

博客

登录 | 注册

Q

≡

linezinglll

anzsoft的技术专栏

专注大数据，存储系统，云计算和分布式系统架构 : Since Dec. 2013

≡ 目录视图

≡ 摘要视图

RSS 订阅

个人资料



anzsoft

+ 加关注

发私信



≡ 恒

访问：885400次

积分：10348

等级：BLOG > ?

排名：第533名

原创：100篇 转载：0篇

译文：1篇 评论：593条

版权声明

鉴于很多网站转载本博客的文章而大部分都没有标明出处，本博客声明如下：本博客所有文章都为原创，未经允许，不得转载；未经允许，不得进行商业发布或者其他的盈利性活动。如发现侵权，将依法追究其责任。

跳吧

博客Markdown编辑器上线啦

那些年我们追过的Wrox精品红皮计算机图书

PMBOK第五版精讲视频教程

火星敏捷开发1001问

原 RabbitMQ消息队列（一）: Detailed Introduction 详细介绍

分类：架构探索

2014-02-20 20:13

11222人阅读

评论(4)

收藏

举报

消息队列

Message Queue

RabbitMQ

模块间通信

系统内部通信

目录(?)

[+]

1. 历史

RabbitMQ是一个由erlang开发的AMQP（Advanced Message Queue）的开源实现。AMQP 的出现其实也是应了广大人民群众的需求，虽然在同步消息通讯的世界里有很多公开标准（如 COBAR的 IIOP，或者是 SOAP 等），但是在异步消息处理中却不是这样，只有大企业有一些商业实现（如微软的 MSMQ，IBM 的 Websphere MQ 等），因此，在 2006 年的 6 月，Cisco、Redhat、iMatix 等联合制定了 AMQP 的公开标准。

RabbitMQ是由RabbitMQ Technologies Ltd开发并且提供商业支持的。该公司在2010年4月被 SpringSource（VMWare的一个部门）收购。在2013年5月被并入Pivotal。其实VMWare，Pivotal和EMC本质上是一家的。不同的是VMWare是独立上市子公司，而Pivotal是整合了EMC的某些资源，现在并没有上市。

RabbitMQ的官网是http://www.rabbitmq.com

2. 应用场景

言归正传。RabbitMQ，或者说AMQP解决了什么问题，或者说它的应用场景是什么？

对于一个大型的软件系统来说，它会有很多的组件或者说模块或者说子系统或者（subsystem or Component or submodule）。那么这些模块的如何通信？这和传统的IPC有很大的区别。传统的IPC很多都是在单一系统上的，模块耦合性很大，不适合扩展（Scalability）；如果使用socket那么不同的模块的确可以部

