

基于流水线技术的虚拟仪器运行性能提升

赵江滨¹, 刘世元^{1,2}, 轩建平¹, 胡友民¹

(1. 华中科技大学机械科学与工程学院, 武汉 430074; 2. 武汉光电国家实验室(筹), 武汉 430074)

摘要: 虚拟仪器的软件系统串行处理结构适合采用流水线技术来提升运行性能。根据该特性, 研究连续采样的虚拟仪器系统的工作过程, 以及自适应流水线的吞吐率与延迟 2 个性能指标, 结合实例测试运行性能。结果显示, 流水线技术能在不增大延迟的情况下, 大幅提高连续采样的虚拟仪器软件系统的吞吐率。

关键词: 虚拟仪器; 流水线; 吞吐率; 延迟

Improvement of Virtual Instrument Performance Based on Pipeline Technology

ZHAO Jiang-bin¹, LIU Shi-yuan^{1,2}, XUAN Jian-ping¹, HU You-min¹

(1. School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074;

2. Wuhan National Laboratory for Optoelectronics, Wuhan 430074)

【Abstract】 Software sequential structure of virtual instrument makes it suited to apply pipeline technology to improve its performance. This paper investigates the running process of continuous sampling virtual instrument system, and introduces throughput and latency, the two performance measures of a self-timed pipeline. It gives an experimental example, result shows that, pipeline technology can greatly improve continuous sampling virtual instrument software system throughput, while not degrade its latency.

【Key words】 virtual instrument; pipeline; throughput; latency

1 概述

虚拟仪器是测试测量领域的一项重要技术, 秉承“软件就是仪器”的理念, 充分利用 PC 的计算、存储、通信、显示等资源, 借助于模块化的测试硬件, 通过软件灵活地实现各种测试测量功能, 克服了传统台式仪器功能固定、缺乏灵活性的缺点, 是一种可重构系统。

由于 PC 具有模块化的体系结构, 因此虚拟仪器系统的软硬件技术发展相对独立, 同时又紧密结合了以实现测试测量功能并满足要求的性能指标。近年来, 测试硬件系统的性能发展很快, 采样以及与 PC 通信的能力大幅提高, 从而对软件系统的数据处理能力(吞吐率)提出了更高的要求。虚拟仪器软件系统对硬件系统采集的数据进行各种分析处理以及显示与存储等操作, 包括多个处理步骤, 是一种数据单向流动的串行结构。虚拟仪器系统的功能越丰富, 所需的处理步骤越多, 完成一次数据处理的耗时也就越多, 在一般情况下, 吞吐率会相应降低。

流水线(pipeline)是一种以串行结构实现并行效率的重要技术, 能够大幅度地减少平均处理时间, 从而提高吞吐率, 在软硬件系统中广为采用, 如微处理器、Web 服务器软件^[1-3]。在笔者开发的 IsVIP 虚拟仪器平台中就应用了流水线技术, 并且在同一运行环境中与标准模式进行了对比测试, 结果表明, 流水线能够显著提高吞吐率, 且步骤越多, 提升效果越明显^[4-5]。但其性能提升的理论极限^[4]与测试系统的特殊运行状态^[5]未得到深入研究。本文将研究应用流水线技术提升虚拟仪器软件的运行性能, 使之能够与不断提升的硬件系统性相匹配, 实现连续采样的在线测试系统不间断地持续运行。

2 虚拟仪器的工作过程

虚拟仪器是一种串行的、多步骤的数据采集与分析处理过程, 数据依次在步骤间单向流动, 各个步骤按照确定的次序依次执行, 完成测试测量任务, 基于 DAQ 的虚拟仪器系统如图 1 所示。整个过程周而复始地重复运行。

