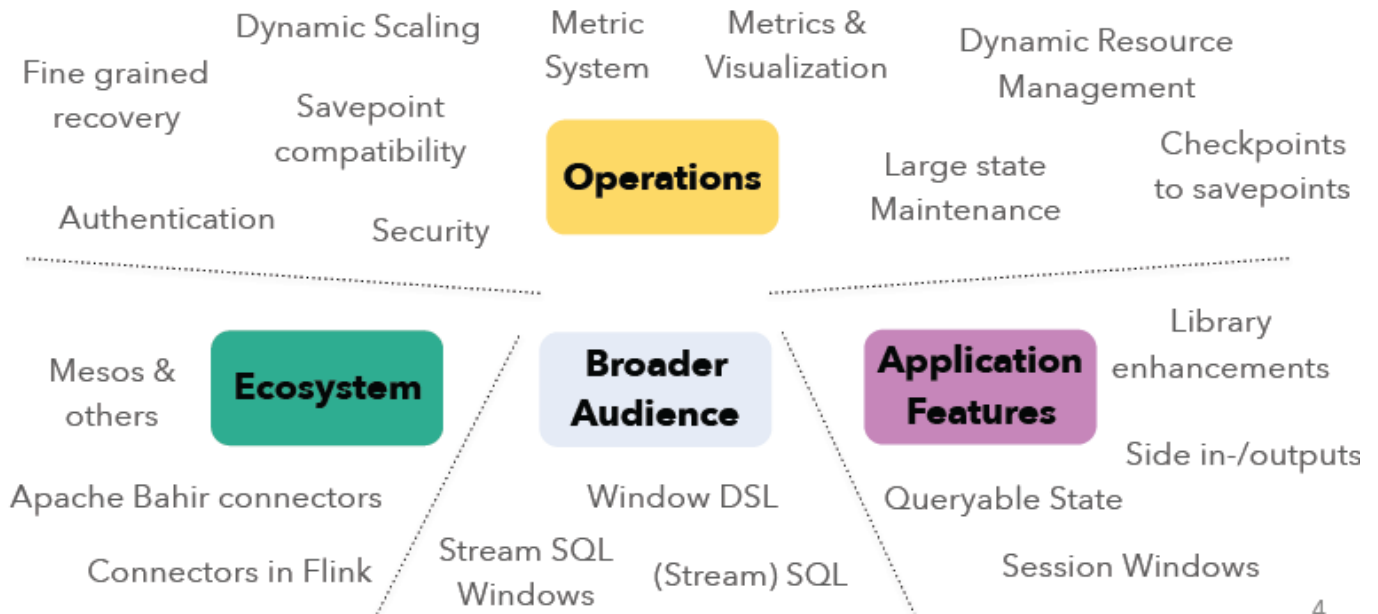


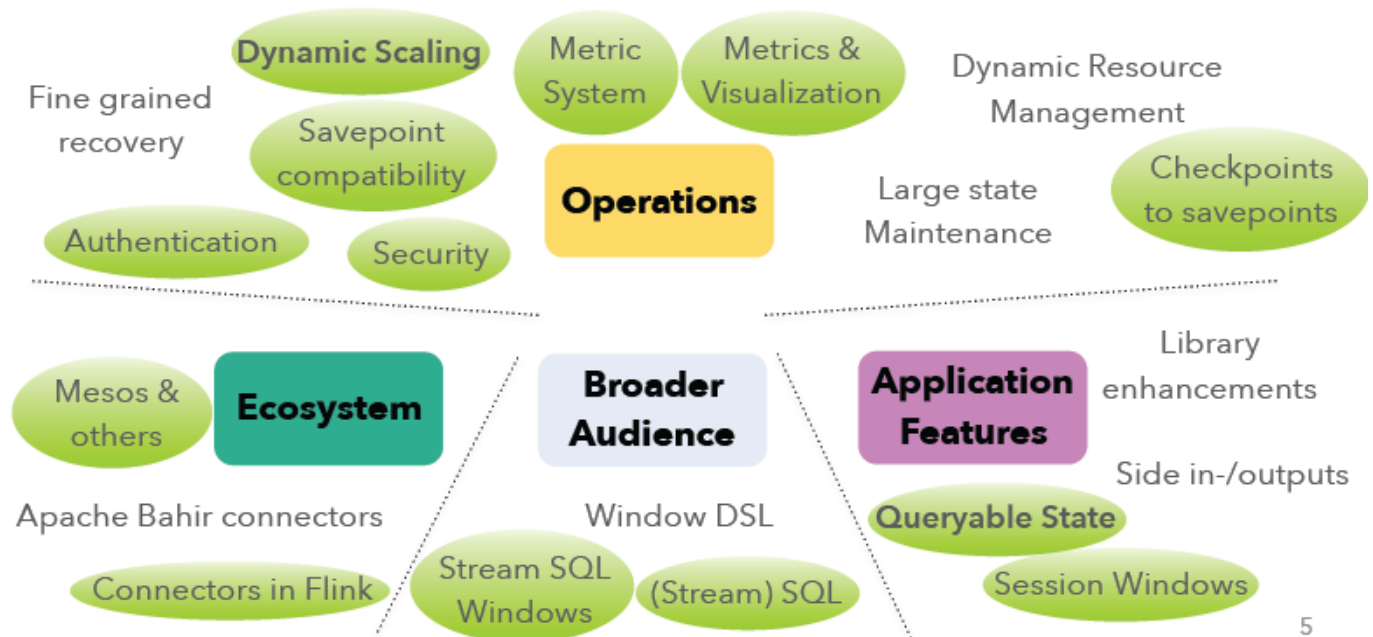
Apache Flink 1.2.0新功能概述

本文将概述即将发布的Apache Flink 1.2.0新功能。在Apache Flink 1.1+版本上，社区主要的集中点在操作性（Operations）、生态系统（Ecosystem）、更广泛的用户（Broader Audience）以及应用特性（Application Features）等方面的开发。各个模块的开发主要包括了如下的方向：



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

而Flink 1.2版本对以下的方面进行了提升，其中动态扩展（Dynamic Scaling）和可查询状态（Queryable State）又是本版本的重中之重。后面我将单独写一篇文章介绍Queryable State的设计目的。



如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

动态扩展（Dynamic Scaling）和可查询状态（Queryable State）

这个算得上是Apache Flink 1.2的重要特性了，后面单独开文章介绍，请关注。

安全以及身份认证（Security / Authentication）

大数据框架的安全一直是一个比较头疼的问题，而在Apache Flink 1.2中，社区对Flink的安全以及身份认证做了比较多的工作，此特性的开发主要由美国信息存储资讯科技公司-易安信(EMC)贡献，主要包括：

- 1、数据访问授权，主要基于Kerberos实现；
- 2、Flink进程之间的通信数据进行加密，所有的通信（包括RPC, 数据交换, Web UI等等）都通过SSL进行连接；
- 3、此外，本版本还防止了恶意用户侵入到Flink作业（hook into Flink jobs）。