第1页 共7页

2015年03月05日 14:37

博客专栏



Spark技术内幕
文章：25篇
阅读：219797



HDFS深入解析
文章：4篇
阅读：37372



存储那些事儿
文章：11篇
阅读：100587



RabbitMQ从入门到精通
文章：9篇
阅读：76737



Linux 调试技巧
文章：9篇
阅读：116724

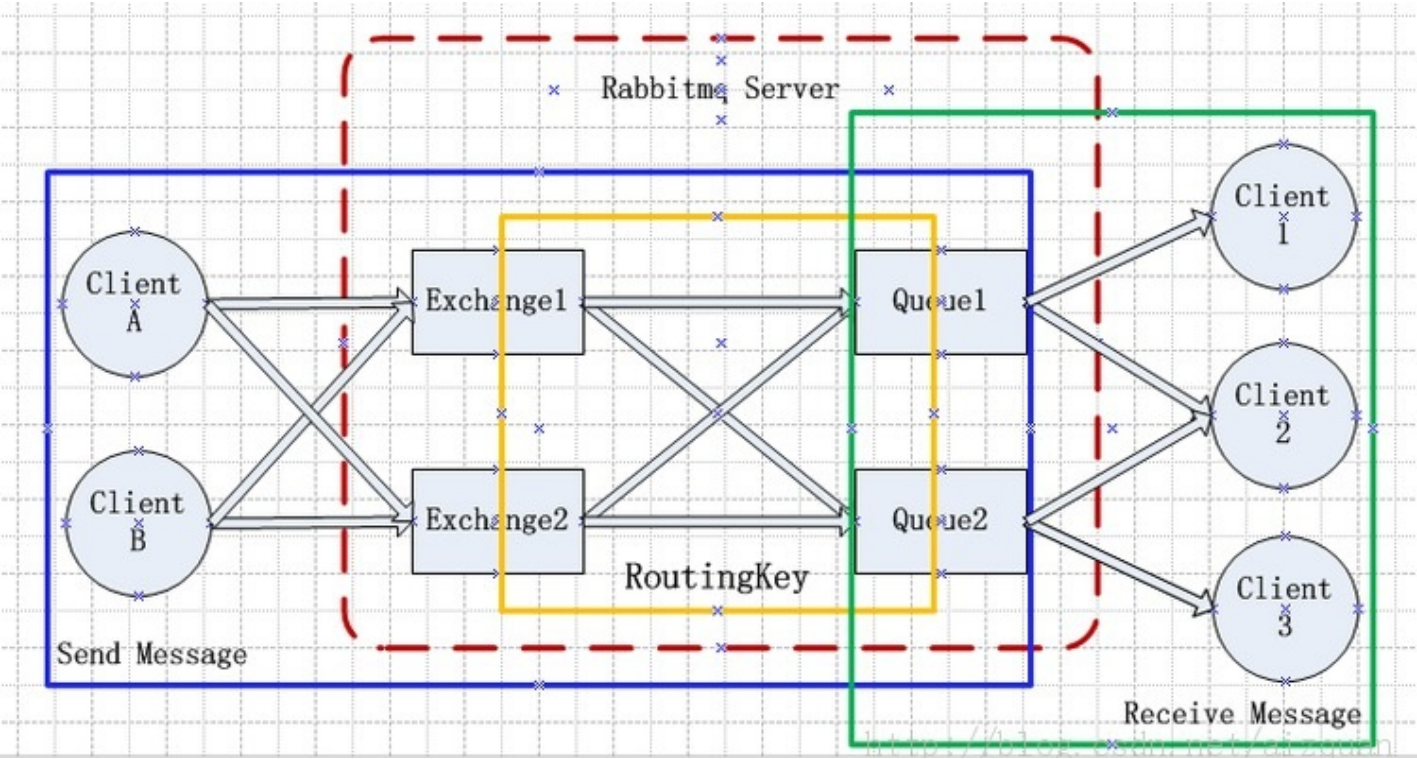
我的微博

- 署到不同的机器上，但是还是有很多问题需要解决。比如：
- 1) 信息的发送者和接收者如何维持这个连接，如果一方的连接中断，这期间的数据如何方式丢失？
 - 2) 如何降低发送者和接收者的耦合度？
 - 3) 如何让Priority高的接收者先接到数据？
 - 4) 如何做到load balance？有效均衡接收者的负载？
 - 5) 如何有效的将数据发送到相关的接收者？也就是说将接收者subscribe 不同的数据，如何做有效的filter。
 - 6) 如何做到可扩展，甚至将这个通信模块发到cluster上？
 - 7) 如何保证接收者接收到了完整，正确的数据？

AMDQ协议解决了以上的问题，而RabbitMQ实现了AMQP。

3. 系统架构

成为系统架构可能不太合适，可能叫应用场景的系统架构更合适。



这个系统架构图版权属于sunjun041640。

RabbitMQ Server: 也叫broker server，它不是运送食物的卡车，而是一种传输服务。原话是

技术问答

快速回复

返回顶部

 微博

 anzhsoft

+ 加关注

转发微博

EMC中国-云计算

：传说中的全异步算法。



TA 的粉丝 (436) [全部»](#)

Nich0las

xuyuantr

- 文章分类
- C++对象模型 (18)

架构探索 (34)

Linux (16)

Spark (27)

云计算 (30)

信息存储 (14)

调试实践 (9)

并发编程 (6)

面试经验 (2)

算法实践 (3)

杂七杂八 (18)

