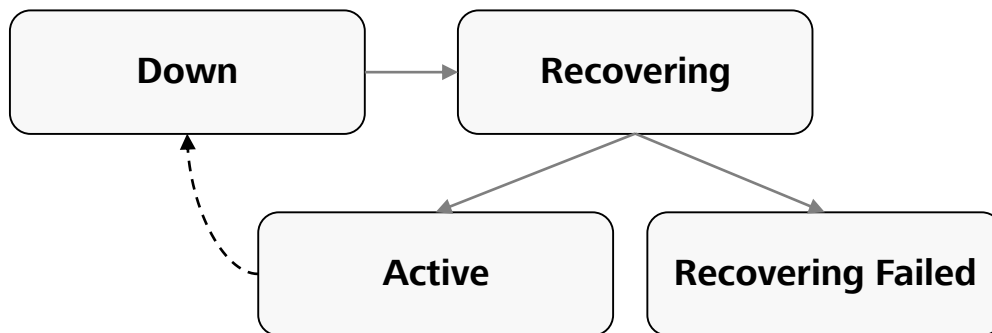
Impl: org.apache.solr.core.NIOFSDirectoryFactory

Solr概念体系—Collection



Solr概念体系—Replica状态

- **Active**: 处于该状态下可接受更新和查询请求。
- **Recovering**:
 - 当备Replica从主Replica同步时, 备Replica处于该状态
 - 新建的Replica从主Replica完全复制时, 此Replica处于该状态
- **Down**: **Recovering**的前一个状态
 - Solr Core启动时, Replica的初始状态
 - Solr Core正常关闭时
