第37卷 第17期
Vol.37 No.17

计 算 机 工 程
Computer Engineering

2011年9月
September 2011

·开发研究与设计技术· 文章编号：1000—3428(2011)17—0268—03 文献标识码：A 中图分类号：N945.13

基于MPI的电网仿真实时并行计算平台

陈 勇，李亚楼，田 芳，张 星

(中国电力科学研究院，北京 100192)

摘 要：在电力系统动态电磁暂态仿真的并行计算中，存在超实时和硬实时的问题。为此，提出一种基于MPI的实时并行计算平台。引入硬实时操作系统RTLinux，采用实时内核和PSDD编程模式对仿真并行计算程序、MPICH并行环境和GM软件等进行重构，以获得硬实时特性。测试结果证明，该平台的平均时间性能提高约10%，时间的最大抖动幅度降低50%~80%，并减少了时间的抖动频率。

关键词：电网仿真；并行计算；硬实时操作系统；实时重构

Real-time Parallel Computing Platform of Power Grid Simulation Based on MPI

CHEN Yong, LI Ya-lou, TIAN Fang, ZHANG Xing

(China Electric Power Research Institute, Beijing 100192, China)

【Abstract】Aiming at electricity system dynamic electromagnetism transient state parallel computing pass real-time and hard real-time problem, this paper proposes a real-time parallel computing platform which is based on MPI. By introducing hard real-time operating system, RTLinux, and using real-time core, PSDD programming model to reconstruct the simulation parallel computing program, MPICH parallel circumstance and GM software, it can have hard real-time character. Test results show that, the average time performance enhances about 10%, and maximum jitter range reduces 50%~80%. It can also reduce time jitter frequency.

【Key words】power grid simulation; parallel computing; hard real-time operating system; real-time reconfiguration

DOI: 10.3969/j.issn.1000-3428.2011.17.090

1 概述

目前，我国电网互联的规模不断扩大，全国大区逐渐联网，电力交易面向市场化引起运行方式多变，这些复杂电网要求建立与实际电网类似的实时仿真试验环境，以保证实时仿真过程与系统运行过程同步。随着廉价PC机群和并行计算技术在仿真应用中的广泛应用，大规模电力系统甚至能实现仿真计算时间比系统实际过程短的超实时动态仿真。大规模电网的并行仿真技术[1-2]将大电力网络分割为若干子网，分配给机群不同的节点机作为子网机，并选一台节点机作为主控机，以实现并行计算。在电力系统全数字实时仿真装置[3]中，对10 000节点级大规模电力系统电磁暂态分网并行仿真，采用MPI并行编程模型和Myrinet网络可以实现电磁暂态超实时仿真，将步长为50 μs的10 s动态过程分4个子网并行计算可将计算总时间缩短为7.71 s。

在仿真应用中，廉价的硬件PC机群系统、Linux操作系统、MPICH并行程序设计运行环境和Myrinet高速网络共同构成仿真并行计算平台。由于Linux操作系统本身是分时系统，其调度算法、不可中断的系统调用、中断屏蔽、设备驱动以及虚拟内存的使用，导致系统时间不可预测，因此引起仿真步长时间抖动，单个步长所带来的时间抖动影响后续所有的步长仿真，故很难实现严格步长时间(50 μs)内的动态硬实时仿真，难以保证电网实时仿真严格与系统运行时间同步甚至超前，在大电网事故发生时难时进行实时计算分析，从而为电网预防控制争取时间。基于此，本文引入硬实时RTLinux系统对Linux操作系统进行实时重构，从而为电网仿真计算应用提供硬实时的并行计算平台。

2 硬实时操作系统RTLinux

RTLinux是一种基于Linux的硬实时操作系统[4]，其在Linux内核与硬件之间增加虚拟层，从而构筑一个与Linux内核相区分的实时内核。RTLinux同时实现高效的、可抢先的实时调度核心，全面接管中断，可在其上面运行实时任务，如MPI仿真计算等。RTLinux独特的双内核技术把Linux作为实时内核上一个较低优先级的进程运行，也可以运行Linux的非实时进程，其结构如图1所示。

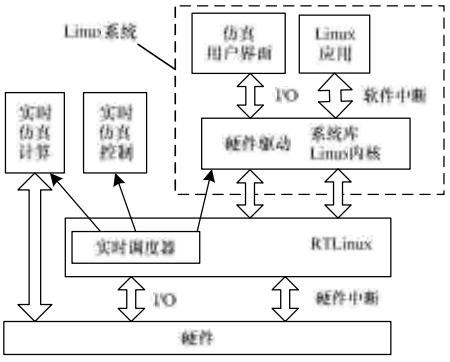

图1 RTLinux结构

基金项目：国家“十一五”科技支撑计划基金资助项目(2008BAA13B03)；中国电力科学研究院青年科研基金资助项目(SF84-09-001)

