

**《实时计算系统设计》**

**课程报告**

|  |  |
| --- | --- |
| 报告内容名称: | 第一次实时计算系统作业 |
| 院系: | 信息科学与工程学院 |
| 专业班级: | 计科1502班 |
| 学号: | 201507010206 |
| 姓名: | 黄郭斌 |
| 课程指导教师: | 安吉尧 教授 |

实验提交时间：2018-11-4

1. 实训实验环境

无

1. 实训实验内容与完成情况

### 1. 针对大数据的实时批量计算，了解与分析一种实时可靠的开源分布式实时计算系统——Storm

Twitter Strom是一个免费开源的分布式实时计算系统，Strom对于实时计算的意义类似于Hadoop对于批处理的意义，Strom可以简单，高效，可靠地处理流数据，并支持多种编程语言。

Strom框架可以方便地与数据库系统进行整合，从而开发出强大的实时计算系统。

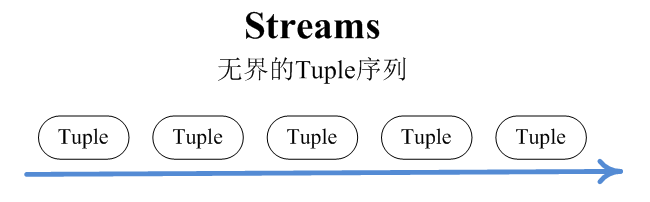
Strom可以用于许多领域中，如实时分析、在线机器学习计算、持续计算、远程RPC、数据ETL等。

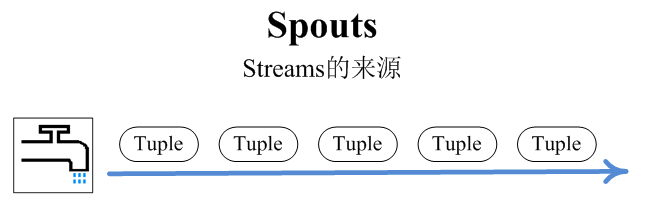
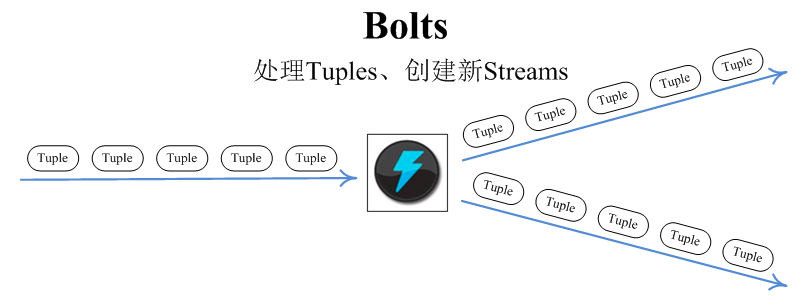
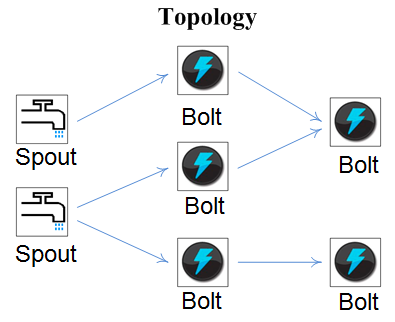
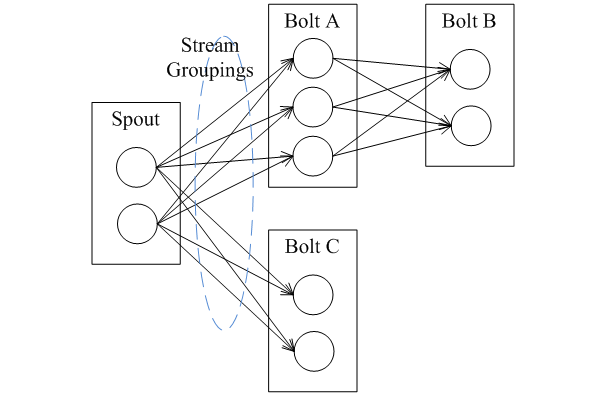
Strom具有以下特点：

* 整合性——与数据库整合
* 简易的API——开发Strom应用方便
* 可扩展性——计算任务可以分散到集群中的各个结点中
* 容错性——自动故障结点的重启，任务重新分配
* 可靠的消息处理——保证每个消息都能执行
* 支持各种编程语言——Python，JAVA等
* 快速部署——整个框架部署方便
* 免费开源