RabbitMQ isn't a food truck, it's a delivery service. 他的角色就是维护一条从Producer到Consumer的路线，保证数据能够按照指定的方式进行传输。但是这个保证也不是100%的保证，但是对于普通的应用来说这已经足够了。当然对于商业系统来说，可以再做一层数据一致性的guard，就可以彻底保证系统的一致性了。

Client A & B： 也叫Producer，数据的发送方。createmessages and publish (send) them to a broker server (RabbitMQ).一个Message有两个部分：payload（有效载荷）和label（标签）。payload顾名思义就是传输的数据。label是exchange的名字或者说是一个tag，它描述了payload，而且RabbitMQ也是通过这个label来决定把这个Message发给哪个Consumer。AMQP仅仅描述了label，而RabbitMQ决定了如何使用这个label的规则。

Client 1, 2, 3： 也叫Consumer，数据的接收方。Consumersattach to a broker server (RabbitMQ) and subscribe to a queue。把queue比作是一个有名字的邮箱。当有Message到达某个邮箱后，RabbitMQ把它发送给它的某个订阅者即Consumer。当然可能会把同一个Message发送给很多的Consumer。在这个Message中，只有payload，label已经被删掉了。对于Consumer来说，它是不知道谁发送的这个信息的。就是协议本身不支持。但是当然了如果Producer发送的payload包含了Producer的信息就另当别论了。

对于一个数据从Producer到Consumer的正确传递，还有三个概念需要明确：exchanges, queues and bindings。

Exchanges are where producers publish their messages.

Queuesare where the messages end up and are received by consumers

Bindings are how the messages get routed from the exchange to particular queues.

还有几个概念是上述图中没有标明的，那就是Connection（连接），Channel（通道，频道）。

Connection： 就是一个TCP的连接。Producer和Consumer都是通过TCP连接到RabbitMQ Server的。以后我们可以看到，程序的起始处就是建立这个TCP连接。

Channels： 虚拟连接。它建立在上述的TCP连接中。数据流动都是在Channel中进行的。也就是说，一般情况是程序起始建立TCP连接，第二步就是建立这个Channel。

那么，为什么使用**Channel**，而不是直接使用**TCP**连接？

对于OS来说，建立和关闭TCP连接是有代价的，频繁的建立关闭TCP连接对于系统的性能有很大的影响，而且TCP的连接数也有限制，这也限制了系统处理高并发的能力。但是，在TCP连接中建立Channel是没有上述代价的。对于Producer或者Consumer来说，可以并发的使用多个Channel进行Publish或者Receive。有实验表明，1s的数据可以Publish10K的数据包。当然对于不同的硬件环境，不同的数据包大小这个数据肯定不一样，但是我只想说明，对于普通的Consumer或者Producer来说，这已经足够了。如果不够用，你考虑的应该 是如何细化split你的设计。

- 技术问答
- 快速回复
- 返回顶部

文章存档

- [2015年02月](#) (1)
- [2015年01月](#) (9)
- [2014年12月](#) (5)
- [2014年11月](#) (4)
- [2014年10月](#) (8)

展开

友情衔接

- [大数据，存储系统与云计算](#)
- [EMC研究院](#)
- [酷壳](#)
- [董博士](#)
- [AllThingsDistributed](#)
- [移动云：韩曙亮CSDN](#)
- [移动云：韩曙亮独立博客](#)

阅读排行

- [从Storm和Spark 学习流式计算](#) (27399)
- [Linux Debugging（一）：从入门到精通](#) (23493)
- [Spark技术内幕：Client，Driver和Executor](#) (20408)
- [Spark：大数据的电花火石](#) (19778)
- [Spark技术内幕：Master和Zookeeper](#) (18591)
- [博客流量分析](#) (17049)
- [Spark技术内幕：一个图例](#) (15954)
- [Linux Debugging（五）：多线程](#) (15770)
- [RAMCloud：内存云存储](#) (15565)
- [Linux Debugging（四）：多线程](#) (15499)

评论排行

- [2014博客之星：请大家投票](#) (67)
- [Spark：大数据的电花火石](#) (45)