作者简介：陈 勇(1975－)，男，工程师、硕士，主研方向：分布式并行技术；李亚楼、田 芳，高级工程师、博士；张 星，工程师、硕士

收稿日期：2011-03-01 **E-mail：**ychen@epri.sgcc.com.cn

MPI仿真计算是建立电力系统元件的数学模型，将全网的数学模型根据算法分割成若干子网模型，并利用Myrinet网络的机群MPI消息传递机制，主控节点将子网模型分配到机群计算节点上进行计算，主控节点和计算节点要保证实时计算和通信。仿真实时控制是计算主控节点和外部物理装置接口之间以50 μs为周期进行实时通信，以实现对外围实时物理装置的控制。仿真用户界面可以对计算结果进行输出、浏览和监视等。

为满足上述计算和应用，RTLinux必须对MPI仿真计算中的仿真计算程序、MPICH、GM进行实时化改造，对仿真实时控制提供网络套接字控制的标准接口RTLinux LNet，并在MPI仿真计算实时应用和仿真用户界面非实时应用之间提供交互管道等。

MPI仿真计算程序的RTLinux重构编程方式有内核编程模式、PSDD编程模式。内核编程模式将程序中的实时处理部分编写成内核模块，并装载到RTLinux内核中，运行于RTLinux的内核态。内核编程模式允许实时任务对硬件的直接访问，使实时任务具有最小的延迟和最优先的处理器使用权。PSDD编程模式允许编程人员在Linux进程的地址空间创建实时线程，可在用户模式下运行实时应用。PSDD线程可在实时调度器控制下运行，但要共享Linux进程有限的受保护地址空间。RTLinux LNet遵循POSIX标准接口API标准，采用零拷贝技术，为仿真实时控制提供标准网络套接字控制的实时网络通信功能。MPI仿真计算实时应用和仿真用户界面非实时应用之间可通过RT_FIFO和共享内存等多种方式交换数据。因此，通过引入硬实时操作系统RTLinux，可为数字电网仿真并行计算平台提供完整的实时解决方案，弥补Linux系统在实时性方面的缺陷。

3 实时计算平台的实现

3.1 实时平台的构建方法

MPI仿真计算的实时重构实施部分包括仿真计算程序、MPICH、GM软件等。其中，仿真计算程序为具体用户提供应用，通过MPI函数接口完成数据和计算程序在机群节点之间的分发和计算。MPICH是MPI标准的一种最重要的实现，MPI是为统一不同MPP产商传递API消息而制订的工业标准。GM是适用于Myrinet网络的用户态通信协议[5]。GM软件结构主要由GM函数库(GM library)、驱动程序(GM driver)和Myrinet控制程序(GM MCP) 3个部分组成。针对上述内容，对仿真计算程序、MPICH和GM library采用PSDD编程方式进行实时改造，对GM driver和GM MCP采用内核编程模式进行改造。这3个部分的结构如图2所示。

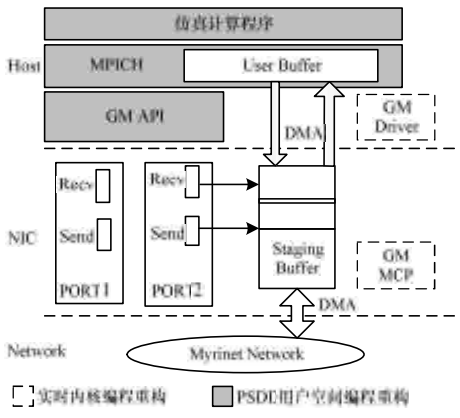

图2 实时并行计算平台结构

3.2 实时内核的编程方式

GM driver的功能包括加载和初始化MCP，将网络接口的内存区域和寄存器映射到用户层等。考虑用户空间驱动程序的局限性，GM driver采用实时内核编程模式，在其运行时被加载到RTLinux内核中。GM MCP经驱动程序初始化加载到Myrinet网卡的SRAM区域运行。GM driver软件必须将Linux系统调用函数替换成实时库函数，例如rtl_open、rtl_close、rtl_read、rtl_write、rtl_ioctl等函数。GM driver软件经过实时化改造后形成GM实时驱动，包括实时库函数和GM驱动程序2个部分。

