

基于网络磁盘结构的 Local Migration Manager 设计

赵延红¹, 掛下哲郎²

(1. 西安交通大学, 陕西 西安 710061;

2. 佐贺大学 理工学部智能情报系统学科, 日本 佐贺 840-8502)

摘要:为构成高性能的数据库, 磁盘阵列的活用是很有必要的。其中尤为重要是实现磁盘阵列之间负载的均衡。为此, 设计了网络磁盘结构系统, 该系统在磁盘阵列高负载的情况下也能够进行动态的负载均衡, 并且具有优良的耐故障性。对于网络磁盘结构构成要素中的 Local Migration Manager (LMM) 进行了设计。列举出了 LMM 连接的 Main Bus 上传送的信息, 定义了信息形式, 信息的传送与 Bus 的宽度相符呈分割进行; 在 LMM 的设计方法中定义了 LMM 的内部数据结构; 在此基础上, 制作了 LMM 的算法。

关键词:网络磁盘结构; local migration manager; 算法

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2011)01-0169-05

Design of Local Migration Manager in Net Disk Architecture

ZHAO Yan-hong¹, KAKESHITA Tetsuro²

(1. Xi'an Jiaotong University, Xi'an 710061, China;

2. Dept. of Information Science, Faculty of Science and Engineering, Saga University, Saga 840-8502, Japan)

Abstract: For constructing high-performance database servers, it is essential to make the best use of parallel disks. It is important to execute load balancing among disks for parallel disks. Hence, proposed the net disk architecture for this purpose. This system can perform efficiently for dynamic load balancing even though the disk arrays are under heavy load conditions. The architecture also achieves high fault tolerance. In this paper, designed the Local Migration Manager (LMM) in the net disk architecture. At first defined message structure on the main bus to which the LMM is connected. Messages are splitted to transfer through the fixed size bus. Next defined LMM specification together with the internal data structure. Based on these, the algorithms of LMM are developed.

Key words: net disk architecture; local migration manager; algorithms

0 引言

CPU 的处理速度以年 60% 提高, 内存的存取速度以年 100% 大幅增加, 而数据存储装置磁盘的存取速度增幅仅为年 7% 以下。CPU、内存和磁盘之间的速度差越来越大, 整个 I/O 吞吐量不能和系统匹配, 磁盘性能已经成为影响计算机性能的最大瓶颈所在。解决这个问题最受关注的是 RAID (Redundant Array of Independent Disks)^[1,2]。RAID 一方面通过冗余数据或校验信息增强容错能力; 另一方面通过硬件部件冗余来提高系统可靠性。但是, RAID 不能保证磁盘间的动态负载均衡。因此, 在此研究的基础上我们首先提

出了动态数据再配置^[3]。通过动态数据再配置, 负载最大的磁盘存储的数据动态的向负载最小的磁盘移动。进而, 我们设计了网络磁盘结构^[4], 该结构是利用动态数据的再配置实现磁盘阵列间的负载均衡, 并且基于硬件的多重化, 实现系统的高信赖性。网络磁盘结构是复数的磁盘阵列, 进行地址变换的组件和控制动态负载均衡的组件(见表 1), 基于网状网络连接起来。网络磁盘结构能够进行磁盘阵列内和磁盘阵列间的动态再配置, 实现系统的负载均衡。

文中面向网络磁盘结构的实用化, 进行了网络磁盘结构的构成要素中的 LMM (Local Migration Manager) 的设计。在此之前我们曾进行了 Switch 的逻辑电路的设计^[5], 而 LMM 进行的处理比 Switch 复杂得多, 用逻辑电路构成将无法实现。所以, LMM 计划使用微程序设计来实现^[6-9], 设计 LMM 的数据构造和算法。

收稿日期: 2010-05-05; 修回日期: 2010-08-11

基金项目: 日本文部科学省科学研究基金

作者简介: 赵延红 (1965-), 女, 陕西西安人, 留日计算机硕士, 工程师, 研究方向为磁盘阵列技术和网站建设; 掛下哲郎, 工学博士, 教授, 研究方向为数据库和软件工学。

表 1 网络磁盘结构构成组件

略称	英文全称
GAM	Global Address Manager
GDD	Global Data Dictionary
GMM	Global Migration Manager
S	Switch
LAM	Local Address Manager
LDD	Local Data Dictionary
LMM	Local Migration Manager
D	Disk
LDA	Local Disk Array

