

海量电能质量数据交换格式文件快速解析方案

张逸¹, 杨洪耕²

(1. 福建省电力有限公司电力科学研究院, 福建 福州 350007;

2. 四川大学 电气信息学院, 四川 成都 610065)

摘要: 提出了一种针对海量电能质量数据交换格式文件的快速解析方案。通过快速判断通道序列定义是否变化来避免重复解析相同的数据源记录。利用文件读取线程分解出观测值记录填入任务池中, 观测值记录解析线程循环访问任务池, 读取观测值记录进行解析后放入数据缓存池, 实现了多线程并行解析。基于支持并行扩展的.NET 4.0 编程模型实现了所提方案。对现场海量电能质量数据交换格式文件进行解析测试, 测试结果显示该方案可显著提高解析效率, 并且加速比随 CPU 核数的增加和数据文件的增多而提高。在四川电能质量一体化数据平台中的成功应用证明了该方案的可靠性和实用性。

关键词: 电能质量; PQDIF; 海量数据; 数据处理; 线程; 并行; 解析; .NET 4.0

中图分类号: TM 734

文献标识码: A

DOI: 10.3969/j.issn.1006-6047.2013.12.019

0 引言

随着电能质量监测系统规模的不断扩大, 缺乏标准统一的电能质量数据存储体系所造成的监测数据管理困难问题越发严重。IEEE 1159.3 标准中提出的电能质量数据交换格式 PQDIF (Power Quality Data Interchange Format)^[1]完全独立于监测设备的软硬件, 不仅能较好地解决多数据源数据兼容问题, 还可以实现电能质量物理属性的多角度观察功能, 符合电能质量监测技术的发展需要^[2-3], 因此被普遍作为电能质量监测系统中监测数据交换的标准格式^[3-4]。

目前电能质量监测系统普遍采用 2 层式架构^[5-7]。在此架构下, 监测主站按周期采用 FTP 等协议直接召唤各监测装置生成的 PQDIF 文件^[8], 并完成随后的文件解析、数据分析、储存和发布等工作。与设置区域子站的 3 层结构相比, 2 层式架构结构简单, 便于管理, 且减少了 1 层设备的建设投资和运维费用, 经济性更好, 但也存在监测主站负担重的问题。随着电能质量监测系统建设工作的推进, 监测点的数量持续增加, 某些地区的监测系统已包含数千个监测点^[5-6]。监测装置周期性地为每个监测点生成一个 PQDIF 文件, 这就意味着监测主站需要同时解析数千个 PQDIF 文件。这种数据文件海量化的趋势已经给传统的基于串行编程方式设计的解析程序带来了很大困难, 其中最显著的问题是解析时间过长、大量资源被占用从而导致监测主站长时间无法正常响应用户请求, 甚至造成主站系统崩溃。目前国内外文献中尚无对上述问题的研究, 因此迫切需要一种针对海量 PQDIF 文件的快速解析方案。

随着计算机硬件技术的发展, 传统的面向单核

的串行编程技术已逐渐被基于多核多线程模式的并行编程技术所取代^[9]。作为实现高速计算的一种重要途径, 并行编程技术为以上问题提供了一种全新的解决方案。

基于上述情况, 本文提出了一种针对海量 PQDIF 文件的快速解析方案。针对传统方案中需要重复解析相同数据源的情况, 提出了 3 种可行方法以快速判断通道序列定义是否改变, 仅解析改变了的数据源记录; 采用并行编程和缓存池技术, 实现了利用多线程的海量观测值记录并行解析。通过分析在不同 CPU 核数、不同 PQDIF 文件数量情况下的解析耗时证明了此方案的实用性。目前, 本文方案已在四川电能质量一体化数据平台中成功应用。

1 PQDIF 文件结构与传统解析方案

1.1 PQDIF 文件结构

PQDIF 文件结构分为物理层和逻辑层。物理层结构定义了基本和复杂的数据类型在文件中的构造形式, 逻辑层结构定义了电能质量数据用物理层数据的诠释方式。

从物理结构上看, PQDIF 文件由一系列的记录所组成, 每个记录均包含记录头和记录体。记录头中包括记录类型、大小以及指向下一个记录的链接。记录体由集合、标量和向量这 3 种类型元素组成: 集合可包含标量、向量, 还可以包含集合; 标量指特定物理类型的单值; 向量为任意大小的数列^[1]。

PQDIF 文件中, 数据源记录和观测值记录是最重要的 2 种记录, 两者相互关联, 前者为后者提供测量值所对应的电能质量指标类型、单位以及相位等信息, 后者根据前者中对应的定义保存实际测量值。在电能质量监测系统中, 监测装置投入实际运行后, 主

