

# 新型配电主站系统消息总线性能分析与测试

张玉中, 刘仁千, 李国杰, 张超永, 杨丰源

(许继电气股份有限公司, 河南 许昌 461000)

**摘要:** 消息总线服务为配电主站系统的各个模块提供基于 topic 的消息订阅和发布, 其主体服务采用的是开源软件 kafka; 针对消息总线的性能, 分析了大数据容量、数据处理速率、消费者上限和稳定性等主要性能指标; 设计了一个海量数据处理及速率统计的测试框架, 该框架主要采用 Qt/C++ 实现; 实际应用证明, 该测试框架简单、易操作, 很大程度上提高了测试效率。

**关键词:** 消息总线; 性能分析; 大数据容量; 数据处理速率; 测试

## Performance Analysis and Test of a New Power Distribution Main Station System Message Bus

Zhang Yuzhong, Liu Renqian, Li Guojie, Zhang Chaoyong, Yang Fengyuan

(XJ Electricity Co., Ltd., Xuchang 461000, China)

**Abstract:** The message bus service provides topic based message subscription and release for each module of the distribution master station system, and its main service is the open source software Kafka. In view of the performance of the message bus, the main performance indexes, such as large data capacity and data processing rate, are analyzed. A test framework for mass data processing and rate statistics is designed. The framework is implemented mainly by Qt/C++. The practical application shows that the test framework is simple and easy to operate, and the test efficiency is improved to a great extent.

**Keywords:** message bus; performance analysis; large data capacity; data processing rate; test

### 0 引言

消息总线服务为进程间(多台计算机之间和一台计算机内部)的消息传输提供支持, 具有消息主题的创建/删除、消息的订阅/发布等功能, 以接口函数的形式提供给各类应用; 具有组播、广播和点到点传输形式, 支持一对多、一对一的信息交换场合<sup>[1]</sup>。当发送者需要分发一组新的消息时, 它向消息总线服务器发出一个请求, 消息总线服务器会负责把请求递送给所有的订阅者。经过对当前各种开源软件和其它行业软件的分析对比, 配电系统决定采用 Kafka 作为消息总线的主体服务。

消息总线服务为配电主站系统的各个模块提供基于 topic 的消息订阅和发布, 消息总线服务应考虑系统的集群部署、系统冗余功能等<sup>[2]</sup>; 消息总线服务应具备足够的吞吐量, 处理大量消息传递能力, 所以较高的性能是消息总线的基本要求。

### 1 消息总线实现机制

消息总线采用 Qt/C++ 对 Kafka 的客户端进行封装, 屏蔽各种网络处理细节, 给平台其它应用一个简单易用的接

口<sup>[3]</sup>。整个消息处理组件主要包括两部分内容: 消息的发布和消息的处理。为了简化应用, 消息体现为函数参数, 可以任意定义消息体结构而不改变程序结构。消息的处理采用回调函数的方式, 向系统注册处理某种 topic 的回调函数即可。下面具体介绍一下消息发布和消息处理的实现机制。

#### 1.1 消息发布机制

消息发布机制是对 kafka 客户端提供的信息发送函数进行封装, 以达到向服务器上传数据的目的<sup>[4]</sup>。由于事先不确定用户上传的数据类型, 封装采用模板模式, 支持任意数据类型。上送方式为单条上送, 即一次只能上送一条数据, 数据大小不做限制。并且接收服务器返回的上送数据状态, 成功返回 0, 失败返回 -1 等。

#### 1.2 消息处理机制

消息处理采用的是向服务器注册机制, 对 kafka 客户端提供的注册函数进行封装, 以达到向服务器订阅数据的目的。封装技术与发布机制类似, 采用模板形式。注册可以分组注册也可以同一组批量注册。同一用户可以注册到服务器的不同组, 但不能在同一组内多次注册。

### 2 消息总线性能指标及测试方法分析

消息总线的性能指标主要包括: 大数据容量、数据处理速率、消费者个数上限、消费者个数上限、稳定性等性能指标。以上几项性能指标中大数据容量、数据处理速率、消费者个数上限和稳定性是最重要的, 接下来我们具体分析这两项性能指标。