1 网络磁盘结构的构成

图 1 是网络磁盘结构的构成图。外部的访问要求由任意的 GAM 接受,数据由磁盘 Dij 分散保存。

GAM:用 GDD 把访问要求的逻辑地址转换成一对 LDA 物理地址和 LDA 内部逻辑地址,然后,把要求信息发送给 LDA,同时,从 GMM 接受再配置开始命令,实行 LDA 间的动态再配置。

GMM:监视各个 LDA 的状态及负载,并把信息发送给 GAM,在 LDA 间发送再配置开始/停止命令。

LDA:由 LAM, LMM, D 构成,通过 Main Bus 连接。

LAM:把被传送要求信息的 LDA 内部逻辑地址

转换成 LDA 内部物理地址,并把访问要求发送给相应的磁盘。这个组件内的地址变换,并不变更 LDA 外部数据的保存地址。同时,从 LMM 接受再配置开始命令,实行 LDA 内部的动态再配置。

LMM:监视 Main Bus 上的传送信息,自动计算各个磁盘的负载,并把这个信息传送给 LAM,在 LDA 内部发送再配置开始/停止命令。

网状网络:在 Switching Bus 和 Request Bus 之间由 Switch 连接的控制线路,实行各个组件间的通信。

2 网络磁盘结构的动作

网络磁盘结构的负载均衡分为 LDA 内部本地动态再配置和 LDA 间全局范围的动态再配置 2 个步骤进行。LMM 控制本地动态再配置。动态再配置和通常的数据访问处理同时实行。本地动态再配置模式时的 Read (Write) 处理如图 2 所示。

2.1 再配置模式时的 Read 处理

如果 GAM_i 发送给 LAM_j 的 Read 要求是给 D_{max} 的,那么 LAM_j 把 Read 要求接受以后,把 ReadMig 信息送给 D_{max},D_{max} 把 readMigAck 信息在 Main Bus 上播放后,这个信息由 LAM_j 和 D_{min} 接受,LAM_j 把要求的信息送回到 GAM_i。D_{min} 把数据保存以后,通过 Main Bus 把信息送给 D_{max},D_{max} 响应以后把旧的数据消去。