站一般也仅解析其上传的 PQDIF 文件中的数据源记录 and 观测值记录。

1.2 传统解析方案及其存在的问题

传统 PQDIF 文件解析方案一般直接调用 Electrotek Concepts 公司开发的 PQDIF COM(Component Object Model, 组件对象模型), 将目前版本 pqdcom4.dll 注册并导入后, 即可利用已有的应用程序编程接口函数解析文件, 解析单个 PQDIF 文件的流程图如图 1 所示。

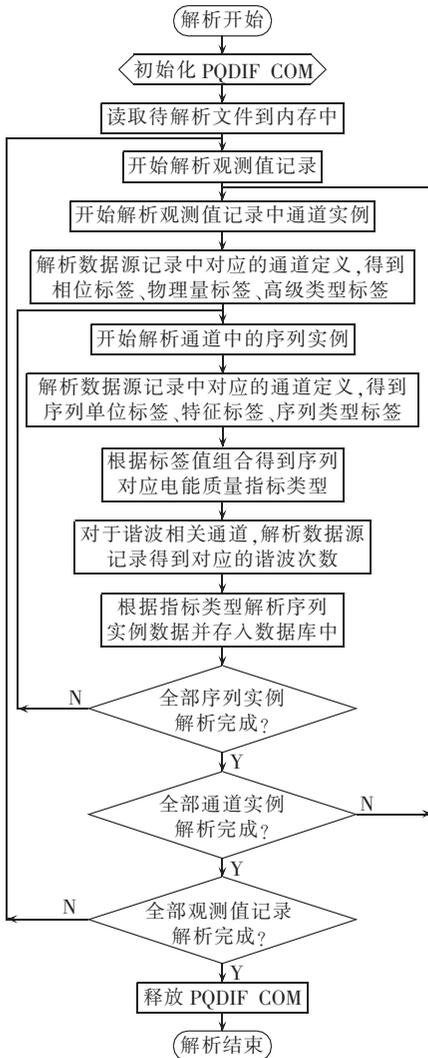


图 1 利用 PQDIF COM 解析单个 PQDIF 文件流程图

Fig.1 Flowchart of single PQDIF file parsing by PQDIF COM

传统的利用 PQDIF COM 的解析方案(下文中简称传统方案)的优点是实现简单, 缺点包括执行速度较慢、编码不灵活、仅支持 Windows 32 位操作系统、处理大文件时内存占用大等^[3,10]。最关键的是, PQDIF COM 自身未提供针对解析多个 PQDIF 文件的优化方法, 只能循环重复单个文件的解析过程, 逐一解析每个文件。

目前, 四川电能质量一体化数据平台(下文中简

称平台)中已包含 343 个监测点, 监测装置每天为每个监测点生成 1 个文件, 文件中包括暂态事件数据和 1 d 内按 1 min 时间间隔保存的统计数据(最大值、最小值、平均值、概率 95 值, 谐波统计到 50 次), 每个文件平均大小约为 3 MB。

考虑到某些监测系统已包含数千个监测点, 笔者利用传统方案测试解析 1 500 个 PQDIF 文件, 所有文件均来自平台中所有监测点不同天的实际监测数据。测试服务器硬件条件为 4 核 2.50 GHz Xeon E5420, 内存 4 GB。由于服务器负载变化, 导致每次解析耗时有不同, 因此采用 K 次最优测量方法^[11]得到解析时间(第 4 节中也采用同样的测量方法), 测试结果如图 2 所示。由于数据库存储时间与所选用的数据库引擎、数据库设计以及存储策略有关, 故未计入总耗时中。

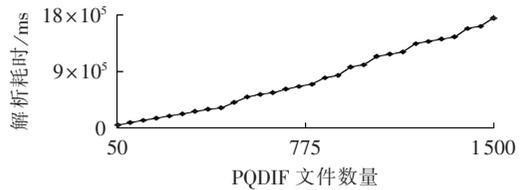


图 2 传统方案解析耗时

Fig.2 Time consumption of traditional parsing schemes

由测试结果可知, 传统方案耗时大约为 0.5 h (1 750 301 ms), 实际情况中需考虑存储数据库的时间, 耗时将更长。这将导致监测主站长时间无法响应用户请求, 而且测试过程中也有系统崩溃的情况出现, 因此, 传统方案已经无法适应目前电能质量监测系统中解析海量 PQDIF 文件的需求。