4. 进一步的细节阐明

4.1 使用ack确认Message的正确传递

默认情况下，如果Message 已经被某个Consumer正确的接收到了，那么该Message就会被从queue中移除。当然也可以让同一个Message发送到很多的Consumer。

如果一个queue没被任何的Consumer Subscribe（订阅），那么，如果这个queue有数据到达，那么这个数据会被cache，不会被丢弃。当有Consumer时，这个数据会被立即发送到这个Consumer，这个数据被Consumer正确收到时，这个数据就被从queue中删除。

那么什么是正确收到呢？通过ack。每个Message都要被**acknowledged**（确认，ack）。我们可以显示的在程序中去ack，也可以自动的ack。如果有数据没有被ack，那么：

RabbitMQ Server会把这个信息发送到下一个Consumer。

如果这个app有bug，忘记了ack，那么RabbitMQ Server不会再发送数据给它，因为Server认为这个Consumer处理能力有限。

而且ack的机制可以起到限流的作用（**Benefitto throttling**）：在Consumer处理完成数据后发送ack，甚至在额外的延时后发送ack，将有效的balance Consumer的load。

当然对于实际的例子，比如我们可能会对某些数据进行merge，比如merge 4s内的数据，然后sleep 4s后再获取数据。特别是在监听系统的state，我们不希望所有的state实时的传递上去，而是希望有一定的延时。这样可以减少某些IO，而且终端用户也不会感觉到。

4.2 Reject a message

有两种方式，第一种Reject可以让RabbitMQ Server将该Message 发送到下一个Consumer。第二种是从queue中立即删除该Message。

4.3 Creating a queue

Consumer和Procuder都可以通过 **queue.declare** 创建queue。对于某个Channel来说，Consumer不能declare一个queue，却订阅其他的queue。当然也可以创建私有的queue。这样只有app本身才可以使用这个queue。queue也可以自动删除，被标为auto-delete的queue在最后一个Consumer unsubscribe后就会被自动删除。那么如果是创建一个已经存在的queue呢？那么不会有任何的影响。需要注意的是没有任何的影响，也就是说第二次创建如果参数和第一次不一样，那么该操作虽然成功，但是queue的属性并不会被修改。

那么谁应该负责创建这个**queue**呢？是**Consumer**，还是**Producer**？

如果queue不存在，当然Consumer不会得到任何的Message。但是如果queue不存在，那么Producer Publish的Message会被丢弃。所以，还是为了数据不丢失，Consumer和Producer都try to create the queue！反正不

技术问答

快速回复

返回顶部

- Spark技术内幕：Stage找 (25)
- Spark技术内幕：Execut (24)
- 抄袭，借鉴？ (19)
- 分布式系统的设计几个要 (19)
- Redefine：Change in the (18)
- 从Storm和Spark 学习流 (16)
- Spark技术内幕：Worker (15)
- RAMCloud：内存云存储 (15)

最新评论

- 2014博客之星，感激自己，谢谢 changshenglugu: 厉害
- Spark技术内幕：Shuffle的性能讨论 鲍礼彬: 大神，膜拜
- 2014博客之星，感激自己，谢谢 anzsoft: @u010850027:新年快乐
- 2014博客之星，感激自己，谢谢 anzsoft: @cdhnlsj:新年快乐
- 2014博客之星，感激自己，谢谢 丁国华: 恭喜博主 新年快乐`(*n_n*)`
- 2014博客之星，感激自己，谢谢 前端开发whqet: 赞！
- 2014博客之星，感激自己，谢谢 竹落: 新年快乐 工作顺利
- 2014博客之星，感激自己，谢谢 Colin丶: 加油。
- Spark技术内幕：Storage 模块整理 anzsoft: @w412692660:翻墙吧。或者直接使用meetup的app。
- Spark技术内幕：Storage 模块整理 然月: @woshifou:兄弟，spark meetup为啥我总是注册不了，上次想报名，结果一直注册总是失...