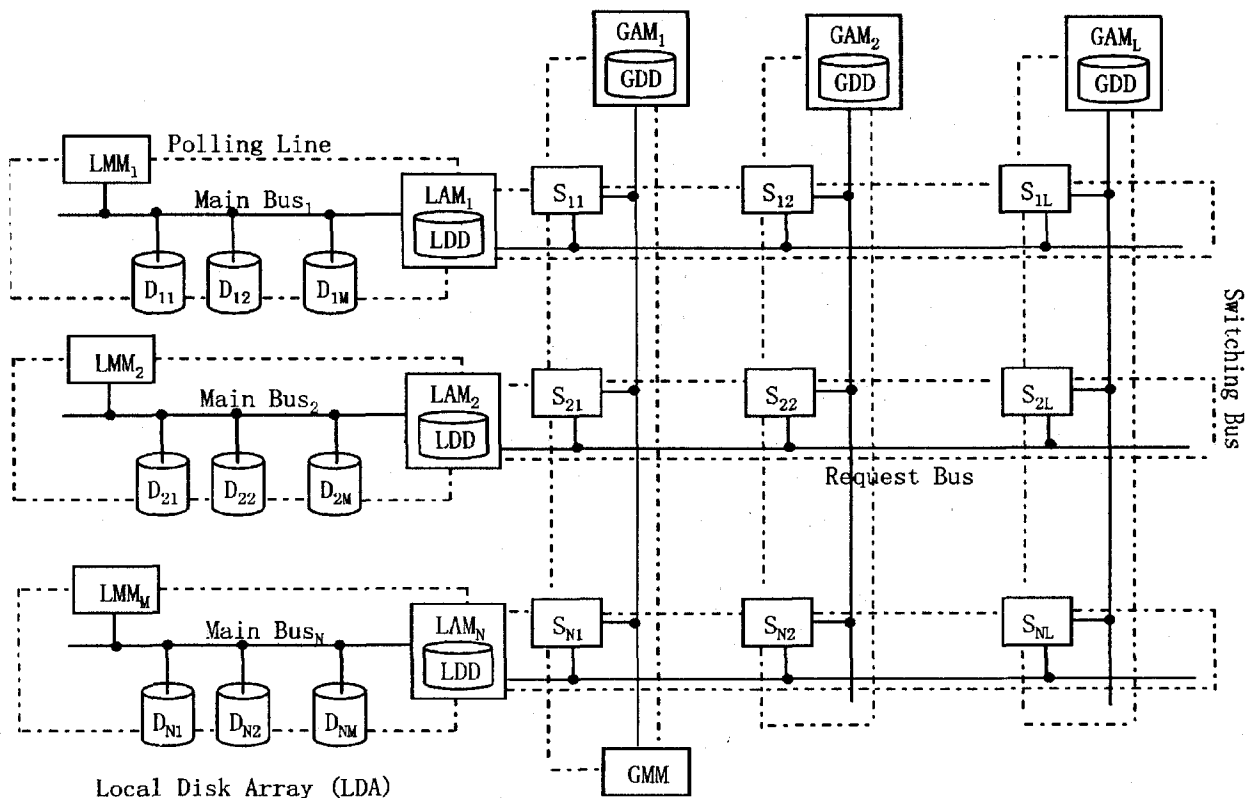
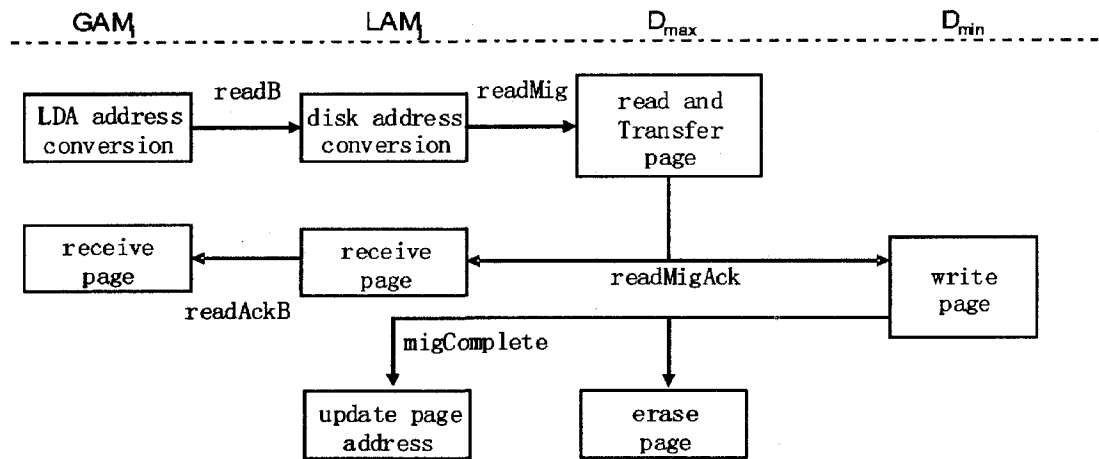
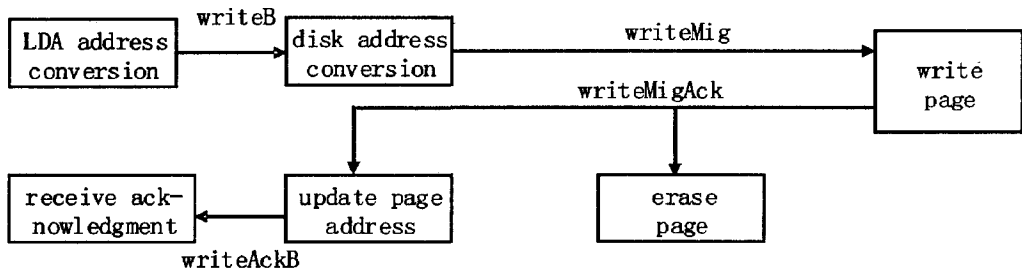


图 1 网络磁盘结构



(a) Read Operation with Local Migration



(b) Write Operation with Local Migration

图 2 本地动态再配置时的 Read(Write) 处理

2.2 再配置模式时的 Write 处理

如果 GAM_i 发送给 LAM_j 的 Write 要求是给 D_{max} 的,那么 LAM_j 把 Write 要求接受以后,把 WriteMig 信息送给 D_{min},D_{min} 把接收到的数据保存以后,把 Acknowledgment 信息送回 Main Bus,信息由 D_{max} 和 LAM_j 接受,D_{max} 把旧的数据消去,LAM_j 把 Acknowledgment 送回 GAM_i。