Storm主要术语包括Streams、Spouts、Bolts、Topology和Stream Groupings：

* **Streams**：Storm将流数据Stream描述成一个无限的Tuple序列，这些Tuple序列会以分布式的方式并行地创建和处理。



* **Spout**：Storm认为每个Stream都有一个源头，并把这个源头抽象为Spout
* 通常Spout会从外部数据源（队列、数据库等）读取数据，然后封装成Tuple形式，发送到Stream中。Spout是一个主动的角色，在接口内部有个nextTuple函数，Storm框架会不停的调用该函数（每次调用都会生成一个Tuple，源源不断地向外发送Tuple）
* 
* **Bolt**：Storm将Streams的状态转换过程抽象为Bolt。Bolt即可以处理Tuple，也可以将处理后的Tuple作为新的Streams发送给其他Bolt（执行用户处理逻辑）
* Bolt可以执行过滤、函数操作、Join、操作数据库等任何操作
* Bolt是一个被动的角色，其接口中有一个execute(Tuple input)方法，在接收到消息之后会调用此函数，用户可以在此方法中执行自己的处理逻辑
* 
* **Topology**：Storm将Spouts和Bolts组成的网络抽象成Topology，它可以被提交到Storm集群执行。Topology可视为流转换图，图中节点是一个Spout或Bolt，边则表示Bolt订阅了哪个Stream。当Spout或者Bolt发送元组时，它会把元组发送到每个订阅了该Stream的Bolt上进行处理
* Topology里面的每个处理组件（Spout或Bolt）都包含处理逻辑， 而组件之间的连接则表示数据流动的方向
* Topology里面的每一个组件都是并行运行的
* 
* **Stream Groupings**：Storm中的Stream Groupings用于告知Topology如何在两个组件间（如Spout和Bolt之间，或者不同的Bolt之间）进行Tuple的传送。每一个Spout和Bolt都可以有多个分布式任务，一个任务在什么时候、以什么方式发送Tuple就是由Stream Groupings来决定的
* 
* 目前，Storm中的Stream Groupings有如下几种方式：
* (1)ShuffleGrouping：随机分组，随机分发Stream中的Tuple，保证每个Bolt的Task接收Tuple数量大致一致
* (2)FieldsGrouping：按照字段分组，保证相同字段的Tuple分配到同一个Task中
* (3)AllGrouping：广播发送，每一个Task都会收到所有的Tuple
* (4)GlobalGrouping：全局分组，所有的Tuple都发送到同一个Task中
* (5)NonGrouping：不分组，和ShuffleGrouping类似，当前Task的执行会和它的被订阅者在同一个线程中执行
* (6)DirectGrouping：直接分组，直接指定由某个Task来执行Tuple的处理

### 2.了解：VxWorks, μClinux, μC/OS-II和eCos等4种性能优良并广泛应用的实时操作系统RTOS

1. **VxWorks** 操作系统是美国WindRiver公司于1983年设计开发的一种嵌入式实时操作系统(RTOS)，是嵌入式开发环境的关键组成部分。良好的持续发展能力、高性能的内核以及友好的用户开发环境，在嵌入式实时操作系统领域占据一席之地。它以其良好的可靠性和卓越的实时性被广泛地应用在通信、军事、航空、航天等高精尖技术及实时性要求极高的领域中，如卫星通讯、军事演习、弹道制导、飞机导航等。在美国的 F-16、FA-18战斗机、B-2 隐形轰炸机和爱国者导弹上，甚至连1997年4月在火星表面登陆的火星探测器、2008年5月登陆的凤凰号，和2012年8月登陆的好奇号也都使用到了VxWorks上。
2. **μClinux**为“MicroController Linux” 的缩写，通常念作“you-see-linux”，是[Linux](https://zh.wikipedia.org/wiki/Linux)的延生版本，专门使用在无[存储器管理单元](https://zh.wikipedia.org/wiki/%E5%86%85%E5%AD%98%E7%AE%A1%E7%90%86%E5%8D%95%E5%85%83)的[嵌入式系统](https://zh.wikipedia.org/wiki/%E5%B5%8C%E5%85%A5%E5%BC%8F%E7%B3%BB%E7%BB%9F)。
3. **μC/OS-II**由[Micrium](https://zh.wikipedia.org/w/index.php?title=Micrium&action=edit&redlink=1)公司提供，是一个可[移植](https://zh.wikipedia.org/wiki/%E7%A7%BB%E6%A4%8D_(%E8%BB%9F%E9%AB%94))、可[固化](https://zh.wikipedia.org/wiki/%E5%9B%BA%E5%8C%96)的、可[裁剪](https://zh.wikipedia.org/w/index.php?title=%E8%A3%81%E5%89%AA&action=edit&redlink=1)的、占先式多任务实时内核，它适用于多种[微处理器](https://zh.wikipedia.org/wiki/%E5%BE%AE%E5%A4%84%E7%90%86%E5%99%A8)，[微控制器](https://zh.wikipedia.org/wiki/%E5%BE%AE%E6%8E%A7%E5%88%B6%E5%99%A8)和[数字处理芯片](https://zh.wikipedia.org/w/index.php?title=%E6%95%B0%E5%AD%97%E5%A4%84%E7%90%86%E8%8A%AF%E7%89%87&action=edit&redlink=1)（已经移植到超过100种以上的微处理器应用中）。同时，该系统源代码开放、整洁、一致，注释详尽，适合系统开发。 μC/OS-II已经通过[联邦航空局](https://zh.wikipedia.org/w/index.php?title=%E8%81%94%E9%82%A6%E8%88%AA%E7%A9%BA%E5%B1%80&action=edit&redlink=1)（FAA）商用航行器认证，符合[航空无线电技术委员会](https://zh.wikipedia.org/wiki/%E8%88%AA%E7%A9%BA%E6%97%A0%E7%BA%BF%E7%94%B5%E6%8A%80%E6%9C%AF%E5%A7%94%E5%91%98%E4%BC%9A)（RTCA）DO-178B标准。