#### 2.1 大数据容量

大数据容量分为单机数据容量和集群数据容量, 是指

收稿日期: 2018-05-11; 修回日期: 2018-06-30。

作者简介: 张玉中(1983-), 男, 硕士, 工程师, 主要从事系统软件的白盒测试方向的研究。

刘仁千(1978-), 男, 高级工程师, 主要从事白盒测试技术方向的研究。

李国杰(1967-), 男, 硕士, 教授级高级工程师, 主要从事电力系统研发测试方向的研究。

消息总线的吞吐量<sup>[5]</sup>。主要测试方法是采用控制变量的方法将测试用例中的消息数量和单个消息大小分别增大, 将数据总量逐级增加, 测试容量上限, 记录上限值与单机最大容量是否成正比例。在上限范围内, 单机或集群各项功能正常工作。

## 2.2 数据处理速率

数据处理速率又包括: 生产者数据处理速率、消费者数据处理速率、网络风暴中单条生产者生产速率和网络风暴中单条消费者接收速率。

生产者数据处理速率是指生产者生产数据的效率。主要测试方法是启用测试用例, 以 50% 容量上限值发送数据, 记录 Producer 的生产时间, 计算生产者的数据处理速率; 更改测试用例, 逐步提升到 80% 容量上限值发送量, 记录 Producer 生产时间, 计算生产者数据处理速率, 直至发送量提升至容量上限值, 借此测试生产者数据处理速率上限。在上限范围内, 集群各项功能正常工作。

消费者数据处理速率指消费者的接收效率。主要测试方法是启用测试用例, 以 50% 容量上限值发送数据, 记录 Consumer 的消费时间, 计算消费者的数据处理速率; 更改测试用例, 逐步提升到 80% 容量上限值发送量, 记录 Consumer 消费时间, 计算消费者数据处理速率, 直至发送量提升至容量上限值, 借此测试消费者数据处理速率上限。在上限范围内, 集群各项功能正常工作。

网络风暴中单条生产者生产速率指网络风暴情况下生产者生产一条数据的效率<sup>[6]</sup>。主要测试方法是更改测试用例, 逐步提升到 80% 容量上限值发送量, 记录 Producer 生产时间, 计算生产者单条数据处理速率, 直至发送量提升至容量上限值, 借此测试生产者单条数据处理速率上限。在上限范围内, 集群各项功能正常工作。

网络风暴中单条消费者接收速率指网络风暴情况下消费者接收一条数据的效率。主要测试方法是更改测试用例, 逐步提升到 80% 容量上限值发送量, 记录 Consumer 消费时间, 计算消费者单条数据处理速率, 直至发送量提升至容量上限值, 借此测试消费者单条数据处理速率上限。在上限范围内, 集群各项功能正常工作。

## 2.3 消费者个数上限

消费者个数上限又分为单机消费者个数上限和集群消费者个数上限。其中单机消费者上限是指单机消费者的最大上限, 主要测试方法是启动测试用例, 使 Producer 以 50% 容量上限值持续生产数据, 在一台服务器上逐步增加 Consumer 个数同时消费, 测试消费者个数上限; 将测试用例调整至容量上限值, 重复上述工作, 测试消费者上限个数。在上限范围内, 各项功能正常工作。集群消费者个数上限是指整个集群的消费者上限者上限, 主要测试方法是启动测试用例, 使 Producer 以 50% 容量上限值持续生产数据, 在多台服务器上逐步增加 Consumer 个数同时消费, 测试消费者总个数上限; 将测试用例调整至容量上限值, 重复上述工作, 测试消费者上限总个数, 在上限范围内, 集群各项功能正常工作。