- **Recovery Failed**, **Recovery**失败。





目录

1. 什么是Solr
2. 应用场景
3. 概念体系
4. 系统架构
5. 关键特性
6. 华为增强特性

Solr架构

- 通过**HTTP Request Servlet**提供搜索相关的**RESTful**接口。
- 通过**Update Servlet**提供索引相关的**RESTful**接口。
- 所有功能都是以组件方式提供。
- 底层基于**Lucene**，保持**Lucene**绝对的独立性，**Lucene/Solr**同步发布版本。

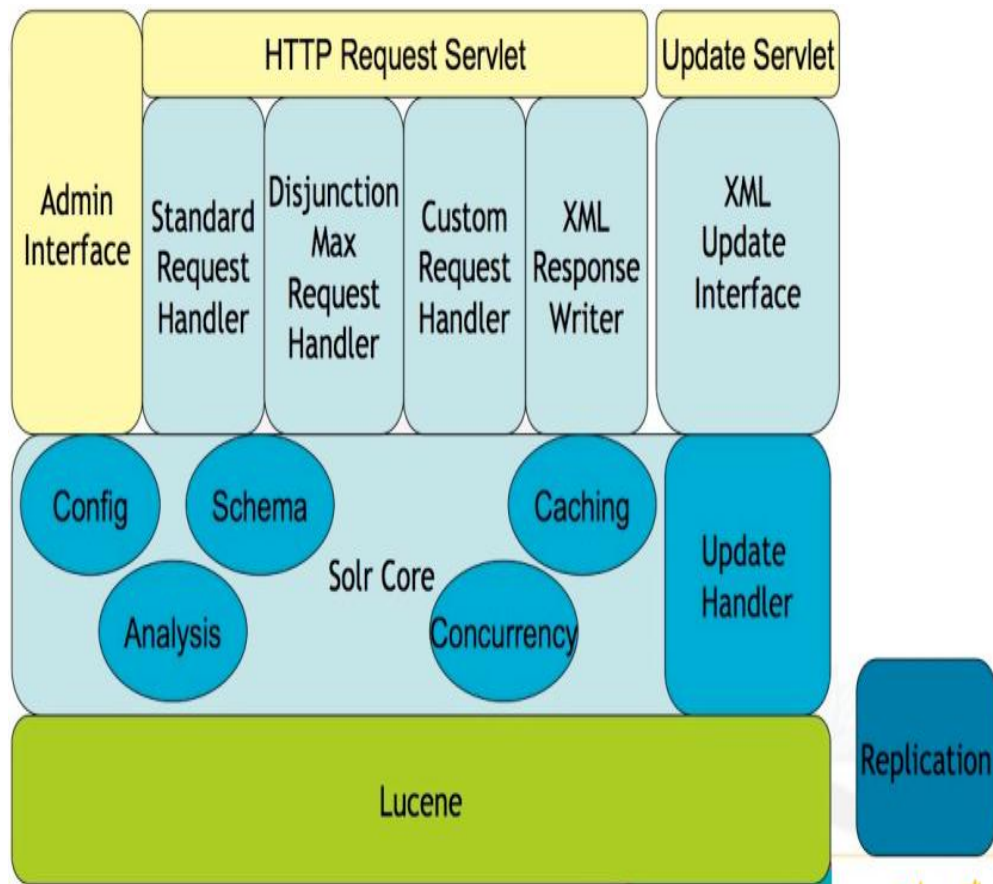
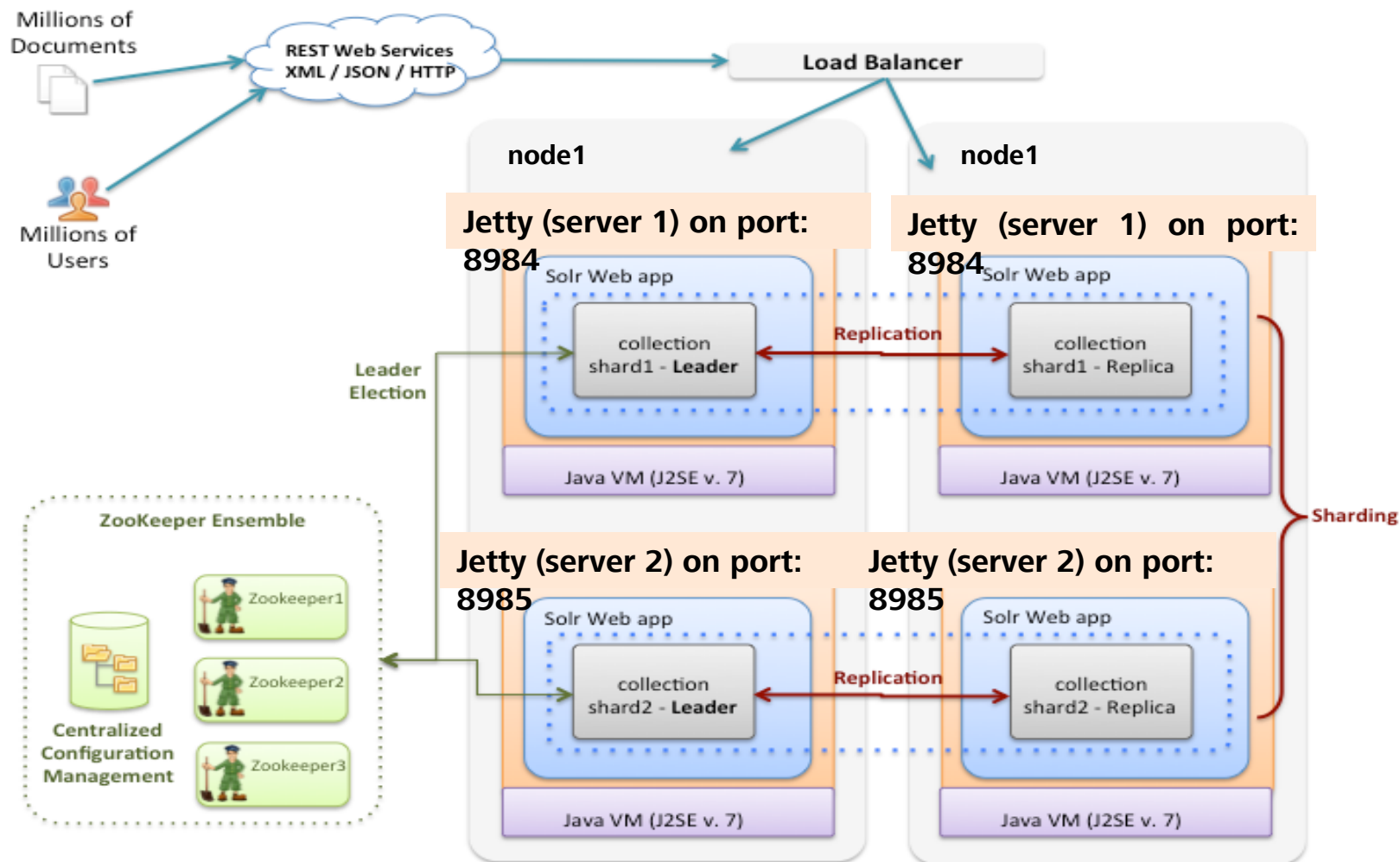


