

目标

flink弹性扩缩容

场景: 不同时间段使用资源不一样、不同租户使用的资源不一样

对研发的效果: 通过扩容底层资源实现优化算法一样的效果, 为优化算法缓冲时间

版本: 1.14

资源

tm数量(tm 副本数)

slot数量(tm数量 x 配置的slot数量)

可用并行度=slot数量

基于taskmanager和jobmanager的资源使用情况使用hpa+ca进行扩容副本数

并行度

优先级: 算子 > env > client > server

参考文档: <https://nightlies.apache.org/flink/flink-docs-master/zh/docs/dev/datastream/execution/parallel/>

算子

```
final StreamExecutionEnvironment env =
    StreamExecutionEnvironment.getExecutionEnvironment();

DataStream<String> text = [...];
DataStream<Tuple2<String, Integer>> wordCounts = text
    .flatMap(new LineSplitter())
    .keyBy(value -> value.f0)
    .window(TumblingEventTimeWindows.of(Time.seconds(5)))
    .sum(1).setParallelism(5);

wordCounts.print();

env.execute("word Count Example");
```

执行环境

```
final StreamExecutionEnvironment env =
StreamExecutionEnvironment.getExecutionEnvironment();
env.setParallelism(3);

DataStream<String> text = [...];
DataStream<Tuple2<String, Integer>> wordCounts = [...];
wordCounts.print();

env.execute("word Count Example");
```

客户端(暂略)

```
./bin/flink run -p 10 ../examples/*wordCount-java*.jar
```

```
try {
    PackagedProgram program = new PackagedProgram(file, args);
    InetAddress jobManagerAddress =
RemoteExecutor.getInetFromHostport("localhost:6123");
    Configuration config = new Configuration();

    Client client = new Client(jobManagerAddress, config,
program.getUserCodeClassLoader());

    // set the parallelism to 10 here
    client.run(program, 10, true);

} catch (ProgramInvocationException e) {
    e.printStackTrace();
}
```

服务端

```
cat flink-conf.yaml |grep "parallelism.default"
```

最大并行度(?)

> 128, < 32768

所以并行度最大值 21845, 需要观察cp/sp时间再降低

客户端通过处理延迟发现运行时过低

假设 最大允许堆积时间 < 10s

mq堆积数量/mq处理速度 > 最大允许堆积时间

客户端调整运行时并发度

时间窗口为1m

mq消费数量

mq生产数量

期望并行度=生产数量/消费数量*当前并行度

不能频繁调整并发度(时间窗口大于1m)

source的并发度应该始终保持为1

服务端发现slot过低

$\text{JobManager.taskSlotsAvailable} / \text{taskSlotsTotal} < 0.2$

调整tm的副本数量, 一次性增加20%副本数

问题

1.19-snapshot中的reactive讲了什么?

在taskmanager扩容之后会调整job的并行度

但是在深度使用flink场景中, job的并行度都是由算子设置的, 所以可能意义不大

行动项

整理思路

部署k8s, p8s, flink

编写wordcount

后续

记忆图谱