大数据开发技术

东北林业大学

卢洋

第二章

Hadoop序列化

1. 序列化概念;
2. 自定义bean对象实现序列化接口(Writalbe);
3. 序列化案列.

2.2

自定义bean对象实现序列化接口(Writable)

- 在开发中，往往常用的序列化类型不能满足所有需求；

- 比如：在Hadoop内部要传递一个bean对象，那么就需要实现序列化接口.

Java类型	Hadoop Writable 类型
boolean	BooleanWritable
byte	ByteWritable
int	IntWritable
float	FloatWritable
long	LongWritable
double	DoubleWritable
String	Text
map	MapWritable
array	ArrayWritable

实现bean对象序列化需要7步

(1) 必须实现Writable接口;

(2) 反序列化时，需要反射调用空参构造函数，所以必须有空参构造:

```
public FlowBean() {
    super();
}
```

(3) 重写序列化方法;

(4) 重写反序列化方法;

实现bean对象序列化需要7步

(3) 重写序列化方法:

```
@override
public void write(DataOutput out) throws IOException {
    out.writeLong(upFlow);
    out.writeLong(downFlow);
    out.writeLong(sumFlow);
}
```

保证写的顺序一直一样

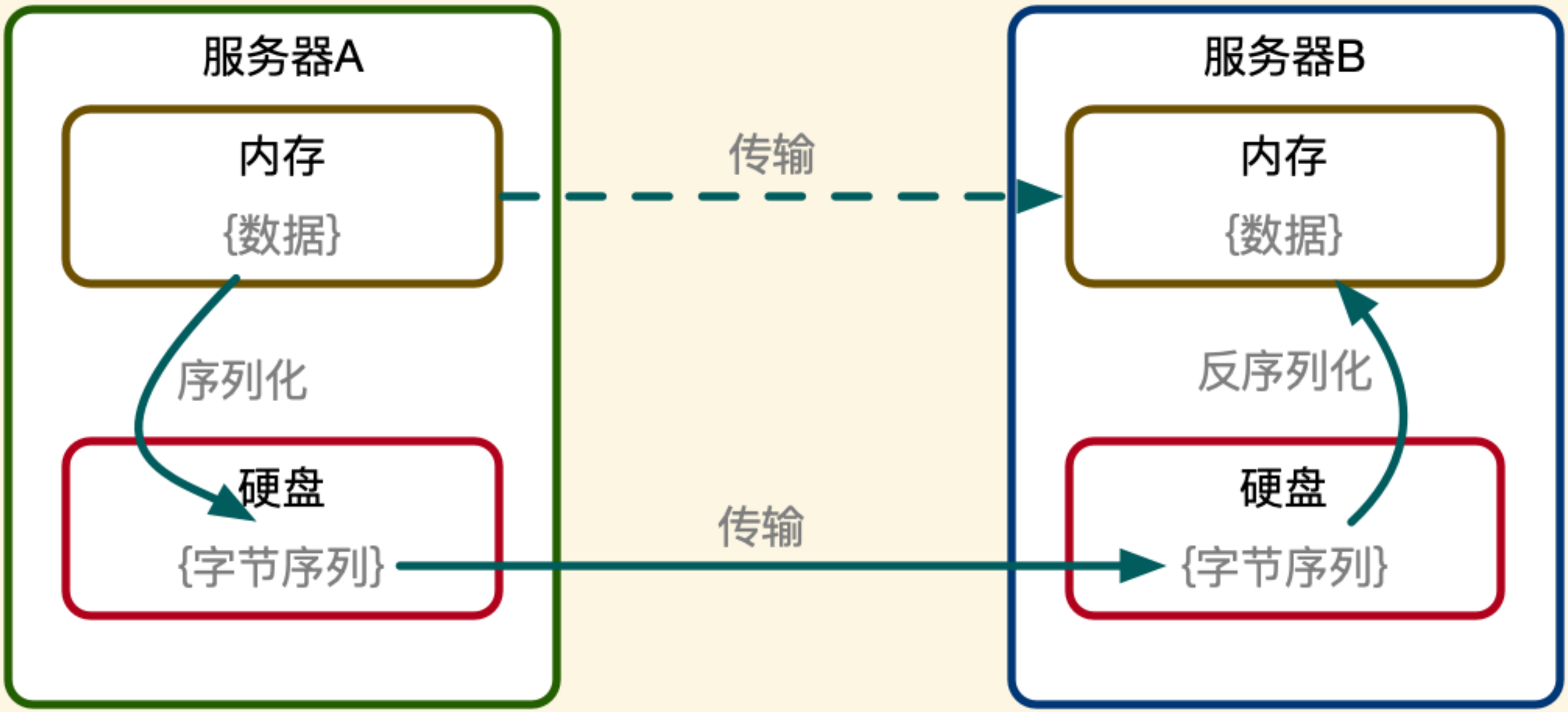
(4) 重写反序列化方法:

```
@override
public void readFields(DataInput in) throws IOException {
    upFlow = in.readLong();
    downFlow = in.readLong();
    sumFlow = in.readLong();
}
```

实现bean对象序列化需要7步

(5) 注意：序列化的顺序和反序列化的顺序完全一致；
先进先出，队列；

(6) 要想把结果显示在文件中，需要重写toString()，可用"\t"分开，方便后续调用；


服务器A
内存
{数据}
序列化
硬盘
{字节序列}
传输
传输
服务器B
内存
{数据}
反序列化
硬盘
{字节序列}

实现bean对象序列化需要7步

(7) 如果需要将自定义的bean放在key中传输，则还需要实现Comparable接口，MapReduce框架中的Shuffle过程要求对key必须能排序.

```
@override
public int compareTo(FlowBean o) {
    // 倒序排列，从大到小
    return this.sumFlow > o.getsumFlow() ? -1 : 1;
}
```