Diagram by Yonik Seeley



SolrCloud架构



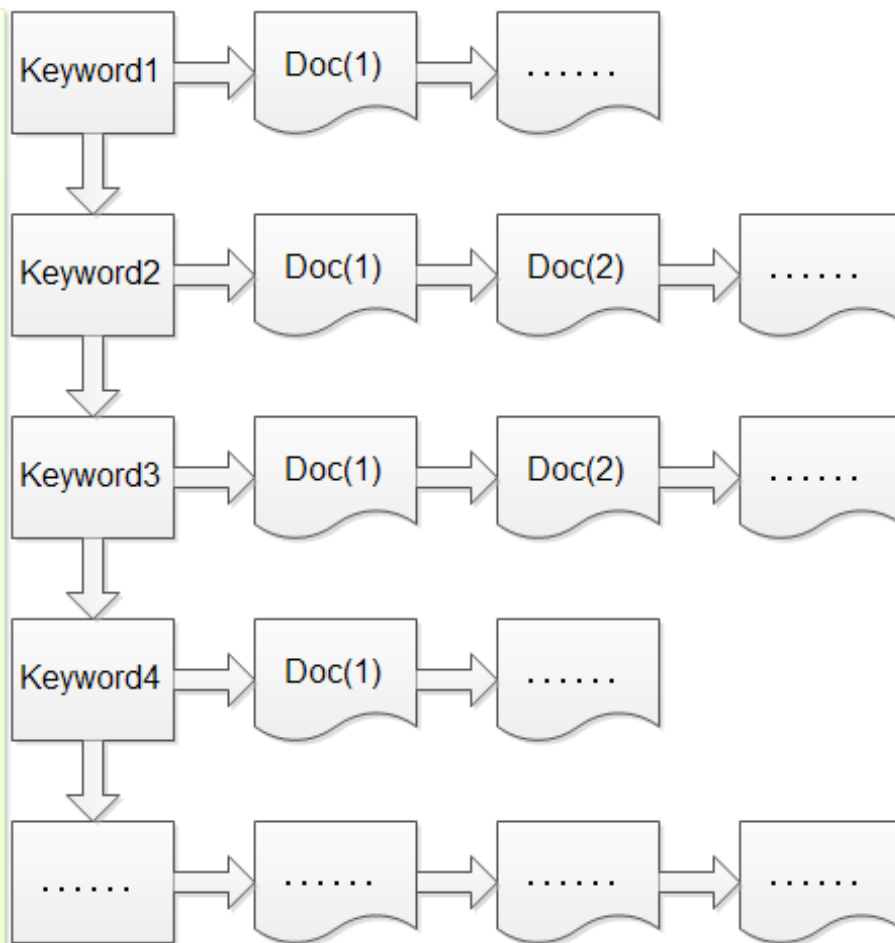


目录

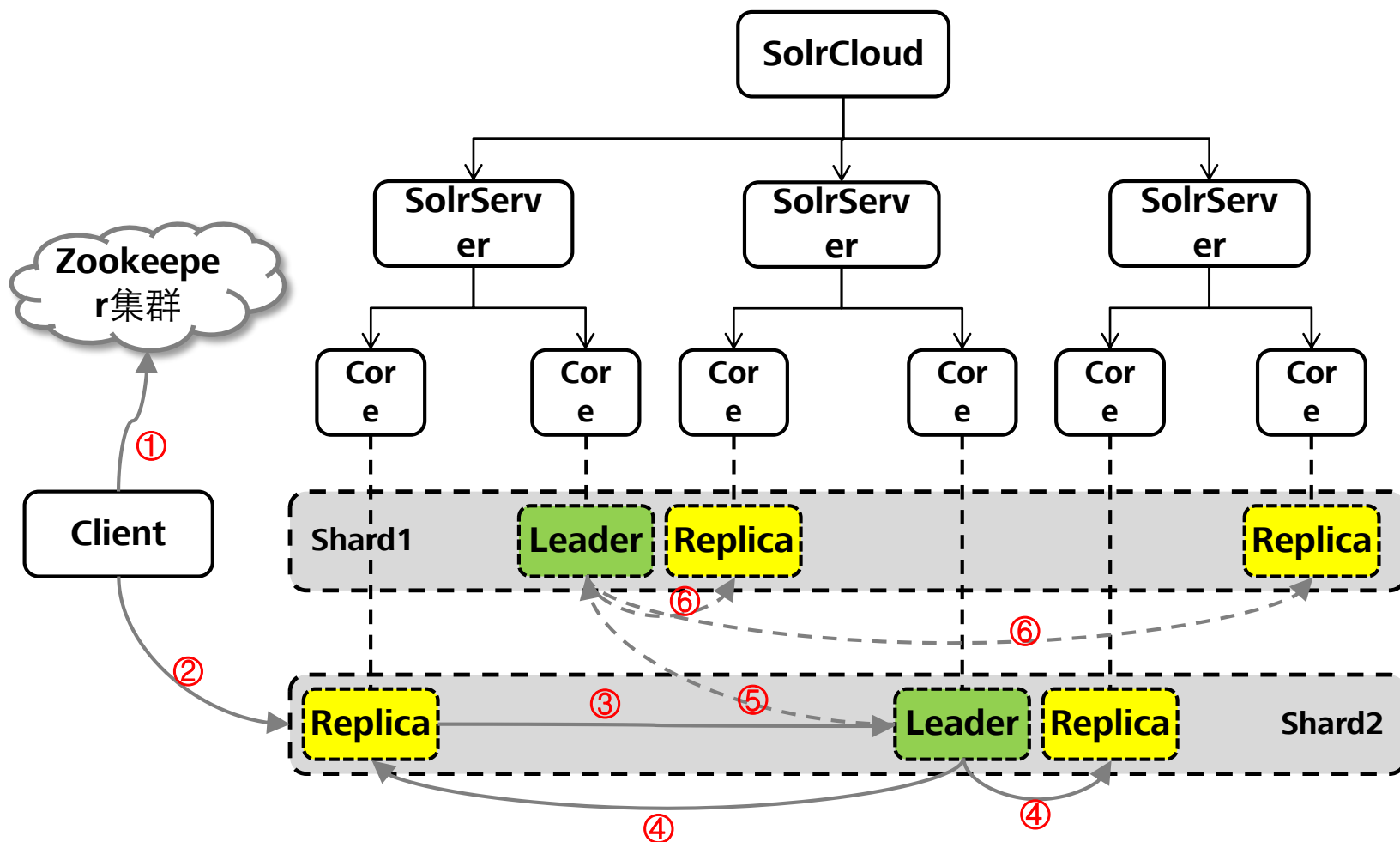
1. 什么是Solr
2. 应用场景
3. 概念体系
4. 系统架构
5. 关键特性
6. 华为增强特性

