EPICS IOC 冗余技术研究

Research on EPICS IOC Redundant Technology

尹聪聪[1,2]　韩利峰[1]　李勇平[1]　陈永忠[1]　汪全全[1,2]

(中国科学院上海应用物理研究所[1],上海　201800;中国科学院大学[2],北京　100049)

摘　要：为了提高反应堆及其他高安全等级控制系统的安全可靠性,设计了一种基于 Heartbeat 软件和 Autosave 软件的 EPICS IOC 冗余测试平台。在此测试平台中,Heartbeat 用于心跳监测,实现 IOC 的冗余切换;Autosave 动态保存 IOC 记录的域到网络文件系统(NFS)服务器中,以实现数据同步。经测试,该系统运行良好,可快速实现 IOC 的冗余切换和数据信息同步。基于 Heartbeat 和 Autosave 的 EPICS IOC 冗余技术可在高可靠性等级控制系统得到进一步应用。

关键词：实验物理及工业控制系统(EPICS)　反应堆　高可靠性　冗余　数据同步

中图分类号：TP273+.5　　**文献标志码**：A

Abstract: With the purpose of improving the safety and reliability of the reactors and other control systems in high security level, the EPICS IOC redundant test platform based on Heartbeat software and Autosave software has been designed. In this test platform, Heartbeat software is used for heartbeat monitoring, and implementing redundant switchover; Autosave software is used for dynamically saving the domains of IOC records into network file system(NFS) server to implement data synchronization. The tests show that the system runs well, and quickly carries out redundant switchover and data information synchronization. Thus the EPICS IOC redundant technology based on Heartbeat and Autosave can be applied in control systems with high reliability level.

Keywords: Experimental physics and industry control system(EPICS)　Reactor　High reliability　Redundancy　Data synchronization

0　引言

钍基熔盐反应堆系统因面临高温、高辐射等较恶劣环境,若发生核事故,其危害是相当恐怖的[1]。因此,对反应堆各个控制子系统的稳定性、冗余能力等提出了更高的要求,而构建分布式冗余控制系统是提高控制系统安全性、可靠性的有效解决方案。实验物理及工业控制系统(experimental physics and industry control system,EPICS)下的分布式冗余主要包括客户端冗余、输入输出控制器(input output controller,IOC)冗余、网络冗余、过程硬件冗余等,其中 IOC 冗余是反应堆冗余控制系统技术的难点。

高可用性软件(Linux high availability,Linux-HA)的主要目的是实现系统的高可用性,达到 99.999% 的可用性。Heartbeat 是 Linux-HA 软件的基石,其包含所有 HA 软件所需要的基本功能,为增强 Linux 的可靠性、可用性和可服务性提供了一个集群解决方案[2-4]。Autosave 是基于 EPICS 开发的一个软件,其可将 EPICS 过程变量的值自动保存到文件系统中。因 Autosave 具有自动保存数据的功能,在主服务器出现故障、备用服务器接管 IOC 资源时可同时实现 IOC 数据的同步[5]。对此,本文以 Heartbeat 软件和 Autosave 自动保存软件为基础,设计了一种 EPICS IOC 冗余系统。

中国科学院战略性先导科技专项基金资助项目(编号:XDA02010300)。
修改稿收到日期:2012-11-07。

第一作者尹聪聪(1986-),女,现为中国科学院上海应用物理研究所核技术及应用专业在读博士研究生;主要从事反应堆信号处理及控制系统的研究。

1　系统结构

冗余技术就是增加多余的设备,以保证系统更加可靠、安全的工作[6]。本设计中的 IOC 冗余是指当主服务器出现故障、IOC 不工作时,备用服务器接替主服务器工作并重新启动 IOC;而当主服务器运行正常后,备用服务器停止 IOC 运行,主服务器对外提供服务。

系统以两台相同配置的联想服务器为硬件平台,Centos6.1 操作系统为基础,基于 Heartbeat3.0 和 Autosave4.7 软件进行 EPICS IOC 冗余技术研究。图 1 为测试系统基本架构。图 1 中:节点 1 和节点 2 分别为主、备服务器,对外提供各种服务;NFS 服务器为节点 1 和节点 2 提供共享磁盘服务;操作员接口(operator interface,OPI)为 EPICS 客户端,进行测试验证。主服务器节点 1 和备服务器节点 2 之间的心跳线同时采用串口线和网线,这样可避免由于单一心跳线出现连接故障时,主、备服务器同时认为对方已损坏,