2 海量 PQDIF 文件快速解析方案

2.1 直接编码解析方法

针对传统方案的缺点, 本文方案没有使用 PQDIF COM, 而是利用标准中提供的 C 语言头文件^[1], 采用直接编码的方式, 自行编写解析方法。其优势包括内存使用效率高、执行速度快、平台通用性强、可方便地进入方法内部调试、便于软件的升级与维护等, 更重要的是此方案编程灵活^[3], 便于程序的优化设计。下面就分别从两方面针对海量 PQDIF 文件解析进行优化。

2.2 数据源记录快速解析方法

在解析 PQDIF 文件中各序列实例的数据时, 需要先获得其在数据源记录中对应的标签值组合, 从而得到此数据对应的电能质量指标类型(如图 1 所示)。在实际已建成的电能质量监测系统中, 一般通过监测装置接入规范规定了装置需采集的指标类型, 故数据源记录相对固定, 而且装置投入正常运行后, 每天生成文件中数据源记录的通道序列定义也

应完全相同,因此,预先构造通道序列定义表(以下简称定义表)保存已知的定义。在解析程序初始化时,将定义表读入内存,随后对装置上传文件中通道序列定义是否改变进行快速判断,如果未改变,则直接利用内存中的定义表得到序列实例对应的指标类型,仅在在有变化的情况下才重新解析数据源记录,更新定义表,从而避免重复解析相同的数据源记录,减少了解析耗时。

新型监测装置接入时,预先解析数据源记录,得到每个序列定义的 6 种标签组合(如图 1 中所示),并唯一地确定序列定义的指标类型,保存在定义表中。考虑到不同类型装置的通道序列定义可能不同,故需在表中保存装置类型标识,也可通过在监测装置接入规范中强制规定数据源记录形式来解决,但此方法通用性稍差。定义表如表 1 所示。

表 1 通道序列定义表
Tab.1 List of channel series definition

序列名	描述
装置类型 ID	装置类型的唯一标识,用于区别不同类型装置对应的通道序列定义
通道序号	从 0 开始的序号,与观测值记录中通道实例的 tagChannelDefIdx 标签值对应
序列索引	从 0 开始的索引,标识序列在通道中的位置
数据类型	此序列定义对应的唯一电能质量指标类型

PQDIF 文件中每个记录的记录头中均保存有 1 个校验和(32 bit 的循环冗余码校验)来保证物理结构的正确性和完整性^[1],2 个完全相同的记录体的校验和相同,故可利用校验和来快速判断数据源记录是否变化,但存在个别装置在除通道序列定义外的其他信息中利用 tagEffective 标签值表示数据源生效时间或在 tagNameDS 标签中保存监测点名称等情况,将造成数据源记录校验和变化。此时,可通过计算 tagChannelDefns 及随后的集合数据块的校验和来判断通道序列定义是否有改变。此时应注意,如果是 tagNameDS 这类向量的数列大小发生变化(如监测点名称长度变化),将会导致其后集合在记录体中的相对地址发生偏移(即 link.linkElement 值改变),故此时需减去偏移量后再计算校验和。