SolrCloud倒排索引

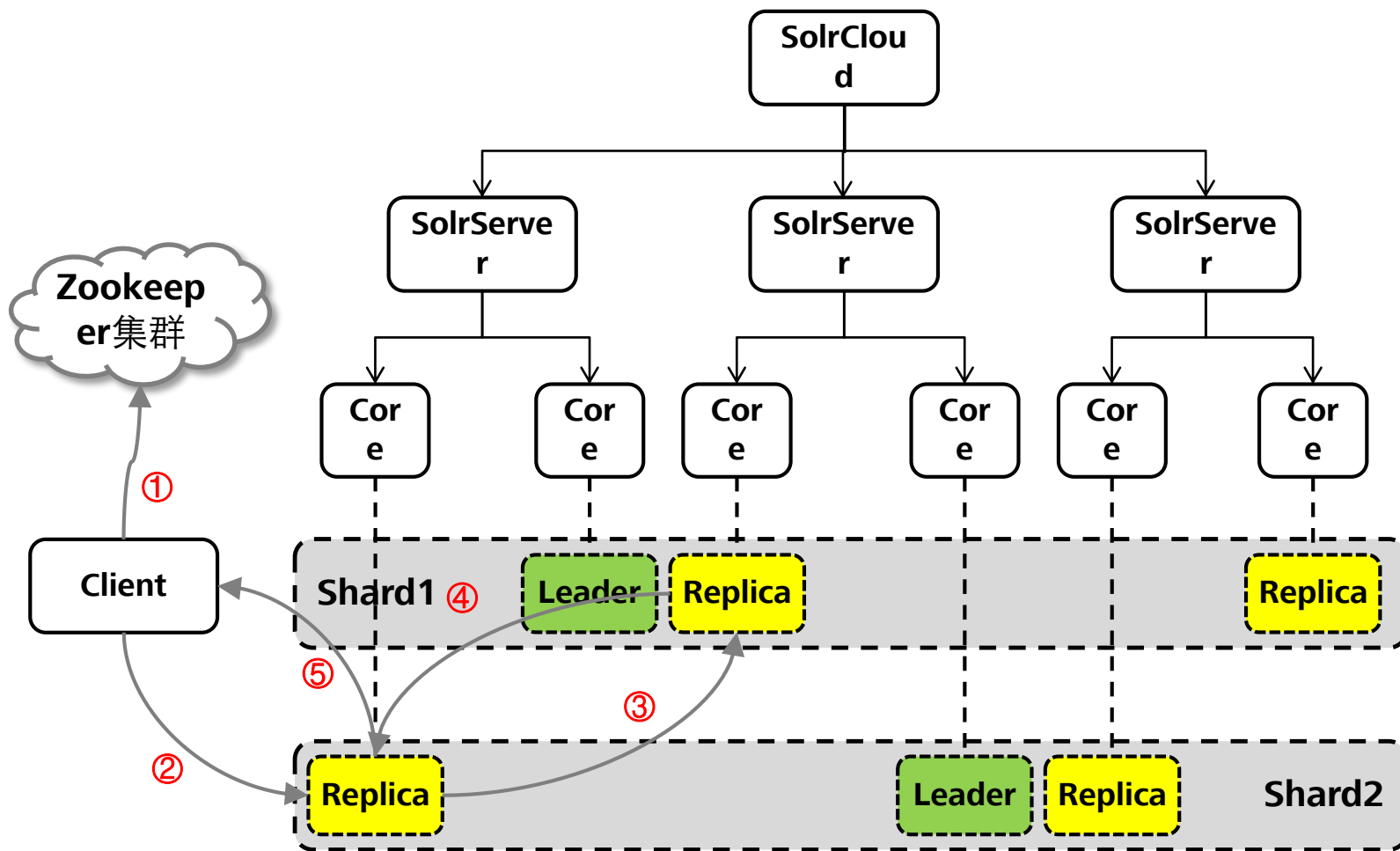
- 传统的搜索方式（正排序索引）是从关键点出发，然后再通过关键点找到关键点代表的信息中能够满足搜索条件的特定信息，即通过**KEY**寻找**VALUE**。通过正排序索引进行搜索，就是从通过文档编号找关键词。
- Solr (Lucene)** 的搜索采用了倒排序索引的方式，即通过**VALUE**找**KEY**。而在中文全文搜索中**VALUE**就是我们要搜索的关键词，存放所有关键词的地方叫词典。**KEY**是文档标号列表（通过文档标号列表我们可以找到出现过要搜索关键词--**VALUE**的文档），具体如下面的图所示：通过倒排索引进行搜索，就是通过关键词查询对应的文档编号，再通过文档编号找文档，类似于查字典，或通过查书目录查指定页码书的内容。



SolrCloud分布式索引流程



SolrCloud分布式搜索流程



SolrCloud路由算法

- **SolrCloud**中提供了两种路由算法，需要在创建**Collection**时指定。
不同的路由算法决定了索引数据的分发。
 - **compositeld Router**：一致性哈希路由，**shards**的哈希范围：
80000000~7fffffff。初始创建**collection**时必须指定**numShards**，**compositeld**路由算法根据**numShards**的个数，计算出每个**shard**的哈希范围，因此路由策略不可以扩展**shard**。
 - **implicit Router**：该路由方式，需要索引时指定每个文档具体路由到哪个**Shard**，这与**compositeld Router**方式（索引可均匀分布在每个**shard**上）不同。同时只有在**implicit Router**下才可创建**shard**。

SolrCloud 扩容

- 扩容场景：
 - 物理资源消耗过大，即**Solr**的服务节点的**CPU**、内存占用率过高、磁盘空间不足
 - **Solr**单实例的索引数据太大，索引的数目达到**10**亿条或者是数据大小达到**1T**
- 扩容方式：
 - 增加**SolrServer**实例
 - 增加节点，在新节点上增加**SolrServer**实例
- 扩容后，采用**implicit Router**的**Collection**数据迁移步骤：
 - **Create shard Rest API**：
/solr/admin/collections?action=CREATESHARD&shard=shard1&collection=test1&createNodeSet= SolrIp:SolrPort _solr
 - **collection=test1**：指定需要进行添加分片的**Collection**名称为**test1**
 - **shard=shard1**：指定需要增加**shard**名称为**shard1**，注意不要与已有的**shard**同名
 - 多副本情况下，**createNodeSet**指定多个（副本个数）“**SolrIp: SolrPort _solr**”，以逗号分隔，可以使其副本创建到不同主机上，避免**存在单点故障风险**。

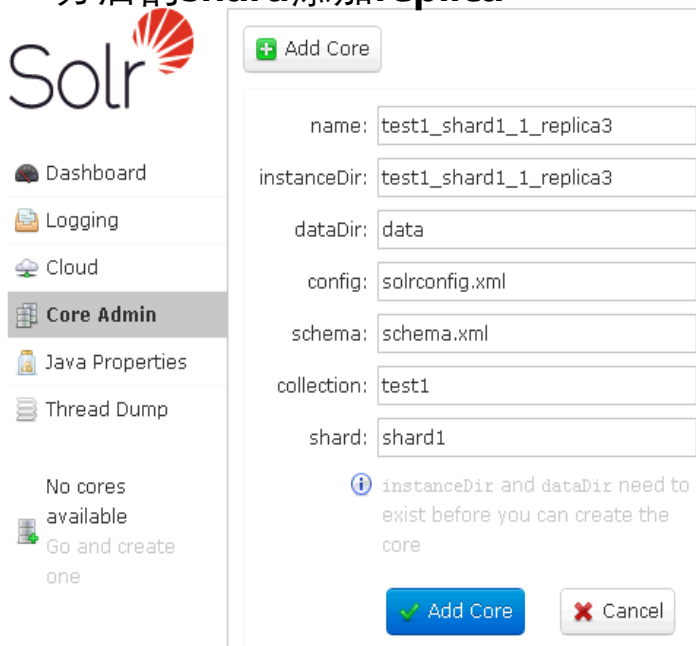
SolrCloud 扩容

- 扩容后，采用 **compositield Router** 的 **Collection** 数据迁移步骤：

1.Split shard:

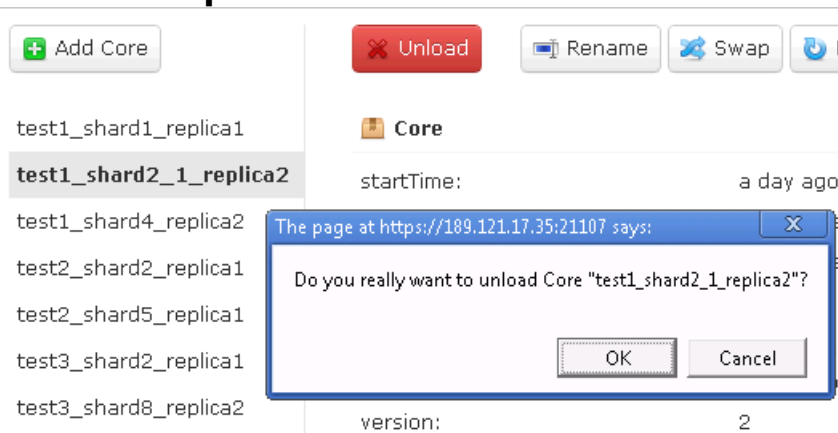
/solr/admin/collections?action=SPLITSHARD&shard=shard1&collection=test1

2. 在新增加的Solr实例上，为拆分后的shard添加replica



Solr Admin UI Screenshot showing the 'Add Core' form. The form fields are: name: test1_shard1_1_replica3, instanceDir: test1_shard1_1_replica3, dataDir: data, config: solrconfig.xml, schema: schema.xml, collection: test1, shard: shard1. A message states: 'instanceDir and dataDir need to exist before you can create the core'. Buttons: Add Core, Cancel.

3. 删除要减少数据的replica, unload replica



Solr Admin UI Screenshot showing the 'Unload' button and a list of cores. The 'Unload' button is highlighted. The list of cores includes: test1_shard1_replica1, test1_shard2_1_replica2, test1_shard4_replica2, test2_shard2_replica1, test2_shard5_replica1, test3_shard2_replica1, test3_shard8_replica2. A modal dialog box is open asking: 'Do you really want to unload Core "test1_shard2_1_replica2"?'. Buttons: OK, Cancel.