## 2.4 稳定性

稳定性主要是指消息总线服务长时间稳定的运行和在高负荷下稳定的运行。具体长时间稳定运行是指最低保证 50% 容量上限值 72 小时稳定运行, 具体测试方法是更改测试用例, 50% 容量上限值发送数据量, 进行 72 小时拷机测试, 拷机期间各项功能正常, 程序不应产生异常或死机等现象, 拷机测试期间观测记录 CPU 平均负荷率, 拷机测试期间, 观测内存占有量, 不发生内存泄露, 拷机测试期间, 各节点网络平均负荷率。并且最低保证 80% 容量上限值 30 分钟稳定运行, 具体测试方法是更改测试用例, 80% 容量上限值发送数据量, 进行 30 分钟拷机测试, 拷机期间各项功能正常, 程序不应产生异常或死机等现象, 拷机测试期间观测记录 CPU 平均负荷率, 拷机测试期间, 观测记录内存占有量, 不发生内存泄露, 拷机测试期间, 记录各节点网络平均负荷率。高负荷稳定运行是指最低保证 100% 容量上限值 5 分钟稳定运行, 具体测试方法是更改测试用例, 100% 容量上限值发送数据量, 进行 5 分钟压力测试, 压力测试期间各项功能正常, 程序不应产生异常或死机等现象, 拷机测试期间观测记录 CPU 平均负荷率, 拷机测试期间, 观测记录内存占有量, 不发生内存泄露, 拷机测试期间, 记录各节点网络平均负荷率。

## 3 测试框架设计及实现

### 3.1 硬件架构设计

消息总线测试环境采用三台服务器搭载 Kafka 服务、组成集群, 系统采用单网千兆网络进行连接, 满足 2k+1 配置, 至少 k+1 台服务器正常使用时, 保证消息总线能正常工作。见图 1。

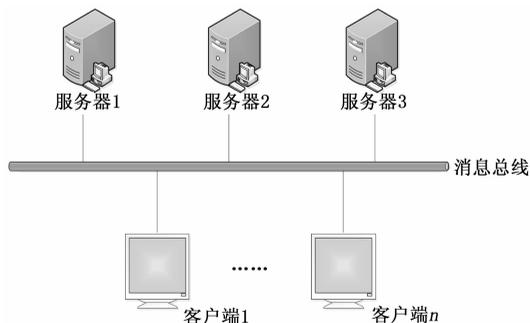


图 1 消息总线性能测试硬件架构图

### 3.2 软件框架设计

如图 2 所示, 消息总线软件测试框架主要包括大数据生成、数据发送、数据接收和时间统计 4 个功能模块。

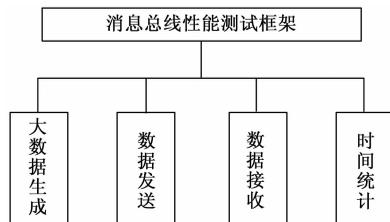


图 2 消息总线性能测试软件框架功能图

数据生成模块的主要功能是生成测试数据，为了便于统计数据大小，这里数据类型统一使用字符型。可以单个添加数据，也可以批量生成大量数据。数据发送模块主要功能是与消息总线服务器建立链接并发送数据到服务器。数据接收模块主要功能是消费者注册和接收数据。时间统计模块主要功能是统计并记录消息总线处理数据的时间。由于要处理海量的数据，这里采用多线程技术，来保证测试框架界面的操作流畅。

### 3.3 软件框架实现

Linux 系统上采用 Qt/C++ 实现软件框架。主窗口类设计如下：

```
class PUITest: public QMainWindow
{
    Q_OBJECT
public:
    explicit PUITest(QWidget * parent = 0);
    ~PUITest();
    int m_DowNum, m_DowNum_reg, outcnt, totalcnt, cb_cnt;
    outputDockWidget * m_InfoOutput; //操作信息输出窗口
    sendThread * sendmsgthread; //数据发送线程
    runthread * runmsgthread; //海量数据发送线程
    bool runOverflag;
    int tId; //线程 ID
    void removeDockTitle(QDockWidget * docwidget); //去除头标志
    void timerEvent(QTimerEvent * event); //定时器自动调用函数
    void InitTabWidget();
    int test2(const QString str);
    void Output(QString s);
    void msgsendThread();
    void magrunThread();
public slots:
    void sendmsgover();
private slots:
    void on_action_about_triggered();
    void on_action_add_triggered();
    void on_action_delete_triggered();
    void n_action_reg_triggered();
    void on_action_send_triggered();
    void on_checkBox_clicked();
    void on_action_unreg_triggered();
    void on_action_addNumbers_triggered();
    void on_action_save_triggered();
    void on_action_open_triggered();
    void on_action_run_triggered();
private:
    Ui::PUITest * ui;
};
线程类设计如下:
class sendThread: public QThread
{
    Q_OBJECT
```

```
signals:
void sendover(); //发送数据结束信号
protected:
void run(); //线程自动调用
};
```