信息的先头字段和信息的终端字段中。信息开始/终了标志是在各个分割信息的先头加上 1bit 的标志位,1 表示信息先头/终端标志,0 表示数据部分。同时,信息先头/信息终端字段的区别是由信息内容部分判断的。信息传送时访问要求、地址和数据都是必要的因素,信息先头字段的信息内容部分如表 3 所示。

表 2 Main Bus 上传送信息的种类

信息名称	通常/再配置	信息类型	读出/写入开始/停止
通常时读出要求(Read)	0	00	0
通常时写入要求(Write)	0	00	1
再配置开始命令(MigStart)	1	10	0
再配置停止命令(MigStop)	1	10	1
再配置时读出要求(ReadMig)	1	00	0
再配置时写入要求(WriteMig)	1	00	1
再配置时读出确认要求(ReadMigAck)	1	01	0
再配置时写入确认要求(WriteMigAck)	1	01	1
再配置时数据消去要求(Erase)	1	11	0

3 Main Bus 上的信息

LMM 的设计方法中首先要确定的就是信息的数据结构。根据网络磁盘结构的动作,决定 Main Bus 上传送信息的数据构造。

3.1 Main Bus 上传送的信息的种类

网络磁盘结构中在 Main Bus 上传送的信息是分类传送的。为了判别 Main Bus 上传送的信息,给各个信息加上识别代号,信息的种类、通常/再配置,读出/写入的分类如表 2 所示。

3.2 与总线宽度相符信息的分割传送

Main Bus 上传送的信息同时发送时,信号线的数量和硬件的复杂性都会增加。所以,决定把信息以 32bit 总线的宽度相符分割后传送,信息内容包含在信

表 3 信息先头字段

标志	数据位数	名称	说明
1	1—4	MessageType	信息类型
	5—8	DiskNumber	磁盘号(通常时);负载最大磁盘号(再配置时)
	9—12	DiskMinAddress	负载最小磁盘号
	13—16	LamAddress	LAM 地址
	17—20	GamAddress	GAM 地址
2	21—53	DiskInternalAddress	数据的磁盘内部物理地址
3	54—86	Address	数据的逻辑地址

4 LMM 的设计方法

在网络磁盘结构和动作以及 Main Bus 上传送信息的数据构造叙述的基础上,确定基于网络磁盘结构的设计方法,设计方法中明确了输入/输出线以及内部数据。LMM 的输入/输出如图 3 表示。

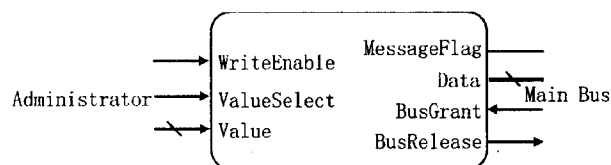


图 3 LMM 的输入/输出

Administrator 侧的信号线中,是系统管理者设定的 LMM 的系统参数,Main Bus 上连接的信号线则是 LMM 动作时使用的。Main Bus 连接的信号线中,MessageFlag 是为了识别信息先头字段/终端字段和数据部分的,Main Bus 上 LMM 以外连接的 LAM、磁盘等的组件是它们之间总线调停所必要的。BusGrant、BusRelease 是根据轮询实现总线调停使用的,各组件通过 BusGrant 取得总线利用权时,信息就可以播放,播放结束以后,通过 BusRelease 把总线利用权交给其他的组件。LMM 的内部数据详见表 4。

表 4 LMM 的内部数据

名称	数据形式	说明	初期值
TempQueue	(信息先头字段)*	临时保存从 Main Bus 接受信息的先头字段	Φ
MainBusQueue	(信息先头字段)*	Main Bus 播放的 MigStart/MigStop 信息保存	Φ
LoadTable	(磁盘的负载数)*	保存各个磁盘的负载数	Φ
Beta	β 的值	动态再配置参数	Φ
Gamma	γ 的值	动态再配置参数	Φ
DiskMax	磁盘号	保存负载最大磁盘号	Φ
DiskMin	磁盘号	保存负载最小磁盘号	Φ
MaxLoad	磁盘的负载数	负载数的最大值	11...1

5 LMM 的设计

本节进行 LMM 的设计。最初表示了 LMM 的全体构造。因为 LMM 的机能复杂,预定用微程序设计实现。包括 LMM 的系统参数(β、γ 以外)的初期设定,Main Bus 上信息被播放时的接收处理和通过 BusGrant 获得总线利用权时处理算法的构建。