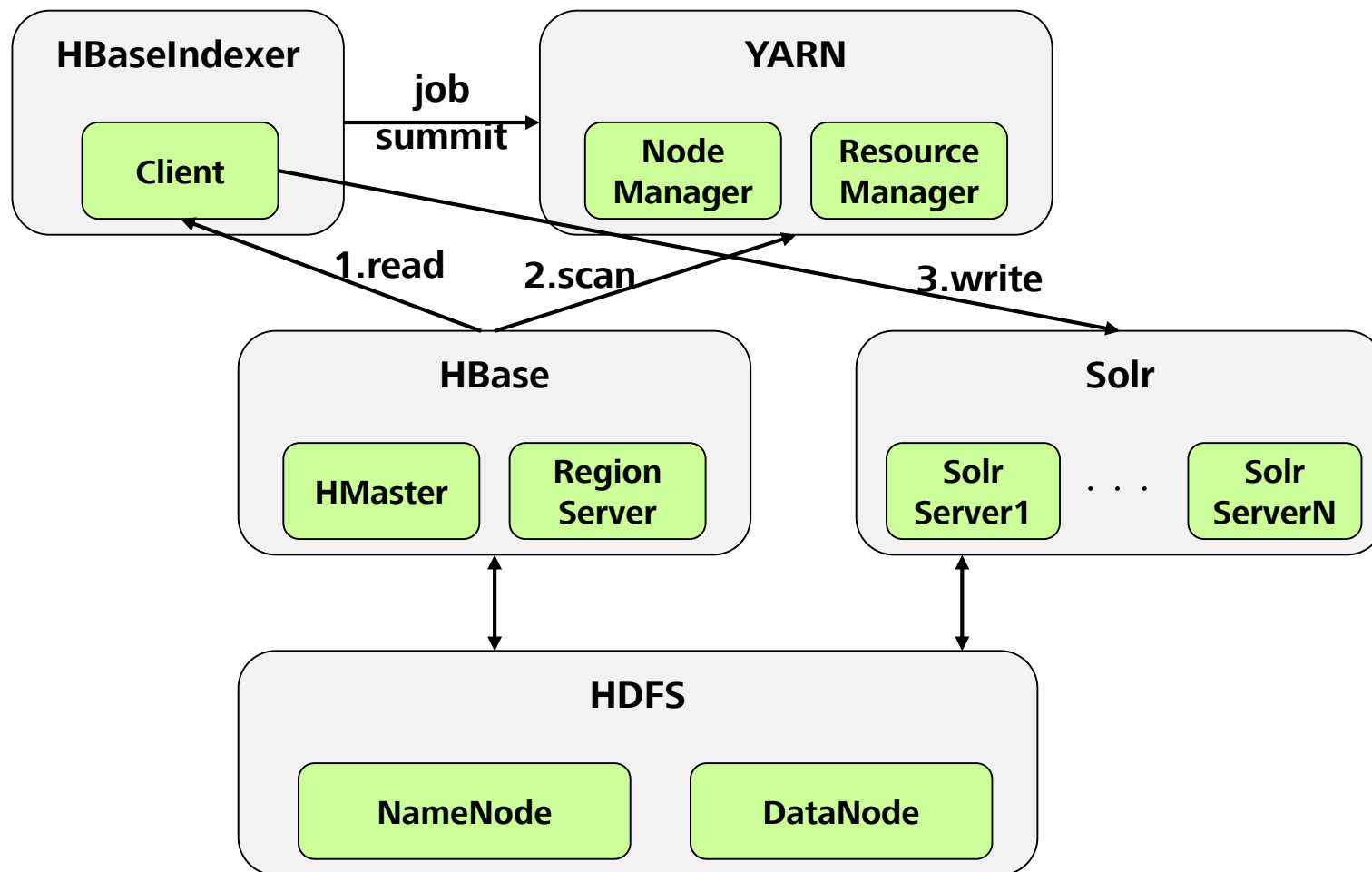
SolrCloud 减容

- 减容场景：
 - 节点需要重新安装操作系统
 - 当Solr创建的shard下的replica过多，需要减少replica数目
- 减容方式：
 - 在FusionInsight Manager上删除solr实例
- 减容注意事项：
 - 确保要删除的实例上的Core所属的shard下的replica在其他的实例存在

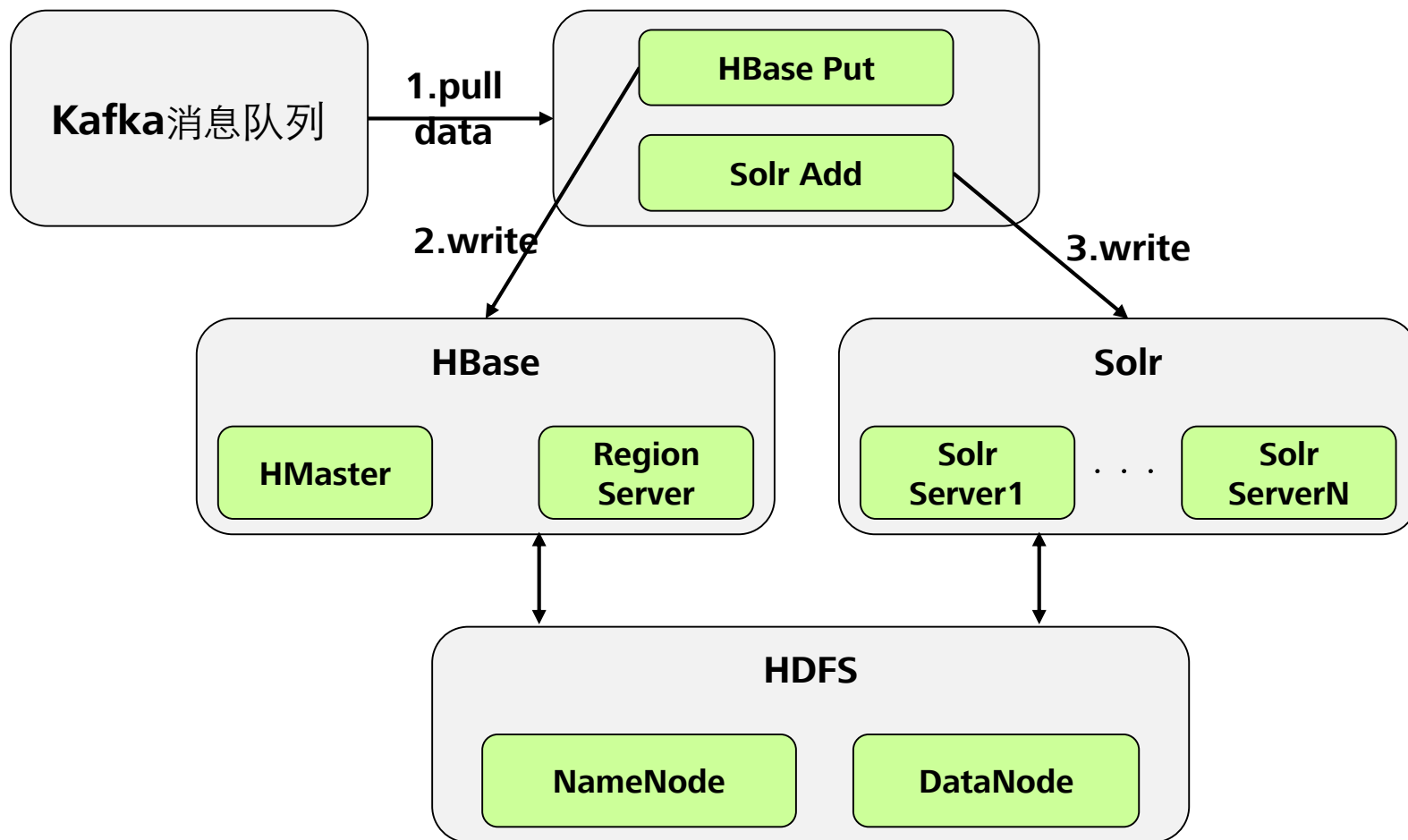
Solr索引HBase数据

- **Solr索引HBase数据**是将**HBase**数据写到**HDFS**的同时，**Solr**建立相应的**HBase**索引数据。其中索引**id**与**HBase**数据的**rowkey**对应，保证每条索引数据与**HBase**数据的唯一，实现**HBase**数据的全文检索。
 - 批量索引：针对**HBase**中已有的数据，通过提交**MR**任务的形式，将**HBase**中的全部数据读出，然后在**Solr**中建立索引
 - 实时索引：使用**HBase Client API + Solrj API**对**HBase**数据进行实时索引