推荐文章

管怎么样，这个接口都不会出问题。

queue对load balance的处理是完美的。对于多个Consumer来说，RabbitMQ 使用循环的方式（round-robin）的方式均衡的发送给不同的Consumer。

4.4 Exchanges

从架构图可以看出，Procuder Publish的Message进入了Exchange。接着通过“routing keys”， RabbitMQ会找到应该把这个Message放到哪个queue里。queue也是通过这个routing keys来做的绑定。

有三种类型的Exchanges：direct, fanout,topic。 每个实现了不同的路由算法（routing algorithm）。

- **Direct exchange:** 如果 routing key 匹配, 那么Message就会被传递到相应的queue中。其实在queue创建时，它会自动的以queue的名字作为routing key来绑定那个exchange。
- **Fanout exchange:** 会向响应的queue广播。
- **Topic exchange:** 对key进行模式匹配，比如ab*可以传递到所有ab*的queue。

4.5 Virtual hosts

每个virtual host本质上都是一个RabbitMQ Server，拥有它自己的queue，exchagne，和bingns rule等等。这保证了你可以在多个不同的application中使用RabbitMQ。

接下来我会使用Python来说明RabbitMQ的使用方法。

请移步 [RabbitMQ消息队列（二）： "Hello, World"](#)

尊重原创，转载请注明出处 anzsoft <http://blog.csdn.net/anzhsoft/article/details/19563091>。



- ^ 上一篇 并发编程（三）： 使用C++11实现无锁stack（lock-free stack）
- ^ 下一篇 RabbitMQ消息队列（二）： "Hello, World"

- 技术问答
- 快速回复
- 返回顶部

- * [【ShaderToy】开篇](#)
- * [FFmpeg源代码简单分析：avio_open2\(\)](#)
- * [技能树之旅: 从模块分离到测试](#)
- * [Qt5官方demo解析集36——Wiggly Example](#)
- * [Unity3d HDR和Bloom效果（高动态范围图像和泛光）](#)

顶

9

踩

0

主题推荐

- 2013年5月
- 系统架构
- springsource
- application
- scalability

猜你在找

- [Redis mysql应用](#)
 - [ehcache memcache redis 三大缓存男高音](#)
 - [去哪儿网面试问题](#)
 - [基于Redis bitmap实现签到功能](#)
 - [Maven 使用Cargo 快速部署Web项目](#)
- [java集群](#)
 - [Redis笔记二主从复制](#)
 - [在MSMQ中使用事务处理消息](#)
 - [elasticsearch中如何高效的使用filter](#)
 - [maven 打包可执行jar的方法](#)

准备好了么？！

更多职位尽在 **CSDN JOB**

▪ 互联网产品系统架构师	我要跳槽	▪ 互联网产品系统架构师	我要跳槽
广州市朗际软件科技有限公司	10-15K/月	广州市朗际软件科技有限公司	10-15K/月
▪ Android 工程师	我要跳槽	▪ 前端交互设计高级工程师	我要跳槽
北京美易科技有限公司	12-25K/月	广州市朗际软件科技有限公司	5-10K/月

查看评论

2楼 [HXqwer](#) 2014-07-29 18:06发表




博主你好我是新手，请问rabbitmq如果队列满了是什么后果？队列长度是多少能不能设置？怎么判断队列溢出？谢谢好心博主

Re: [anzhsoft](#) 2014-07-30 10:08发表





回复HXqwer：可以看这篇博文：<http://backend.blog.163.com/blog/static/202294126201322511327882/>

1楼 [津山珊](#) 2014-03-13 20:34发表



学习Hadoop也使用了RabbitMQ。读了这篇文章很受益。

Re: anzsoft 2014-03-13 20:44发表 



回复津山珊：算是很成熟的消息队列了。而且易用。






您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目											
全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack	
VPN	Spark	ERP	IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery
BI	HTML5	Spring	Apache	.NET	API	HTML	SDK	IIS	Fedora	XML	LBS
Splashtop	UML	components	Windows Mobile	Rails	QEMU	KDE	Cassandra	CloudStack	FTC		
coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	SpringSide	Maemo			
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr	
Angular	Cloud Foundry	Redis	Scala	Django	Bootstrap						

- 技术问答
- 快速回复
- 返回顶部

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

 网站客服  杂志客服  微博客服  webmaster@csdn.net  400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 