* μC/OS-II可管理多达63个应用任务，并可以提供如下服务：
  + 信号量
  + 互斥信号量
  + 事件标识
  + 消息邮箱
  + 消息队列
  + 任务管理
  + 固定大小内存块管理
  + 时间管理
* 另外，在μC/OS-II内核之上，有如下独立模块可供用户选择：
  + μC/FS文件系统模块
  + μC/GUI图形软件模块
  + μC/TCP-IP协议栈模块
  + μC/USB协议栈模块

1. **eCos**（embedded configurable operating system）是一个在1997年由Cygnus Solutions Inc.[[2]](https://zh.wikipedia.org/wiki/ECos#cite_note-2)开发的小型[开放原始码](https://zh.wikipedia.org/wiki/%E9%96%8B%E6%94%BE%E5%8E%9F%E5%A7%8B%E7%A2%BC)[实时操作系统](https://zh.wikipedia.org/wiki/%E5%AE%9E%E6%97%B6%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)。后来该公司被[Redhat](https://zh.wikipedia.org/wiki/Redhat)收购。在2002年的早些时候[[3]](https://zh.wikipedia.org/wiki/ECos#cite_note-3)，[Redhat](https://zh.wikipedia.org/wiki/Redhat)停止了eCos开发，并解散了开发队伍。很多原开发者继续开发eCos，并且成立了他们自己的公司来为该软件提供服务。迫于开发者的要求，Redhat在2004年一月份将eCos的版权转移给了[自由软件基金会](https://zh.wikipedia.org/wiki/%E8%87%AA%E7%94%B1%E8%BD%AF%E4%BB%B6%E5%9F%BA%E9%87%91%E4%BC%9A)[[4]](https://zh.wikipedia.org/wiki/ECos#cite_note-4)，并于2008年5月份最终完成了版权转移。

* 此系统和[嵌入式](https://zh.wikipedia.org/wiki/%E5%B5%8C%E5%85%A5%E5%BC%8F)[Linux](https://zh.wikipedia.org/wiki/Linux)系统的差异是它采用静态链接[(static library)](https://zh.wikipedia.org/w/index.php?title=(static_library)&action=edit&redlink=1)的方式，最低编译核心可小至10K的级别，适合用于做[bootloader](https://zh.wikipedia.org/wiki/Bootloader)增强，或者用于构建微小型系统。 eCos的发行版还包括[RedBoot](https://zh.wikipedia.org/wiki/RedBoot)，一个[开放原始码](https://zh.wikipedia.org/wiki/%E9%96%8B%E6%94%BE%E5%8E%9F%E5%A7%8B%E7%A2%BC)的[应用程序](https://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F)。它使用[硬件抽象层](https://zh.wikipedia.org/w/index.php?title=%E7%A1%AC%E4%BB%B6%E6%8A%BD%E8%B1%A1%E5%B1%82&action=edit&redlink=1)提供[嵌入式系统](https://zh.wikipedia.org/wiki/%E5%B5%8C%E5%85%A5%E5%BC%8F%E7%B3%BB%E7%BB%9F)的启动[固件](https://zh.wikipedia.org/wiki/%E5%9B%BA%E4%BB%B6)。
* 除了自由版本以外，eCos还有一个称为**eCosPro**的商业版本。它是eCos的一个商业分支，由[eCosCentric](https://zh.wikipedia.org/w/index.php?title=ECosCentric&action=edit&redlink=1)开发，并包含有私有组件。它是稳定并经过测试和认证的版本[[5]](https://zh.wikipedia.org/wiki/ECos#cite_note-5)。在2017年，2017, eCosCentric发布了[[6]](https://zh.wikipedia.org/wiki/ECos#cite_note-6)对于全部树莓派单板计算机的eCos移植，并同时发布了这些移植的自由版本。

1. 出现的问题

无

1. 解决方案（列出遇到的问题和解决方案，列出没有解决的问题）

无

1. 个人学习体会

通过这次作业，我对实时计算系统Strom和实时操作系统RTOS的理解变得更深入了。