在RTLinux系统中，当GM实时驱动被加载时，RTLinux中断向量表被写入设备驱动程序的中断服务子程序调用位置，同时修改内核资源符号表，该驱动模块所提供的服务和资源由RTLinux系统接收硬件中断，并由RTLinux系统调用中断服务例程。

对GM驱动程序代码进行微小改动，将其正确编译并链接到RTLinux实时库后，加载到实时内核中就可以使GM driver为GM library提供实时访问和响应。

3.3 PSDD编程方式

仿真计算程序、MPICH和GM library在用户空间实现实时访问，采用PSDD模式将实时函数部分置于实时线程，在实时调度器的控制下被优先调用。PSDD重构方案主要包括2个方面：

(1)实时线程中的函数调用路径上的函数及其子函数实现，避免有malloc()、free()等内存分配函数和fork()、exec()、popen()、system()等系统调用函数，也不能使用一些可能更改内存映射区的libc函数；

(2)在实时应用初始化时要使用mlockall(MCL_CURRENT|MCL_FUTURE)函数锁定进程内存空间，并为用户使用内存进行预分配，在后续的内存操作中尽量使用已分配内存，通过rtl_growstack (stacksize)扩大程序堆栈空间。通过这些改造，可消除原有函数调用所带的时间不确定性，确保函数调用返回的实时性。

经过上述方法改造后的MPICH库中支持实时调用的MPI函数如表1所示。

仿真计算程序的实时处理函数部分置于实时线程，仿真实时处理函数向下调用MPI实时函数，然后MPI实时函数调用GM library实时GM访问函数，通过GM library中的初始化、消息发送和接收、令牌管理等主要通信函数直接访问网络接口内存区，通过直接和GD MCP的交互完成消息的收发和控制操作。

表1 改造后的实时MPI函数

函数名	作用
MPI_COMM_RANK	获取本进程的进程号
MPI_RECV	接收其他进程发送过来的消息
MPI_ALLGATHERE	收集所有进程数据
MPI_BARRIER	进程同步函数
MPI_COMM_SIZE	获取总进程数
MPI_SEND	发送消息给特定的进程
MPI_ALLGATHERV	收集所有进程不同长度的数据块
MPI_BCAST	消息广播到组内所有进程
MPI_Comm_group	根据组创建新的通信域
MPI_SENDRECV	执行一个阻塞的发送和接收
MPI_SCATTER	散发相同长度数据块
MPI_Allreduce	全归约操作

4 测试结果与分析

MPI 实时机群使用 3 个 PC 服务器和 Myrinet 网络等硬件进行测试，具体配置如图 3 所示。

节点硬件环境	
主机名	配置
Node1 MPI Process0	2×Intel Xeon 3.2 GHz CPU(1 MB L2 Cache) 2 G ECC DRR RAM 1×36 G Ultra320 10 k SCSI Harddisk
Node2 MPI Process1	2×Intel Xeon 3.2 GHz CPU(1 MB L2 Cache) 1 G ECC DRR RAM 1×36 G Ultra320 10 k SCSI Harddisk
Node3 MPI Process2	2×Intel Xeon 3.2 GHz CPU(1 MB L2 Cache) 2 G ECC DRR RAM 1×36 G Ultra320 10 k SCSI Harddisk
网络环境	
Myrinet 交换机	M3F-SW16
Myrinet 网卡	M3F-PCIXF-2
操作系统	
Linux OS	RedHat Linux AS 3.0 Update 1
Linux 内核	Linux 2.6.9rtl SMP
RTLinuxPro	2.2.0-prelease
软件版本	
MPICH-GM	MPICH-GM-1.2.6 for Linux
GM	GM-2.1.21 for Linux

图 3 测试环境

测试方法采用测试圆周率的 MPI 计算，Node1 的时间性能测试周期为测试一次圆周率的操作周期。Node2 和 Node3 的时间性能测试周期为测试一次圆周率的非根进程操作。

时间周期长度从 3 个进程(process0, process1, process2)同步后开始计时，并在对应 MPI 操作完成后为一个计时周期。由于计时周期测试次数为 10 000 次，因此测试结果比较密集，图中深黑色的点表示其测试结果较为松散，具体结果如图 4~图 9 所示。图 4 的最大单次循环时间是 1 447.2 μs；图 5 的最大单次循环时间是 779.3 μs；图 6 的最大单次循环时间是 324.5 μs；图 7 的最大单次循环时间是 186.8 μs；图 8 的最大单次循环时间是 184 μs；图 9 的最大单次循环时间是 180.6 μs。