因此,针对如何快速判断通道序列定义是否发生变化,有以下 3 种可行的方法。

a. 在监测系统装置接入规范中规定只有当数据源记录需要改变时,装置才在上传的文件中包含数据源记录。

b. 规范各装置上传的数据源记录的内容(不包含随时间或监测点变化的信息),通过比较记录头的校验和快速判断数据源记录是否改变。

c. 无需对监测装置进行规范,对数据源记录进行初步解析得到 tagChannelDefns 及其后的集合数

据块,将其中 LinkElement 值减去偏移量后计算数据块校验和,比较判断其中定义是否发生变化。

3 种方法解析耗时和实现复杂程度依次增加,但通用性也随之提高,可根据监测系统实施的实际情况进行选择,本方案采用方法 c。

2.3 观测值记录多线程并行解析方法

目前,1 个 PQDIF 文件中通常包含数个观测值记录,考虑到时间序列的差异,一般分为除闪变外的稳态统计数据(谐波、电压偏差、三相不平衡等)、闪变相关数据(电压波动、短时闪变和长时闪变)以及暂态事件数据(电压暂降、短时中断等)3 类记录,以便共用时间序列,节约存储空间。由于各观测值记录序列实例之间没有数据依赖关系,因此可独立并行解析。并行模式中,并行数据的粒度选择非常重要,其与领域的特征和算法的数据依赖关系密切,同时要权衡普适性和实现难度^[12]。方案设计初期曾考虑参考文献[13]中并行解析电力系统暂态数据交换通用格式(COMTRADE)文件的数据分块方法,将所有观测值记录平均分成固定大小的数据块交由各线程独立解析,并部署专门的线程对块边缘进行合并解析,此方法并行粒度高,且各线程负载较均衡。但测试后发现,由于 PQDIF 结构较 COMTRADE 复杂很多,数据块边缘合并困难,不但易出错,而且耗时较长,甚至出现解析线程等待边缘合并完成的情况,造成了额外的同步等待耗时,而且,专门设置边缘解析线程也较浪费系统资源。因此,本方案直接选择观测值记录整体作为并行数据粒度,虽然对于少量文件,并行程度较低,但针对海量文件,各线程之间负载的细微差异基本可以忽略,而且程序实现简单,不易出错。由于时间序列解析在观测值记录内部解析过程中进行,因此,不同类型数据的采集时间间隔差异并不影响解析效率。并行解析原理图见图 3。

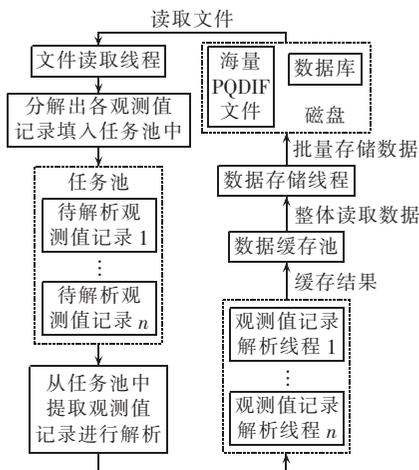


图 3 多线程并行解析观测值记录原理图
Fig.3 Schematic diagram of multi-threading parallel parsing of observation records

本文方案专门部署一个 PQDIF 文件读取线程 FRT(File Reading Thread),其与观测值记录解析线程 OPT(Observation Parsing Thread)形成流水线并行,并使用任务池协调其速度差异,由于读取文件,解析数据源记录,并分解出各观测值记录的耗时仅为解析观测值记录耗时的 8% 左右,因此,此优化机制能为 OPT 提供持续不断的数据。

FRT 分解得到各观测值记录后,填入任务池中,随后继续读取下一个文件。这里有 2 种方法可供选择:第 1 种是 FRT 将各观测值记录的所属文件地址和记录绝对地址索引(前一记录头中的 linkNextRecord 值)填入任务池中,各 OPT 根据地址和索引读取记录;第 2 种是 FRT 直接将分解出的未解压的观测值记录数据块填入任务池中。测试后发现,第 1 种方法虽然减少了任务池的内存占用,但各线程并行读取文件,将导致访问文件的位置有很强的随机性,无法利用文件系统的预取优化,降低了磁盘的访问效率;第 2 种方法中每个观测值记录解压前的大小仅为 100 KB 左右,并不会大量占用内存。综合考虑,本文采用第 2 种方法。

OPT 循环访问任务池,取走待解析的观测值记录,解压后独立并行解析。为了减少有限存储带宽的竞争,将载入内存的定义表拷贝到各线程局部变量中。OPT 完成解析观测值记录后并不直接将结果存入数据库中,而是将其放入数据缓存池中,待缓存数据达到一定容量后,由专门的数据存储线程 DST(Data Saving Thread)一次性批量插入数据库中,这样不但避免了多个 OPT 频繁操作数据库造成的堵塞,还能够更好地发挥批量索引的插入优势。

3 解析方案实现

3.1 编程环境选择

根据四川电能质量一体化数据平台采用 Windows Server 2008 操作系统的实际情况,本文方案基于微软最新的 .Net Framework 4.0(以下简称“.NET 4.0”)编程模型,选择 Visual Studio 2010 作为编程开发工具,它们提供了新的运行时、新的并行扩展类以及新的诊断工具,增强了对并行编程的支持,简化了并行开发^[14]。解决方案采用微软推荐最适合 .NET 环境的 C# 语言作为并行解析项目的开发语言,为了利用标准中提供的 C 头文件^[1],利用 C++/CLI 编写底层 PQDIF 文件解析函数,将其作为类库项目包含在整体解决方案中,以便与 C# 项目同时调试,并引用其中定义的各 PQDIF 逻辑结构。单个通道定义结构体声明代码如下:

```
public value struct OneChanelDefn
```

```
{
    String^ tagChannelName; //通道名称
    ID tagPhaseID; //相位标签
    GUID tagQuantityTypeID; //高级类型标签
    ID tagQuantityMeasuredID; //物理量标签
    Collections::Generic::List<OneSeriesDefn>^ tagSeriesDefns;
    //此通道定义包含的序列定义链表
};
```