5.1 LMM 的全体构造

LMM 的全体构造如图 4 所示。

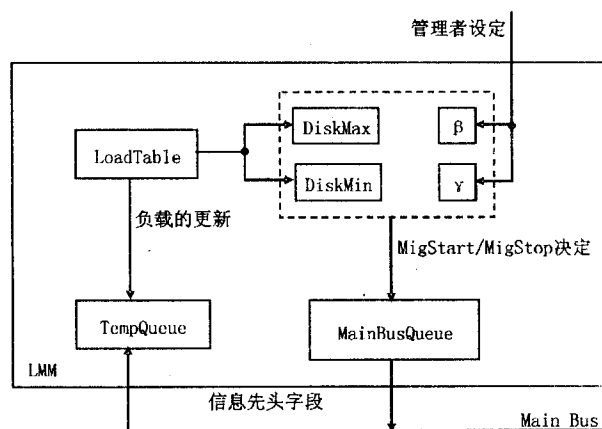


图 4 LMM 的全体构造

Main Bus 上传送的信息中被访问磁盘地址和信息先头字段在 LMM 的 TempQueue 中一时保存。根据保存的信息更新 LoadTable,记录各个磁盘的负载。根据各个磁盘的负载,寻求最大负载、最小负载的磁盘号码 DiskMax, DiskMin。根据 DiskMax, DiskMin 的负载和 β、γ 的比较,判断是否进行动态的再配置。进行再配置时 MigStart/MigStop 命令在 MainBusQueue 中追加。BusGrant 输入后,MainBusQueue 中保存的全部信息向 Main Bus 播放。

5.2 系统参数的初期设定

清空 TempQueue, MainBusQueue, LoadTable。

—Queue 清零—

TempQueue, MainBusQueue 中存放的一个信息的先头标志是在开始地址 ~ 开始地址+12 分割放置的,也就是说,13 个存储器地址存放一个信息的先头标志。

相应 Queue 的先头地址开始到末尾地址,重复进行以下的处理:

(1) 清空地址的内容。

(2) 清空的地址加 1 为下一个地址。

5.3 信息的收信处理

影响磁盘负载的因素有四种类型:Read, Write, ReadMig, WriteMig。同时,为了使 LoadTable 登记的磁盘负载不要溢出,任意的磁盘负载在到达 MaxLoad 时,每次都把全部的磁盘负载设定为 1/2。考虑以上的因素,算法表示如下:

(1) 来自 Main Bus 上信息的先头字段写入 TempQueue。

(2) LoadTable 的最大负载和 MaxLoad 相等时,在 LoadTable 内把各个磁盘的负载数变为 $1/2$ 。

(3) MessageType 是 Read, Write, ReadMig 的任意一个时,在 LoadTable 内 DiskNumber 对应的负载数加 1。

(4) MessageType 是 ReadMig, WriteMig 的任意一个时,在 LoadTable 内 DiskMinAddress 对应的负载数加 1。

(5) 在 TempQueue 中删除该信息。

(6) 寻求 DiskMax, DiskMin。

(7) DiskMax 和 DiskMin 的负载数的差值大于 β 时,在 MainBusQueue 中写入 MigStart, DiskMax, DiskMin。

(8) DiskMax 和 DiskMin 的负载数的差值小于 γ 时,在 MainBusQueue 中写入 MigStop, DiskMax, DiskMin。

●—在 TempQueue 中写入信息的先头标志—

信息从存储器的先头地址开始到末尾地址写入,如果是末尾地址,就从先头地址开始写入。

MessageType 是 Read, Write, ReadMig, WriteMig 的任意的一个时,进行以下的处理:

(1) 写入的地址如果不是末尾地址,进行以下的处理。

(1-1) 写入地址中写入信息的先头字段。

(1-2) 在 TempQueue 中保存的信息总数增加 1。

(2) 写入的地址如果是末尾地址时,进行以下的处理:

(2-1) 在开始地址中写入信息的先头字段。

(2-2) TempQueue 中保存的信息总数加 1。

(3) 写入地址增加 1,做为下一个地址。

●—寻求 DiskMax, DiskMin—

(1) DiskMax, DiskMin 清零。

(2) LoadTable 的开始地址到末尾地址重复磁盘总数的循环次数做以下的处理:

(2-1) 指定地址的负载数比 DiskMax 地址的负载数大时,指定地址为 DiskMax。

(2-2) 指定地址的负载数比 DiskMin 地址的负载数小时,指定地址为 DiskMin。

(2-3) 指定地址增加 1 为下一个地址。

●—给 MainBusQueue 中写入信息—

(1) MainBusQueue 如果为空时,在 MainBusQueue 中写入信息的先头字段的 MessageSave, DiskMax, DiskMin。

(2) 如果不为空,MainBusQueue 中保存的最后一

个信息和信息的 MessageSave, DiskMax, DiskMin 比较,如果有任意的一个不同,进行以下的处理:

(2-1) 在 MainBusQueue 中写入信息的先头字段 MessageSave, DiskMax, DiskMin。

(2-2) MainBusQueue 中的存入的最后信息的位置保存。

(2-3) MainBusQueue 中保存的信息总数加 1。

5.4 获得总线使用权时的处理

BusGrant 到达时,MainBusQueue 中登陆的全部信息向 Main Bus 播放。

(1) MainBusQueue 中的全部信息向 Main Bus 播放。

(2) BusRelease 成为 1。

6 结束语

网络磁盘结构能够实现磁盘阵列 LDA 内和 LDA 间的负载均衡,并且系统具有良好的耐故障性。本文从网络磁盘结构的实用化考虑,进行了 LMM 的设计。在定义了 Main Bus 上传送信息的数据构造和 LMM 的输入/输出线和内部数据的基础上构建了 LMM 的算法。LMM 的功能是通过微程序设计实现的。微程序设计克服了组合逻辑控制单元线路庞杂的缺点,能够实现更加复杂的功能,同硬布线比较具有规整性,灵活性,可维护性等一系列优点^[10]。

今后,我们将考虑使用比微程序内容更容易理解的高级语言编写程序,在此基础上使用汇编语言编写程序^[11-12]。同时,进行网络磁盘结构的其他组件的设计。

参考文献:

- [1] Patterson D A, Gibson G, Katz R H. A case for redundant arrays of inexpensive disks (RAID) [C]//Proc. of International Conference on Management of Data (SIGMOD). Chicago: [s. n.], 1988:109-116.
- [2] アンドリユー・S・タネンバウム. 構造化コンピュータ構成[M]. 第4版,株式会社ピアソンエデュケーション, 2000.
- [3] Kakeshita T, Kubo S. A transaction processing architecture for effective load balancing utilizing high speed bus [C]//Proc. Int. Symp. on Cooperative Database Systems for Advanced Applications (CODAS). Kyoto: [s. n.], 1996:376-379.
- [4] Kakeshita T, Zhang S. The net disk architecture for dynamic load balancing among disk arrays [C]//Processings of the Seven International Conference on Parallel and Distributed System (ICPADS). Iwate: [s. n.], 2000:315-322.

(下转第 177 页)

$$\hat{p}_{xx}(f) = \frac{|X_L(f_k)|^2}{f_s L}$$

$$f_k = \frac{k f_s}{N} \quad k = 0, 1, \dots, N-1, f_s \text{ 是采样频率, } L \text{ 为}$$

信号长度,其中 $X_L(f_k) = \sum_{n=0}^{N-1} x_L[n] e^{-2\pi j k n / N}$, 计算步骤:

$$k = 0, X_L(0) = \sum_{n=0}^{N-1} x_L[n] e^{-2\pi j 0 n / N}$$

$$k = N-1, X_L(N-1) = \sum_{n=0}^{N-1} x_L[n] e^{-2\pi j (N-1) n / N}$$

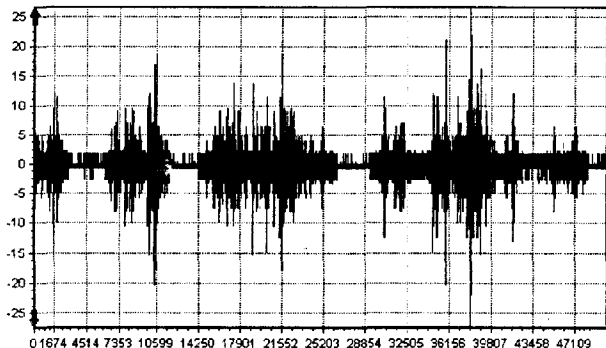


图5 加速度数据

根据上面的算法,得到5号节点采样数据对应的功率谱密度如图6所示。加速度数据的功率谱密度反映了数据中各个频率分量的作用大小。图中幅度峰值对应的频率值为 $f_0 = 7.907\text{Hz}$, 根据功率谱密度的意义,桥梁的固有频率即为 $f_0 = 7.907\text{Hz}$ 。根据桥梁结构的建设资料,其设计的固有频率理论值为 7.5Hz ,两者差值在允许范围内。表明系统监测到的数据是有效的,桥梁处于健康状态。