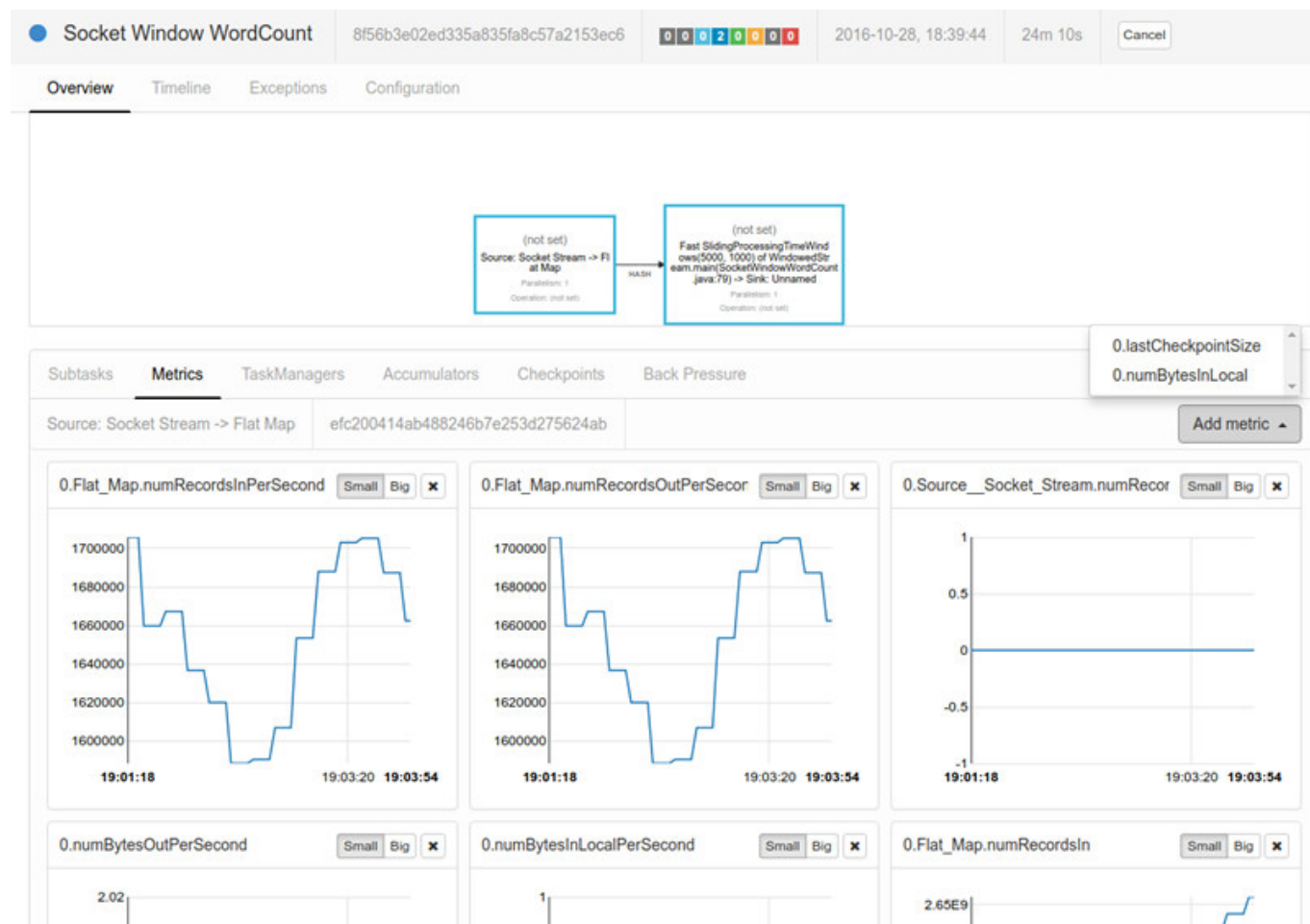
集群管理（Cluster Management）

大家应该知道，直到Apache Flink 1.1.x，内置支持的集群管理主要包括：Standalone和Flink on Yarn。但是我们也都知道，Apache Mesos也是一款很不错的开源分布式资源管理框架；虽然我们可以自己做一些修改能让Flink运行在Apache Mesos之上，但是怎么能够和内置支持方便性来比较呢？为此，社区在 FLINK-1984 引入了Flink on Mesos特性，我们得感谢EMC公司的贡献！

在将来（Apache Flink 1.2之后），阿里巴巴和 dataartisans 公司将联合努力为Flink提供与各种集群管理器无缝互操作的功能，比如Docker。

指标监控 (Metrics)

如果你使用过Apache Spark，你肯定知道Apache Spark的WEB UI提供了诸如 Input Rate、Scheduling Delay、Processing Time以及Total Delay的图形展示功能，极大方便了用户的使用和监控。不过别羡慕了，Apache Flink 1.2开始，内置也会在WEB UI监控的页面提供各种图形监控如下：



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

增强Savepoint 和 Checkpoint的健壮性

主要包括：

- 1、从检查点恢复作业；
- 2、使用较旧的检查点 (Checkpoint) 从失败中恢复；
- 3、忽略失败的Checkpoints；
- 4、向后兼容。

Table API 和 Stream SQL

在Table API 和 Stream SQL方面，Apache Flink 1.2主要提供了：

1、Group-windows

```
table.groupBy('iteblog')  
.window(Session withGap 10.minutes on 'rowtime')  
.select('uid', 'product.count')
```

2、更多SQL算子，包括：EXISTS, VALUES, LIMIT等；

3、更多内置的scalar functions，包括CURRENT_DATE, INITCAP, NULLIF等

4、更多的数据类型和更好的集成

```
pojo.get('field')  
pojo.flatten()
```

5、用户自定义的scalar functions：

```
table.select('uid',parseName('userJson'))
```

其他新功能

- 1、支持Kafka 0.10
- 2、Bucketing Sink: divides output into different file w.r.t. user logic
- 3、Detached execution: first step in programatically controlled job
- 4、异步IO操作：外部系统非阻塞查询
- 5、可扩展性，鲁棒性改进以及一些错误修复。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接：[【】（）](#)