Solr对HBase数据批量索引



Solr对HBase数据实时索引



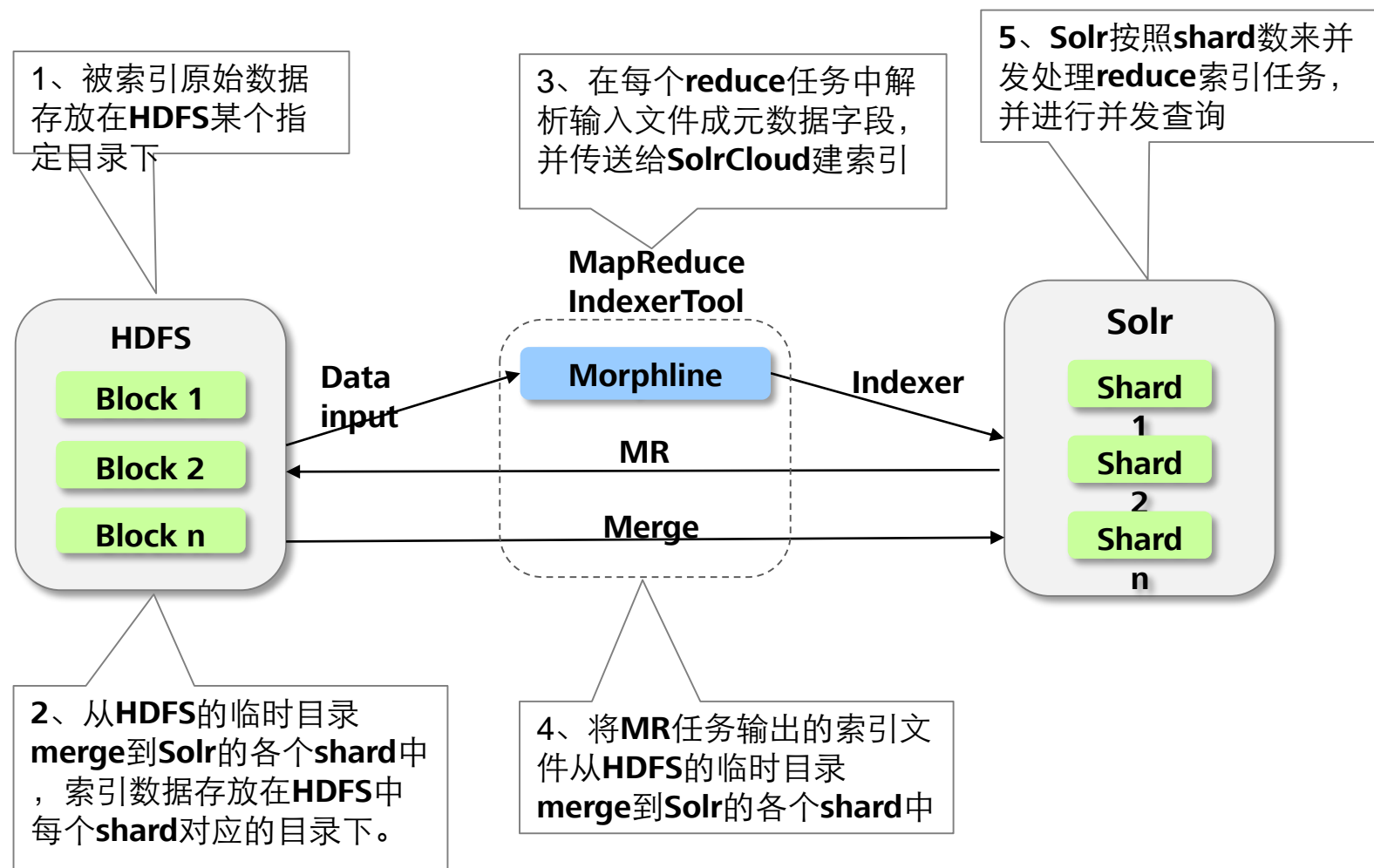
Solr索引HDFS数据

- **Solr over HDFS**主要利用**MapReduce**框架读取**HDFS**上的各种非结构化数据（**doc**、**xml**、**txt**、**csv**等），在**SolrCloud**中创建索引文件，同时将索引文件存储在**HDFS**上。

特点：对于大量非结构化数据的全文检索，创建索引速度，处理数据量大，索引存储量大。

- 在**HDFS**上已有大量数据情况下，可以使用**HDFSIndexer**来批量索引。**HDFS Indexer**通过创建**MR**任务，扫描**HDFS**上指定目录下的结构化文件，采用**Merge index**文件方法向**Solr**建立索引。