3.2 数据源记录快速解析实现

预先构造的定义表可利用数据库或 XML 文档存储。解析程序初始化时,将其载入内存中并转化为哈希表结构,可利用 Hashtable 类实现。利用装置类型 ID、通道序号以及序列索引组合出唯一的 Key 值,随后可通过 Key 值快速检索对应的指标类型。

第 2.2 节提到的 3 种方法中,前 2 种方法的实现较为简单,主要功能 C++/CLI 代码如下:

```
c_record_mainheader* pHeader=(c_record_mainheader*)&pByte[0];
//得到首个记录头指针,其中 pByte 为读入内存的 PQDIF 文件指针
LINKABS4 link=pHeader->linkNextRecord; //下一个记录地址
pHeader=(c_record_mainheader*)&pByte[link]; //得到下一个记录头
if(PQDIF_IsEqualGUID(pHeader->tagRecordType,tagRecDataSource))
//判断是否是数据源记录,可遍历全部记录判断是否包含数据源记录
{UINT4 crcChecksum=pHeader->checksum; //得到记录头中校验和
```

方案 c 实现较复杂,具体流程图如图 4 所示,在装置初次接入时需要用同样方法计算并保存校验和以及 tagChannelDefns 集合中的 linkElement 值。

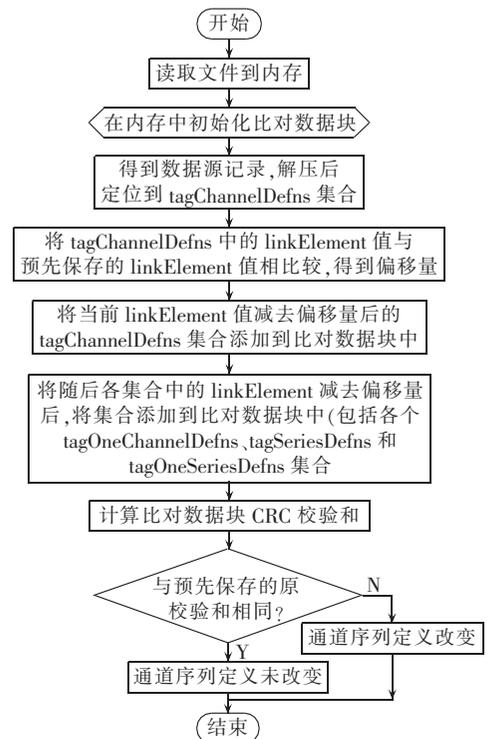


图 4 方案 c 实现流程图

Fig.4 Implementation flowchart of method c

3.3 多线程并行解析观测值记录实现

在 .NET 4.0 中的并行编程是依赖任务并行库

TPL(Task Parallel Library)实现的,其最基本的执行单元是任务,代表了一个可以被计算机并行执行的异步操作^[14]。TPL 负责创建并管理线程来执行任务,以下 C# 代码创建并运行一个观测值记录解析任务:

```
Action<object>ParseObservation=delegate(object observationRecord)
{ //具体观测值记录解析代码...}; //定义任务中并行解析代码的委托
Task taskParseOb=new Task(ParseObservation, observationRecord);
//创建并行解析观测值记录的任务对象
taskParseOb.Start(); //开始并行解析观测值记录
```

为了确保每个线程都能够正确地访问共享存储区(任务池和数据缓存池)内的数据,传统方法需要在程序中采用封锁机制^[15],不但编程复杂,而且增加了出错风险,降低了程序效率。本方案采用.Net 4.0 新增的 System.Collections.Concurrent 命名空间中提供的线程安全集合类构造共享存储区,可以方便地并发访问其中的数据^[16]。利用线程安全的先进先出队列类 ConcurrentQueue 实现任务池的主要 C# 代码如下:

```
ConcurrentQueue<observationRecord> jobsQueue
=new ConcurrentQueue<observationRecord>(); //创建任务池实例
jobsQueue.Enqueue(observationRecord newRecord);
//FRT 中使用,将分解出的待解析观测值记录添加到任务池中
bool isSuccessed=jobsQueue.TryDequeue(out observationRecord);
//OPT 中使用,取出目前任务池开始处首个待解析的观测值记录
```