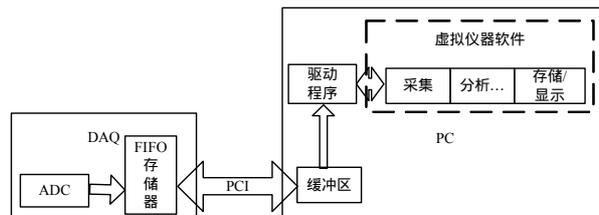


图 1 虚拟仪器系统结构

许多测试系统对信号质量要求较高, 在很多情况下要求不间断地连续采样, 并保持样本之间严格的时序关系, 这需要利用硬件时钟控制 DAQ 设备的采样过程。在此模式下搭建在线测试系统, 要求软件系统及时的读取并处理数据, 具有一定的实时性要求。DAQ 设备的采样性能越高, 软件系统

基金项目: 国家自然科学基金资助项目“面向制造系统的高可靠性柔性可重构自组织 E-诊断技术”(50575087); 国家自然科学基金资助项目“基于小波的机械故障警戒机方法与应用”(50675076); 教育部新世纪优秀人才支持计划基金资助项目(NCET-06-0639)

作者简介: 赵江滨(1976 -), 男, 博士后, 主研方向: 自动测试系统, 传感器网络, 虚拟仪器, 状态监测与智能维护; 刘世元, 教授、博士、博士生导师; 轩建平, 副教授、博士; 胡友民, 教授、博士
收稿日期: 2009-02-17 **E-mail:** zhaojiangbin@163.com

的压力越大。由于 DAQ 不间断的连续采样，采样过程与测试软件的数据处理并行运行，整个测试系统需要采用提供者/消费者工作模式，通过位于 PC 的缓冲区交换数据。

测试系统通常将连续的多个样本作为一个整体进行处理，例如进行 1 024 点的 FFT 频谱分析，就是选择连续的 1 024 个振动信号样本进行 FFT 计算。假设 DAQ 的采样频率设置为 f_s Hz，则样本间隔为 $1/f_s$ s，采集 n 个样本需要耗费的时间为 n/f_s s，测试软件处理 n 点数据的时间必须不超过 n/f_s s，才能使得连续采样的在线测试系统持续运行。例如，某款 DAQ 的采样速率为 1.25 MS/s，则采样间隔为 0.8 μ s，采集 1 024 个样本所需的时间稍大于 0.8 ms，软件系统需要在此时间内完成所有的分析处理。由于缓冲区的存在，在大多数测试系统应用场合，并不严格要求每次数据处理时间均不超过数据采集时间，只要平均处理时间满足这一要求，使得缓冲区内的数据得到及时处理，不致被新的采样数据覆盖即可。

许多测试系统包含多条数据处理分支，每条分支包含多个处理步骤，一般来说，测试系统的功能越多，则分支越多，步骤越多，耗时也越多。有些虚拟仪器软件开发工具能够为每条分支建立独立的线程，使得多条分支并行工作，一定程度上消除了多分支带来的影响。而要减少乃至消除单一分支内多步骤的影响，则需要采用并行流水线技术。并行流水线通过使处理步骤并行运行，减少了数据平均处理时间，提高了吞吐率。

3 并行流水线性能分析

在串行结构中，各个步骤按照确定的次序依次运行，前一步骤的计算结果就是后一步骤的输入，如此直至最后一个步骤，获得整个过程的最终结果。如果步骤独立运行，每一步骤完成当前数据的处理后，能够立即着手处理下一数据，而不必等待整个过程处理完毕，能够大幅减少平均处理时间，提高吞吐率，这就是并行流水线技术的基本思想。

流水线有多种同步(synchronization)方法，最简单的是采用全局时钟进行同步，该方法在硬件系统中简单易行，广为采用。自适应(self-timed)同步是一种更高性能的分布式同步方式，其思想是步骤自身根据数据的可用情况决定是否执行，在软件系统中容易实现。本文的讨论限于自适应流水线过程。