框架实现主界面如图 3 所示，主界面中有两个分页面，一个是发送信息配置，主要包括数据生成、发送次数设置、发送延时设置等。一个是接收信息配置，主要是注册信息的设置等。数据可以单个添加，更可以批量生成。另外还有操作提示窗口，时间统计窗口等。



图 3 消息总线测试软件框架主界面

## 4 消息总线性能测试及效果分析

### 4.1 性能测试

采用前几节中提出的测试方法及设计的测试框架，对消息总线性能进行测试。测试中使用三台服务器组成消息总线集群，两台客户端安装测试程序。具体软硬件信息如表 1 和表 2 所示。

表 1 软硬件配置环境

测试机	浪潮 NF5280	浪潮 NF5280	HP DL560
	M2	M2	G8
物理内存大小	4G	4G	8G
CPU	Intel Xeon(R) CPU E5606 @2.13GHz * 8	Intel Xeon(R) CPU E5506 @2.13GHz * 8	Intel Xeon(R) CPU E5-4610 0 @2.4GHz * 12
内核版本	3.10.0-514.2.2el7.x86_64		
操作系统	Red Hat Enterprise Linux Server 7.3		
网卡带宽	1000Mbps		
Kafka 版本	0.10.1.1		
Java 版本	1.8.0_111		

表 2 软件消息

软件名称	版本	字节	日期
消息总线测试工具	1.10	47,856	2017.2.20
Zsb_loaderc	/	47,872	2017.2.20
Zsb_libecore.so	/	68,888	2017.2.20
Zsb_libmsgshandler.so	/	101,408	2017.2.20
Zsb_librdKafka.so	/	2,823,976	2017.2.20
Zsb_librdKafka++.so	/	881,864	2017.2.20
Zsb_libconnectslot.so	/	54,720	2017.2.20
Zsb_libpanservertest.so	/	31,536	2017.2.20
Zsb_libclienttest.so	/	60,376	2017.2.20

### 4.2 测试效果

通过测试框架的测试软件自动生成海量的测试数据, 并自动统计数据处理时间和读取服务器资源占用信息。以下所有记录均在 HP DL560 G8 服务器上进行测试, 测试 10 次求平均值, 性能测试结果如表 3~6 所示。

表 3 生产者生产速率

发送消息大小(byte)	发送量(条)	最大发送时间(s)	平均发送时间(s)	最小发送时间(s)	平均发送量(万条/s)
4	400000	1.26	1.15	1.06	34.93
16	400000	1.24	1.15	1.24	34.78
64	400000	1.47	1.33	1.18	30.08
256	400000	1.79	1.74	1.65	22.98
1024	400000	3.67	3.63	3.57	11.02

表 4 消费者消费速率

接收消息大小(byte)	接收量(条)	最大接收时间(s)	平均接收时间(s)	最小接收时间(s)	平均接收量(万条/s)
4	400000	6.54	5.92	5.69	6.76
16	400000	6.57	6.41	1.11	6.24
64	400000	11	10.44	10.2	3.83
256	400000	24.6	24.4	24.2	1.64
1024	400000	28.7	28.3	28	1.41

表 5 网络风暴中单条生产信息生产时间

网络风暴消息大小(byte)	网络风暴发送速率(万条/秒)	生产者发送消息大小(byte)	生产者生产时间最大值(ms)	生产者生产时间最小值(ms)	生产者生产时间平均值(ms)
4	34.93	4	2	1	1.4
16	34.78	4	2	1	1.4
64	30.08	4	2	1	1.6
256	22.98	4	2	1	1.4
1024	11.02	4	2	1	1.4