方案中还利用 ManualResetEventSlim 类协调 OPT 与 FRT,以免写入的观测值记录超过任务池预设大小;利用 Task 类的 WaitAll()方法保证所有 OPT 均完成后结束解析;利用 AggregateException 类处理并行解析过程中的异常;使用 SqlBulkCopy 类实现大批量数据快速导入数据库等,篇幅所限,在此就不全部展开详述。

4 测试结果

笔者将本文所提方案在四川电能质量一体化数据平台中应用实施,并用现场 PQDIF 文件对方案实用性和可靠性进行验证。测试平台如表 2 所示。

表 2 测试平台软硬件条件

Tab.2 Software and hardware conditions of test platform

服务器型号	CPU 型号	描述	内存	操作系统
HP xw8600	Xeon E5420 2.50 GHz	4 核	4 GB	Windows Server 2008
IBM System x3650 M3	Xeon X5675 3.06 GHz	6 核	4 GB	Windows Server 2008
IBM System x3850 X5	Xeon X7560 2.26 GHz	8 核	4 GB	Windows Server 2008

在与第 1.2 节中相同的测试条件下(4 核服务器),本文方案与传统方案解析耗时对比如图 5 所示。

由图 5 可见,本文方案相对传统方案解析耗时明显减少,并可随着 PQDIF 文件数量的增加获得更高的加速比,解析 1500 个文件的耗时可控制在 5 min

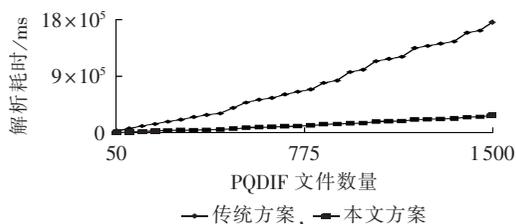


图 5 快速解析方案与传统方案解析耗时对比

Fig.5 Comparison of parsing time consumption between proposed scheme and traditional scheme

以内(282306 ms)。

由测试结果可知,由于本文方案直接选择观测值记录整体作为并行解析的数据粒度,因此可显著提高存储在观测值记录中的每种类型数据(包括稳态数据和暂态数据)的解析效率。各类型数据的解析耗时对比如表 3 所示。

表 3 各类型电能质量监测数据解析耗时对比

Tab.3 Comparison of parsing time consumption for different types of power quality monitoring data

监测数据类型	解析耗时/ms		加速比
	传统方案	本文方案	
谐波类数据	1302150	239806	5.43
间谐波类数据	95404	18473	5.16
电压偏差	39001	7847	4.97
频率偏差	20080	4165	4.82
电压(电流)值	39887	8107	4.92
三相不平衡类数据	35467	6954	5.10
电压波动与闪变类数据	4871	1130	4.31
暂态类数据	86402	15212	5.68

表 3 中,某类型数据的解析耗时为 1500 个文件中此类数据解析耗时的总和,且解析耗时为观测值记录中各类型数据的解析时间,不包括数据源记录的解析耗时,均采用 K 次最优测量方法得出;谐波类数据包括 2~50 次谐波电压幅值(相角)、谐波电流幅值(相角)、谐波有功(无功)功率以及谐波电压(电流)总畸变率;间谐波类数据包括前 25 个间谐波组电压(电流)幅值;三相不平衡类数据包括正序(负序、零序)电压(电流)值、电压(电流)负序不平衡度;电压波动与闪变类数据包括:电压波动值、短(长)时闪变值。暂态类数据包括电压暂降(升)事件有效值记录、电压暂降(升)事件波形采样值记录、短时中断事件有效值(波形采样值)记录。

测试解析少量 PQDIF 文件时,本文方案也可提高解析效率,但由于本文方案选取的并行粒度较大,少量文件的并行程度较低,影响了解析效率的提高。随文件增多,解析效率的提高将更为显著。少量文件的解析耗时对比如表 4 所示。

本文方案在不同核数服务器条件下的解析耗时对比如图 6 所示。由图 6 可见,随着服务器 CPU 核数的增加,本文方案性能提升显著。解析 1500 个文件时,采用 8 核服务器的耗时(150966 ms)相对 4

表 4 解析少量 PQDIF 文件耗时对比

Tab.4 Comparison of parsing time consumption between proposed scheme and traditional scheme for different PQDIF file quantities

PQDIF 文件数量	解析耗时/ms		加速比
	传统方案	本文方案	
1	513	208	2.47
2	1145	386	2.97
3	1863	605	3.08
4	2516	841	2.99
5	3188	1059	3.01
10	6603	2116	3.12
20	13266	4224	3.14
50	35678	8919	4.00