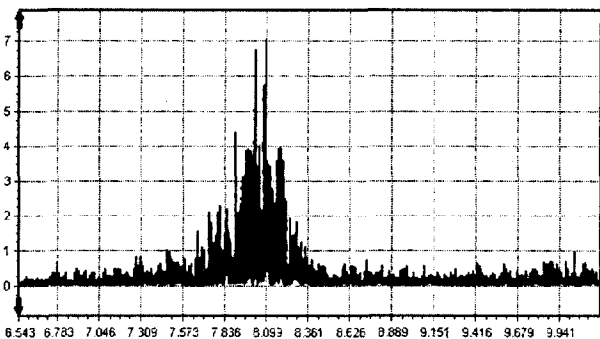


图6 测试结果分析

4 结束语

经实践证明,基于无线传感器网络技术的桥梁健康监测系统体积小,安装方便,实时性好,检测结果可靠,适用于对桥梁健康状态的实时监测。

参考文献:

- [1] Akyildiz I F, Su W, Sankarasubramaniam Y, et al. Wireless sensor networks: a survey [J]. IEEE Computer Networks, 2002, 38: 393-422.
- [2] Kim N S. Wireless sensor networks for structural health monitoring [D]. University of California at Berkeley, 2005.
- [3] Kim S, Pakzad S. Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks [C] // In: 6th international conference on Information processing in sensor networks. Cambridge, Massachusetts, USA: [s. n.], 2007: 254-263.
- [4] Lynch J P, Law K H, Anne, et al. Validation of a Wireless Modular Monitoring System for Structures [C] // SPIE's 9th Annual International Symposium on Smart Structures and Materials. San Diego, CA, USA: [s. n.], 2002: 17-21.
- [5] 吴春倩, 郑明春, 秦继林. 无线传感器网络协议研究 [J]. 计算机技术与发展, 2006, 16(8): 27-29.
- [6] 姜连祥, 汪小燕. 无线传感器网络硬件设计综述 [J]. 单片机与嵌入式系统应用, 2006(11): 13-16.
- [7] Hill J, Szewczyk R, Culler A. System architecture directions for networked sensors [M] // In: Architectural Support for Programming Languages and Operating Systems. [s. l.]: [s. n.], 2000: 93-104.
- [8] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks [J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660-670.
- [9] 汤强, 汪秉文. 邻居多跳分布式分簇路由协议 [J]. 华中科技大学学报: 自然科学版, 2010, 38(2): 26-29.
- [10] 胡晓娅, 朱德森, 汪秉文. 网络控制系统的时延补偿策略研究 [J]. 系统工程与电子技术, 2005(11): 1932-1934.
- [11] 孙利民, 李建中, 陈渝, 朱红松. 无线传感器网络 [M]. 北京: 清华大学出版社, 2005.
- [12] 姚天任, 江太辉. 数字信号处理 [M]. 武汉: 华中科技大学出版社, 2005.

(上接第173页)

- [5] 趙延紅, 掛下哲郎. NetDiskアーキテクチャにおけるスイッチの論理回路設計 [C] // 電気関係学会九州支部第56回連合大会講演論文集, 2003.
- [6] 石田晴久. マイコンコンピュータのプログラミング [M]. 共立出版, 1978.
- [7] 楠田喜宏. マイコン再入門 [M]. 日刊工業新聞社, 1981.

- [8] 相原隆文. 手作りマイコン [M]. 技術評論社, 1985.
- [9] 伊藤誠. 基本ハードウェア技術 [M]. CQ出版社, 1978.
- [10] 车海康, 杨银堂, 周拥华, 等. 数值协处理器中微程序设计 [J]. 微电子学与计算机, 2003, 20(6): 57-61.
- [11] Hayes J P. Computer architecture and organization [M]. McGRAW-HILL International Editions, 1988.
- [12] 王文东, 李竹林, 尚建人. 汇编语言与C语言的混合程序设计 [J]. 计算机技术与发展, 2006, 16(8): 18-20.