表 6 网络风暴中单条消费消息收时间

网络风暴消息大小(byte)	网络风暴接收速率(万条/秒)	消费者接收消息大小(byte)	消费者接收时间最大值(ms)	消费者接收时间最小值(ms)	消费者接收时间平均值(ms)
4	6.76	4	672	22	163
16	6.24	4	2096	9	631
64	3.83	4	2648	31	219
256	1.64	4	83	8	705
1024	1.41	4	2952	25	640

表 3 是生产者生产效率的测试结果, 从该测试结果中可以发现, 发送量一定的情况下, 随着单条数据字节数的增加, 发送效率逐渐降低。表 4 是消费者消费效率的测试的测试结果, 从该测试结果中可以发现, 接收量一定的情况下, 随着单条数据字节数的增加, 接收效率逐渐降低。通过对比发现, 消费效率要明显高于生产效率。表 5 是网络风暴中单条信息的生产效率测试结果, 从该测试结果中可以发现, 不同程度网络风暴情况下, 生产单条数据的效率相对稳定。也就是说网络风暴对生产者的生产效率影响

不大。表 6 是网络风暴中单条消费信息的消费效率的测试结果, 从该测试结果中可以发现, 不同程度网络风暴情况下, 消费单条数据的效率不稳定。也就是说网络风暴对消费者的效率影响较大。

消费者上限测试, 当消费者为 10 000 时, 系统各项指标均正常运行。这以远远超出我们实际应用时的上限, 新型配电主站系统实际应用时的消费者数量不超过 1 000, 所以这里不再具体测试。

稳定性测试时接入不少于 10 000 台配电终端, 每台终端定义 100 个“三遥”数据信息接入被测系统, 系统应不间断运行维持 72 h, 被测系统 cpu 及内存使用情况如下:

- 1) 当 50% 发送速率拷机, 单条消息字节长度为 1 024 byte 时, 内存占用率和网络负荷负荷率正常;
- 2) 当 80% 发送速率拷机, 单条消息字节长度为 1 024 byte 时, 内存占用率和网络负荷负荷率正常;
- 3) 当 100% 发送速率拷机, 单条消息字节长度为 1 024 byte 时, 内存占用率和网络负荷负荷率较高。

经性能测试, 除了掌握以上性能规律外, 还发现了两个有价值的问题, 问题 1: 当向消息总线服务器发送海量数据, 导致服务器存储空间不足时, 消息总线服务直接崩溃; 问题 2: 向消息总线服务器长时间发送 1 024 byte 大小的数据流时, 服务器内存占用过大。以上性能数据及问题得到了设计人员的高度认可, 充分体现了文中设计的测试框架的价值。

### 5 结论

随着配电信息数据规模的不断扩大, 对消息总线的性能要求越来越高, 而数据吞吐量和数据处理效率是消息总线性能的重要指标。本文通过对 kafka 为服务主体的消息总线性能进行分析后, 提出了效果较好的测试方法, 设计了实用性很强的测试框架。经实际应用证明, 该测试框架简单易用, 在很大程度上提高了性能测试的效率。该框架有较好的通用性, 其他消息总线的测试也可以使用。

### 参考文献:

- [1] 马浩. 配网自动化系统的研究 [D]. 济南: 山东大学, 2011, 2-55.
- [2] H Liu, S Ge. Research on Power Supply Model of medium voltage distribution system [A]. Innovative Smart Grid Technologies - Asia (ISGT Asia), 2012 IEEE [C]. IEEE, 2012; 1-5.
- [3] 温东旭, 杨辉, 王旭宁, 等. 电力系统保护装置保障性自动测试方案研究与实践 [J]. 电力系统保护与控制, 2015, 43(10): 135-138.
- [4] 纪陵, 李忠明, 蒋衍君, 等. 智能变电站二次系统仿真测试和集成调试新模式的探索和研究 [J]. 电力系统保护与控制, 2014, 42(22): 119-123.
- [5] 笃峻, 叶翔, 王长瑞, 等. 智能变电站设计配置一体化功能规范研究及工具开发 [J]. 电力系统自动化, 2014, 38(20): 85-89.