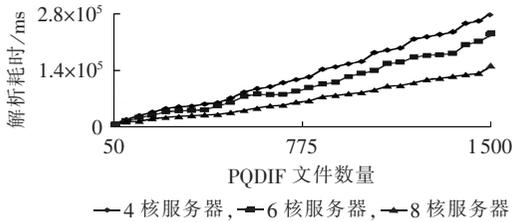


图 6 不同 CPU 核数服务器条件下快速解析方案解析耗时对比测试结果

Fig.6 Comparison of parsing time consumption among different CPU core quantities

核服务器减少了约 46.52%,因此,本文方案具有一定的可扩展能力。

目前,基于本文方案的解析程序已在平台中长期稳定运行。程序每日凌晨自启动并解析主站采集的 343 个 PQDIF 文件,耗时约 40 s,可满足实际需要,且运行过程中未发生服务器操作系统崩溃、内存溢出等错误,证明了本文方案的可靠性。

5 结语

本文提出一种海量 PQDIF 文件的快速解析方案,相对于传统方案有以下几点优势:解析效率高,速度快;硬件可扩展性强;程序稳定性高;软件可维护性强。

目前此方案已有效应用于四川电能质量一体化数据平台。实践证明,此方案具有较高的实用性和可靠性。但值得注意的是,解析后数据存入数据库的耗时也是相当可观的,因此,下一步将研究海量电能质量数据的存储策略。

参考文献:

[1] IEEE Standards Coordinating Committee 22 on Power Quality. IEEE Std1159.3-2003 IEEE recommended practice for the transfer of power quality data[S]. New York,NY,USA:IEEE,2004.

[2] 杨进,肖湘宁. 电能质量监测技术发展新趋势——网络化、信息化、标准化[J]. 电力自动化设备,2004,24(11):82-87.

YANG Jin,XIAO Xiangning. New development tendency of power quality supervision technology-netlike,informization and standardization[J]. Electric Power Automation Equipment,2004,24(11):

82-87.

[3] 张逸,杨洪耕,林瑞星,等. 电能质量数据交换格式文件通用解析方案[J]. 电力系统自动化,2012,36(11):86-91.

ZHANG Yi,YANG Honggeng,LIN Ruixing,et al. A general parsing scheme for power quality data interchange format files[J]. Automation of Electric Power Systems,2012,36(11):86-91.

[4] 王芳,顾伟,袁晓冬,等. 面向智能电网的新一代电能质量管理平台[J]. 电力自动化设备,2012,32(7):134-139.

WANG Fang,GU Wei,YUAN Xiaodong,et al. Power quality management platform for smart grid[J]. Electric Power Automation Equipment,2012,32(7):134-139.

[5] 何维国,谢伟,张健. 上海电网电能质量监测系统[J]. 华东电力,2010,38(4):497-499.

HE Weiguo,XIE Wei,ZHANG Jian. Power quality monitoring system of Shanghai Grid[J]. East China Electric Power,2010,38(4):497-499.

[6] 韦永忠,郑爱霞,袁晓冬,等. 江苏电网电能质量评估方法及应用[J]. 江苏电机工程,2011,30(5):1-4.

WEI Yongzhong,ZHENG Aixia,YUAN Xiaodong,et al. Application of power quality assessment in Jiangsu Power System[J]. Jiangsu Electrical Engineering,2011,30(5):1-4.

[7] 范瑞祥,辛建波,上官帖,等. 浙赣电气化铁路江西段开放式电能质量监测系统的开发与应用[J]. 电网技术,2008,32(21):68-74.

FAN Ruixiang,XIN Jianbo,SHANGGUAN Tie,et al. Development and application of open-type power quality monitoring system for Jiangxi section of Zhejiang-Jiangxi electrified railway[J]. Power System Technology,2008,32(21):68-74.

[8] 朱伟立,王俊,王巍,等. PQDIF 在嵌入式电能质量监测终端上的实现与应用[J]. 江苏电机工程,2011,30(2):59-62,65.

ZHU Weili,WANG Jun,WANG Wei,et al. Realization and application of PQDIF in embedded power quality monitor terminal[J]. Jiangsu Electrical Engineering,2011,30(2):59-62,65.

[9] AKHTER S,ROBERTS J,REINDERS J,et al. Multi-core programming:increasing performance through software multi-threading[M]. Hillsboro,Dexas,USA:Intel Press Business Unit,2004:2-10.

[10] 聂晶晶. 电能质量监测系统及 PQDIF 数据存储格式的研究[D]. 南京:东南大学,2006.

NIE Jingjing. Research on power quality monitoring system and PQDIF data storage format[D]. Nanjing:Southeast University,2006.