Solr对HDFS数据批量索引





目录

1. 什么是Solr
2. 应用场景
3. 概念体系
4. 系统架构
5. 关键特性
6. 华为增强特性

索引元数据自动恢复

- 问题：

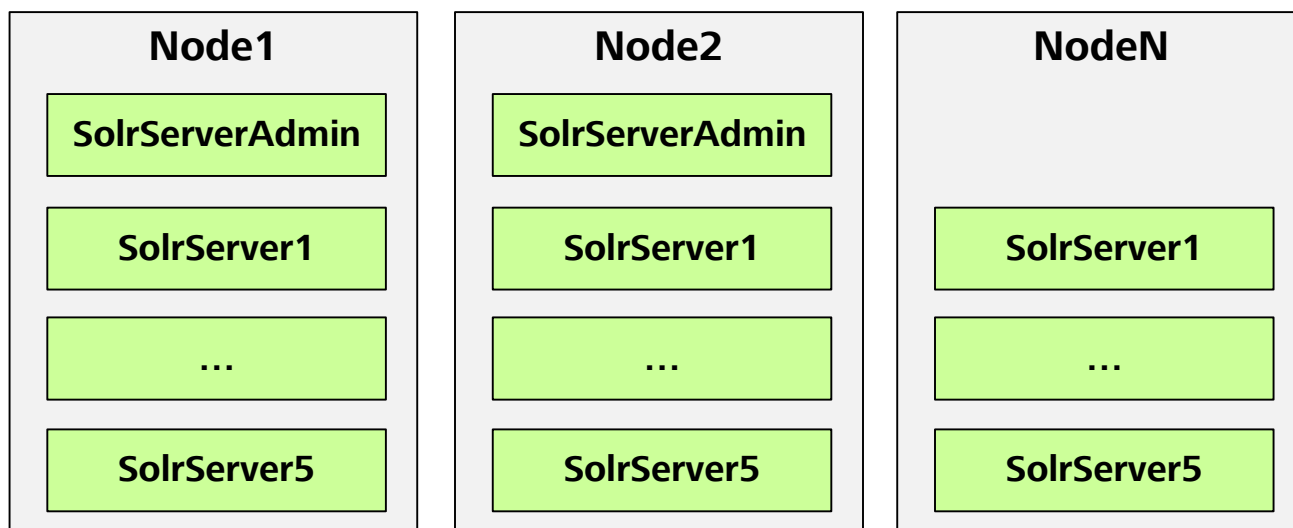
开源**Solr**的索引元数据存储在本地图盘，并且只有一份，当元数据丢失后（如磁盘故障），**Solr**服务无法自动恢复。

- 增强：

自动从**ZooKeeper**中恢复元数据，通过副本机制恢复索引数据。

支持单节点多实例部署

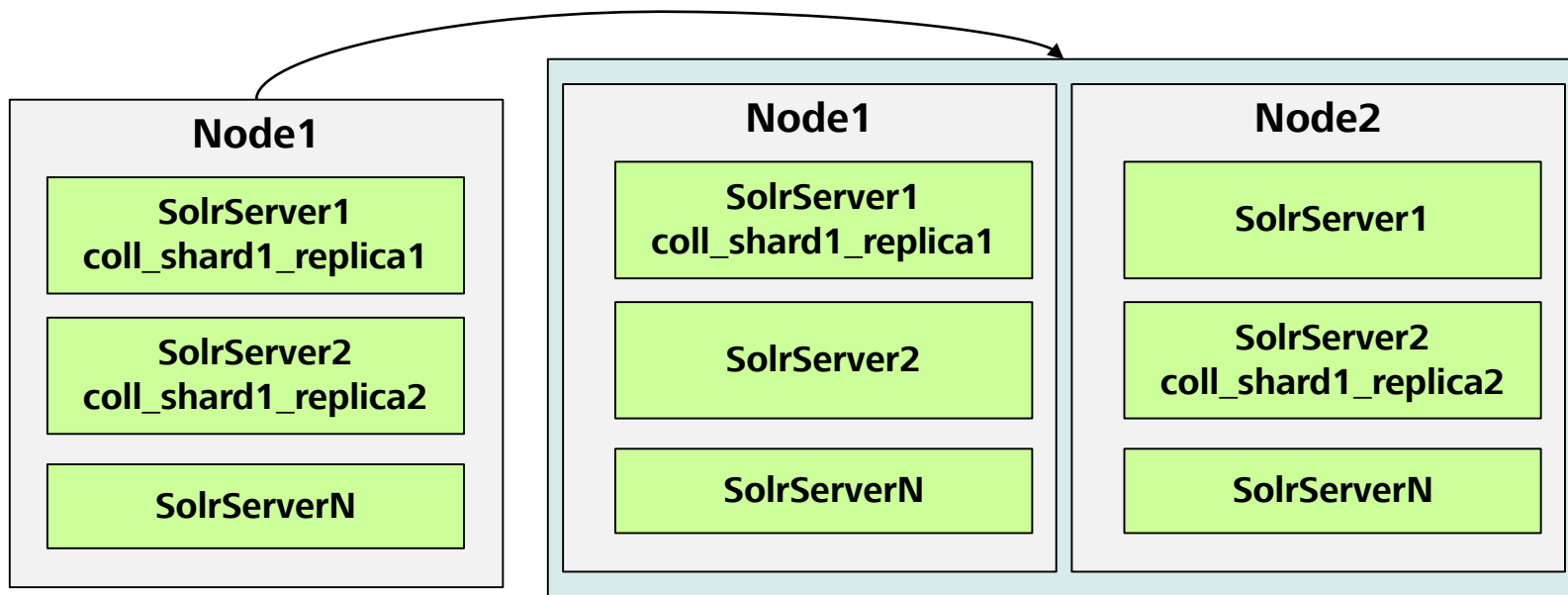
- 在同一个节点上部署多个**Solr**实例，根据**IP**和不同的端口号来区分不同的**Solr**实例
- 提高单节点**CPU**、内存和磁盘的利用率，同时提高**SolrCloud**的索引和搜索能力



副本自动跨节点分配策略

问题：单节点多实例部署下，多副本时，只做到了跨实例分配，存在单点故障。

增强：创建多副本**Collection**时，自动将副本分配到不同节点上。



习题

1. **Solr**组件的**SolrCloud**模式强依赖于哪个组件（ ）。

A. HBase B. HDFS C. ZooKeeper D. YARN

2. **Solr**底层基于哪个开源软件进行开发的（ ）。

A. MySQL B. MongoDB C. Memcached D. Lucence

习题

3. **Solr**可索引哪些数据类型（ ）。

- A. 结构化数据 B. 非结构化数据
- C. 半结构化数据 D. 以上都是

4.请解释**Solr**组件中常见的术语的含义，什么是**collection**、**shard**、**replica**、**core**、**leader**？。

思考题

1、请思考一下**Solr**的基本功能是什么？

Thank you

www.huawei.com