吞吐率与延迟是流水线 2 个重要的性能指标，为了便于说明流水线的性能，本文给出以下几个定义：

(1)步骤处理时间(T_s)：一个步骤实际用于计算处理数据所花费的时间，不包括等待时间。

(2)过程处理时间(T_{total})：过程中的所有步骤处理时间之和。

(3)平均处理时间(T_{avg})：过程处理 1 个数据块平均花费的时间。

(4)吞吐率(Th)：单位时间内处理数据块的数量，等于 $1/T_{avg}$ 。

(5)延迟($Latency$)：数据从进入第 1 个步骤，到通过最后步骤获得最终结果所历经的时间。

(6)数据提供间隔(T_{push})：数据源提供数据的最小间隔。

如果一个流水线过程有 n 个步骤，假设其第 k 个步骤是瓶颈步骤，也就是最耗时步骤， $T_{max}=T_{s_k}$ 。 T_{max} 是流水线是否满负荷工作的临界点， $T_{push}<T_{max}$ 时，流水线满负荷工作； $T_{push}>T_{max}$ 时，则低载工作。流水线满负荷工作时， $T_{avg}=T_{max}$ ，

相对于通常模式，吞吐率提高到 T_{total}/T_{max} 倍，如果步骤时间相等，则提高到 n 倍。流水线轻载工作时， $T_{avg}=T_{push}$ ， T_{max} ，吞吐率提高到 T_{total}/T_{push} 倍，只要 $T_{push}<T_{total}$ ，吞吐率就会有提高。

对于延迟，流水线满负荷工作时，延迟为

$$Latency = k \times T_{max} + \sum_{i=k+1}^n T_{s_i} - \sum_{i=1}^k T_{s_i} \quad (1)$$

可能比通常模式有所增大，如果瓶颈步骤为第 1 步骤，则延迟不变。

而流水线轻载工作时，延迟为

$$Latency = \sum_{i=1}^n T_{s_i} = T_{total} \quad (2)$$

与通常模式时相同。

瓶颈步骤的位置不影响吞吐率。通常模式能够处理的数据间隔为 $[T_{total}, +\infty]$ ，流水线过程能够处理的数据间隔为 $[T_{max}, +\infty]$ ，其中包括通常模式不能处理的间隔为 $[T_{max}, T_{total}]$ 的数据处理请求。进一步地，流水线相对于通常模式的另一个明显优势是：增加处理步骤不会明显的影响吞吐率，只要新增步骤的处理时间不超过原有的瓶颈步骤时间，吞吐率就保持不变。

在流水线满负荷工作时，瓶颈步骤的位置会明显的影响延迟。瓶颈越置后，延迟增大越多，如果处于第 1 步骤的位置，则不会增大；而如果流水线轻载工作，延迟不会增大。

4 实验结果与讨论

在软件系统中，操作系统为并行流水线的每一步骤分配一个独立的线程，使得所有步骤并行工作。但在 PC 中，步骤的处理时间难以保持恒定。多个线程不可能真正的并行工作，尤其在单核 PC 中，每一时刻只可能有 1 个线程真正运行，由操作系统负责调度，线程数量越多，每一线程能够分配到的处理时间越少，相应的步骤处理耗时也越大。PC 的硬件配置也直接影响着步骤处理时间，CPU、内存等关键模块的性能越高，步骤处理时间越少。而由于 DAQ 精确的硬件时钟，采样时间则保持恒定。这样，同样的连续采样在线测试系统，在有些 PC 机能够持续的运行，有些则不能。

为了排除各种干扰因素，揭示流水线技术在测试系统中的性能提升本质，本文进行了理想化处理，使得步骤处理时间基本保持恒定，并通过选取较大的处理时间，减小各种误差带来的影响。实验研究选取了如图 2 所示的双分支/多步骤虚拟仪器系统。

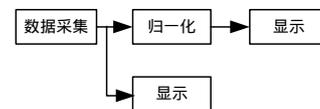


图 2 双分支/多步骤的虚拟仪器系统

测试实验在一台笔记本电脑上进行，其配置为：Intel T7100 1.80 GHz CPU，1 GB DDR2 内存，Windows XP 操作系统；DAQ 设备采用 NI 公司的 USB-6009，设置为采用硬件时钟的连续采样模式，采样频率为 8 kHz，数据块大小取为 1 024，这样，DAQ 采集一个数据块所需的时间为 128 ms。