[11] BRYANT R E,O'HALLARON D. 深入理解计算机系统[M]. 龚奕利,雷迎春,译. 北京:中国电力出版社,2004:68-72.

[12] 赵长海,晏海华,王宏琳,等. 面向地震数据处理的并行与分布式编程框架[J]. 石油地球物理勘探,2010,45(1):146-155.

ZHAO Changhai,YAN Haihua,WANG Honglin,et al. Seismic data processing oriented parallel and distributed programming framework[J]. Oil Geophysical Prospecting,2010,45(1):146-155.

[13] 桂勋,姚兰,钱清泉,等. 海量 COMTRADE 数据并行解析算法[J]. 电力系统自动化,2009,33(15):62-66.

GUI Xun,YAO Lan,QIAN Qingquan,et al. A massive COMTRADE data parallel parsing algorithm[J]. Automation of Electric Power Systems,2009,33(15):62-66.

[14] HILLAR G C. C# 并行编程高级教程:精通.NET4.Parallel Extensions[M]. 郑思遥,房凤慈,译. 北京:清华大学出版社,2011:15-26.

[15] 苏振强,鲁先平,石乐明,等. 分布式多线程并行处理技术在大规模化学结构检索数据库系统中的应用[J]. 计算机与应用化学, (下转第 127 页 continued on page 127)

工程师,硕士,研究方向为互感器、电能计量(**E-mail**:lsf117@tom.com);

杨世海(1976-),男,安徽淮北人,高级工程师,硕士,研究方向为互感器、电能计量(**E-mail**:ysh.young@163.com);

陈刚(1984-),男,湖北黄梅人,工程师,硕士,研究方向

为互感器、电能计量(**E-mail**:cg09_jsepre@sina.com);

徐敏锐(1975-),男,江苏涟水人,高级工程师,硕士,研究方向为互感器、电能计量(**E-mail**:xuminrui18@163.com);

周玉(1982-),男,江苏镇江人,工程师,硕士,从事互感器、电能计量方面的研究工作(**E-mail**:longhu205@163.com)。

Research of high AC and DC hybrid generator

WANG Zhongdong¹, LUO Pandian², LU Shufeng¹, YANG Shihai¹,

CHEN Gang¹, XU Minrui¹, ZHOU Yu¹

(1. Jiangsu Electric Power Company Research Institute, Nanjing 211103, China;

2. College of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: A high AC and DC hybrid generator is developed, which consists of four parts: AC source, DC source, current combination device and monitoring & analysis system. The working principles and functions of each part are introduced. Large-capacity capacitors are serially connected in AC circuit to prevent DC. There is a common branch of AC and DC circuits in the hybrid generator to superpose AC and DC together, without interfering in the AC or DC source. Results of performance test show that, the AC circuit and DC circuit are immune to each other and the high current flowing through the tested current transformer contains both AC and DC.

Key words: AC; DC; capacitors; current transformer

(上接第 121 页 continued from page 121)

2004, 21(5):659-664.

SU Zhenqiang, LU Xianping, SHI Leming, et al. The application of multithreading parallel and distributed computing scale chemical structure searching database[J]. Computers and Applied Chemistry, 2004, 21(5):659-664.

作者简介:

张逸(1984-),男,福建福州人,博士,主要从事电能质量分析与控制系统开发(**E-mail**:zhangyiscu@163.com);

杨洪耕(1949-),男,四川成都人,教授,博士研究生导师,主要从事电能质量分析与控制的研究与教学工作。

Fast parsing of massive PQDIF files

ZHANG Yi¹, YANG Honggeng²

(1. Fujian Electric Power Research Institute, Fuzhou 350007, China;

2. School of Electrical Engineering and Information, Sichuan University, Chengdu 610065, China)

Abstract: A fast parsing scheme is proposed for PQDIF(Power Quality Data Interchange Format) files, which quickly judges the change of channel series definitions to avoid the repeated parsing of records from same data source. The file reading thread decomposes the observation records and stores them in the task pool while, in parallel, the parsing thread circularly reads and parses the observation records of the task pool and then stores them in the data buffer pool. The proposed scheme is implemented based on the .NET 4.0 programming model which supports the parallel extensions. Test with field massive PQDIF files shows that, the parsing efficiency is significantly improved and the acceleration ratio increases along with the increase of CPU core and data files. Its successful application in Sichuan power quality integrated data platform proves its reliability and practicality.

Key words: power quality; PQDIF; massive data; data processing; thread; parallel programming; parsing; .NET 4.0