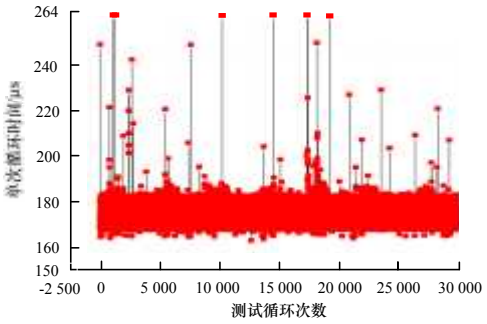

图 4 Linux MPICH-GM Node1 的时间性能

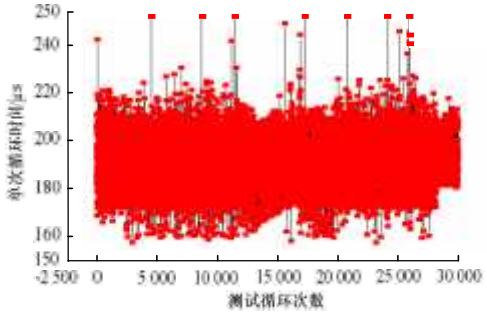

图 5 Linux MPICH-GM Node2 的时间性能

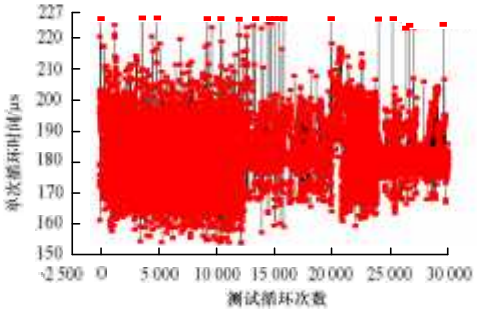

图 6 Linux MPICH-GM Node3 的时间性能

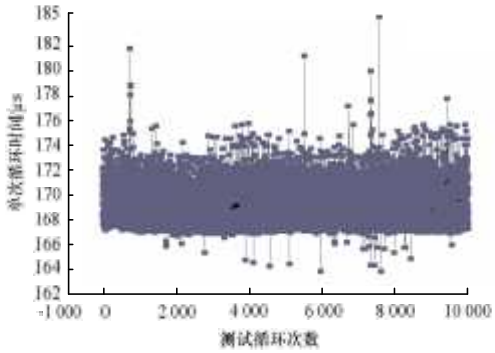

图 7 实时 MPICH-GM Node1 的时间性能

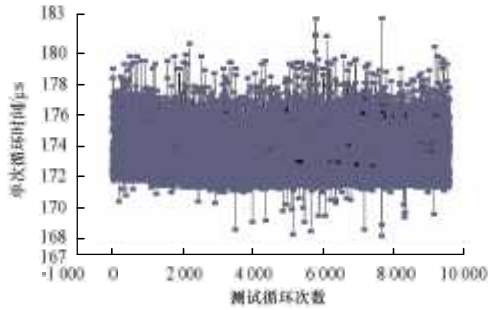

图 8 实时 MPICH-GM Node2 的时间性能

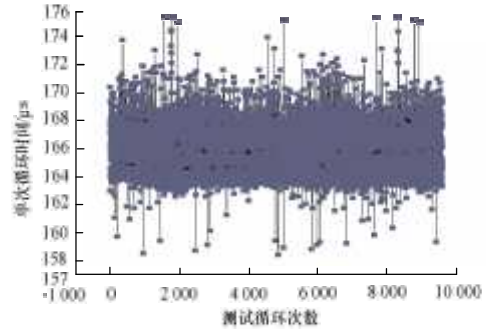

图 9 实时 MPICH-GM Node3 的时间性能

测试发现，实时 MPICH-GM 和 Linux MPICH-GM 相比平均时间性能提高约 10%，其循环时间的最大抖动幅度降低 50%~80%。实时化 MPI 通信在相同测试循环次数内，时间抖动的次数有较大减少，且实时化 MPI 通信中的时间抖动幅度具有可控性，从而满足 MPI 实时应用的硬实时需求。

5 结束语

本文提出一种基于 MPI 的电网仿真实时并行计算平台。该平台采用 MPI 编程模型、实时内核编程模式和 PSDD 编程模式对 MPICH 和 GM 软件进行重构。测试发现，通过使用实时 MPI 程序设计环境和仿真并行应用的 PSDD 编程重构规范，能快速为各种不同并行计算应用带来硬实时特性，使原

(下转第 273 页)