整个过程有 2 条数据处理流程，分别是：流程 1：数据采集→归一化→显示；流程 2：数据采集→显示。测试程序采用笔者自主研发的即时可重构虚拟仪器平台 IsVIP 搭建^[5]，数据采集、归一化与显示均实现为流水线步骤，2 个显示步骤完全相同。

2 个流程必须在 128 ms 的时间内完成全部处理，才能实

现测试系统的在线持续运行。受置于 DAQ 硬件采样性能的制约,数据采集步骤读取 1 024 点数据块的时间,等于 128 ms,其中主要是等待时间,实际的读取操作耗时极少;后续步骤的处理时间必须不大于 128 ms。即第(1)个步骤也就是数据采集步骤,必须是整个流水线过程的瓶颈。

1 024 点的归一化与显示操作的实际处理时间极短,在本实验环境下不超过 1 ms,为了测试及验证流水线的性能,人为添加了等待时间,使得单个步骤的处理时间小于 128 ms,而归一化与显示步骤的时间之和大于 128 ms,这样,通常的处理模式将无法持续运行。

本文将归一化与显示步骤分别增加了 100 ms 或 60 ms 的等待时间,交叉互换进行了 2 组测试。

测试 1: $T_{s_{\text{归一化}}}=60 \text{ ms}$, $T_{s_{\text{显示}}}=100 \text{ ms}$ 。

如果采用常规运行模式,则所需的处理时间为

$$T_{s_{\text{归一化}}}+2 \times T_{s_{\text{显示}}}=260 \text{ ms}$$

远远大于 128 ms,不能持续运行。如果采用多线程模式(每流程赋予 1 个线程)则流程 1 需要时间: $T_{s_{\text{归一化}}}+T_{s_{\text{显示}}}=160 \text{ ms}$; 流程 2 需要时间: $T_{s_{\text{显示}}}=100 \text{ ms}$ 。由于流程 1 不能在 128 ms 内完成处理,整个过程仍然不能持续运行。

测试 2: $T_{s_{\text{归一化}}}=100 \text{ ms}$, $T_{s_{\text{显示}}}=60 \text{ ms}$ 。

如果采用常规模式,则处理时间需要

$$T_{s_{\text{归一化}}}+2 \times T_{s_{\text{显示}}}=220 \text{ ms}$$

远远大于 128 ms,不能持续运行;多线程模式,则流程 1 需要时间: $T_{s_{\text{归一化}}}+T_{s_{\text{显示}}}=160 \text{ ms}$; 流程 2 需要时间: $T_{s_{\text{显示}}}=60 \text{ ms}$ 。流程 1 不能在 128 ms 内完成处理,整个过程不能持续。

测试结果见表 1,表明流水线模式能够持续稳定地运行。

表 1 流水线性能测试结果

序号	步骤处理时间/ms		数据采集 间隔/ms	延迟/ms	
	归一化	显示		流程 1	流程 2
1	60	100	128	290	229
2	100	60	128	289	189

表 1 中的数据采样间隔指的是数据采集步骤连续 2 次开始采集数据的时间间隔。表中的数据均为平均值,其中,测试 1 运行了 513 次;测试 2 运行了 350 次。可以看出,数据采集间隔稳定在 128 ms,与 DAQ 采集 1 024 点数据块的时间相同,说明整个处理流程能够及时地处理数据,测试系统能

(上接第 98 页)

或 $e' \cap e = \emptyset$, 则 $H_{i+1} = (N_i \cup e', E_i \cup e')$ 。

规则(1)说明,在原超图 H_i 中若有一条边包含连接节点集 $e' \cap N_i$,且该边与原超图中其他边集的公共节点也包含在连接节点集中,则可将超边 e' 加入到 H_i 中。规则(2)说明,原超图 H_i 中的边如果与连接节点集有公共节点,则该边必须包含连接节点集 $e' \cap N_i$ 。如果每个设计步都能按规则(1)或规则(2)进行,则不会生成 XML_γ 环。因为规则(1)和规则(2)保证了结果超图中的每条边与其余边集的公共节点不会出现在不同的边中,因此,不会形成 XML_γ 环。

5 结束语

本文在一定范围内定义、分析了 XML 文档中的 XML_γ 环问题,但其中还存在一些问题,比如对无 XML_γ 环文档的特征分析得不够透彻、准确,而且此类文档可能存在其他一些特征,有待进一步挖掘。同时,对 XML 文档无环程度

够持续地运行下去。对于延迟,可以看到,流程 1 的延迟稳定在 290 ms,基本上等于 3 个步骤的处理时间之和:

$$T_{s_{\text{数据采集}}}+T_{s_{\text{归一化}}}+T_{s_{\text{显示}}}=128+60+100=288 \text{ ms}$$

同样,对于流程 2,延迟也基本上等于 2 个步骤的处理时间之和,没有增大。

在连续采样的在线测试系统中应用流水线技术,其步骤时间必须小于 DAQ 设备的采样时间,因此流水线实际上处于轻载工作状态,能够在不增大延迟的情况下,大幅度地提高吞吐率,使得通常模式不能运行的系统得以运行。同时,只要新增步骤的处理时间小于 DAQ 设备的采样时间,整个过程就能够持续运行,因此,可以添加更多数量的处理模块,实现功能丰富的在线处理,而不必担心性能退化。

5 结束语

本文研究了利用流水线技术提升虚拟仪器软件系统的运行性能,给出吞吐率和延迟这 2 个性能指标的计算方式,并进行了验证。应用流水线技术的连续采样虚拟仪器实际上处于轻载工作状态,能在不增大延迟的情况下大幅提高吞吐率,使得在通常模式下不能运行的系统得以运行。应用流水线技术能够处理更高采样率的数据,更好地满足性能要求,可添加更多功能模块,实现丰富的功能。

参考文献

- [1] Intel Corporation. The Microarchitecture of the Pentium 4 Processor[EB/OL]. (2001-01-01). http://www.intel.com/technology/itj/q12001/pdf/art_2.pdf.
- [2] 姚念民, 郑名扬, 鞠九滨. 基于流水线的高性能 Web 服务器[J]. 软件学报, 2003, 14(6): 1127-1133.
- [3] MIT OCW. Pipelining[EB/OL]. (2002-01-01). <http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-004-Computation-StructuresFall2002/29F152E6-EC37-4DC4-8550-B663E164C5A0/0/L09Pipelining.pdf>.
- [4] 赵江滨, 史铁林, 来五星, 等. IsVIP: 一种即时可重构虚拟仪器平台[J]. 仪器仪表学报, 2007, 28(4): 634-639.
- [5] 赵江滨. 传感器网络应用开发基础技术研究[D]. 武汉: 华中科技大学, 2007.

编辑 顾姣健

的分类也需要进一步分析(类似于关系数据库中的 α 环、 β 环、 γ 环)。为了消除文档的冗余,需要对文档进行分解,这就要使用 IDREFS, IDREF 类型的节点来相互引用,如果分解不当,就会出现 XML_γ 环,因此,需要分析如何分解这类文档,从而避免产生环。

参考文献

- [1] Beeri C, Frain R, Maier D, et al. Properties of Acyclic Database Schemas[C]//Proc. of ACM Symp. on Theory of Computing. [S. l.]: ACM Press, 1981.
- [2] Goldman R, Windom J. Data Guides: Enabling Query Formulation and Optimization in Semi-structured Database[C]//Proceedings of the 23rd International Conference on Very Large Data Bases. Athens, Greece: [s. n.], 1997: 436-445.
- [3] Arenas M, Libin L. A Normal Form for XML Documents[C]//Proc. of PODS'02. [S. l.]: ACM Press, 2002: 85-96.

编辑 张帆