出现“脑裂”现象,以致争用资源的情况发生。局域网为双环网网络,可增强网络传输的可靠性。

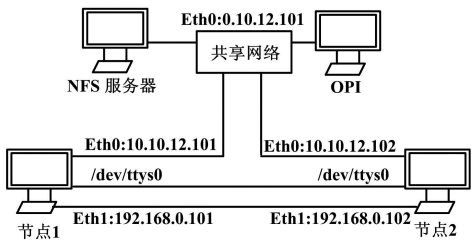


图 1　系统架构图

Fig. 1　System architecture

2　软件结构

图 2 为系统软件架构图。由图 2 可知,最底层为基于硬件的 Centos 操作系统,最上层为 EPICS IOC 及其他应用软件,中间层为连接操作系统与 EPICS IOC 的 Heartbeat 和 Autosave 软件。EPICS IOC 与 Heartbeat 之间的通信是通过 IOC 资源脚本文件实现的。当系统的主节点发生故障时,次节点将在几秒钟甚至更短的时间内自动接管主节点的资源。Heartbeat 为 EPICS IOC 冗余研究提供了一种软件基础。

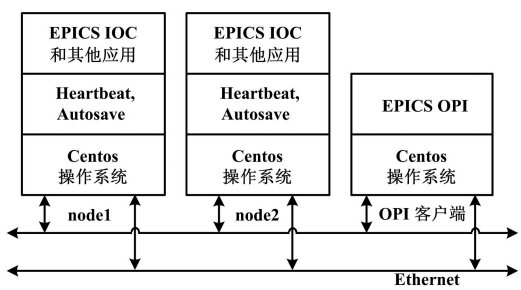


图 2　系统软件架构图

Fig. 2　System software architecture

2.1　EPICS 系统

EPICS 是由美国 Argonne 和 Los Alamos 等实验室为大型物理和工业装置联合开发的分布式控制软件,其基于标准的 Server/Client 模式,具有结构灵活、易于扩展、源代码完全开放等优点[7-9]。EPICS 目前应用在世界上一百多个大型科学研究工程项目的控制系统中,涉及同步辐射装置、对撞机等领域。其分布式控制系统可分为 OPI、IOC 及连接 OPI 和 IOC 的局域网。IOC 上运行实时数据库,通过记录支持程序、设备支持程序和驱动支持程序访问下层 I/O 设备,通过通道访问机制与上层 OPI 进行通信;OPI 为显示界面、报警、存档等软件提供访问数据库的接口。整个系统软件类型多,数量大。为试验测试需求,IOC 中数据库的部分记录如下所示。

```
record(bo,“testHost:a”){
    field(DTYP,“Soft Channel”)
    field(VAL,0)
    field(UDF,1)
    field(FLNK,“ testHost:calc1”)
    field(ZNAM,“off”)
    field(ONAM,“on”)
    field(SCAN,“1 second”)
}
record(calc,“testHost:calc1”){
    field(INPA,“testHost:a”)
    field(CALC,“! A”)
    field(FLNK,“testHost:c”)
}
record(bi,“testHost:c”){
    field(DTYP,“Soft Channel”)
    field(INP,“testHost:calc1.VAL”)
    field(ZNAM,“off”)
    field(ONAM,“on”)
}
```

2.2　Heartbeat 实现 IOC 冗余切换

开源软件 Heartbeat 是 Linux-HA 的核心部件,本设计采用 Heartbeat3.0,其主要组成部分为 Heartbeat、Cluster Glue、Resource Agent 和 Pacemaker[10]。Heartbeat 负责维护集群各节点的信息及它们之间的通信。Cluster Glue 主要包含本地资源管理模块(local resource manager,LRM)和问题节点从集群环境中脱离模块(shoot the other node in the head,STONITH)两部分,其中,LRM 负责本地资源的启动、停止和监控,STONITH 主要用于监控节点状态,当一个节点出现问题时,处于正常状态的节点通过 Fence 设备将其重启或关闭,以释放 IP、磁盘等资源。Resource Agent 控制服务启动停止,监控服务状态的脚本集合,其被 LRM 调用。Pacemaker 也就是 Heartbeat3.0 版本之前的集群资源管理模块(cluster resource manager, CRM),客户端通过 Pacemaker 配置、管理和监控整个集群。

Heartbeat 主要有 ha.cf、authkeys 和 haresources 三个配置文件。简要说明如下。

ha.cf 是 Heartbeat 的主要配置文件,其心跳间隔时间、死区时间、心跳传输采用串口线或网线等都在此设置。测试系统心跳间隔时间为 50 ms,即每 50 ms 发送一次心跳信号。系统采用串口线和网线两种传输方式进行心跳信号的传递,当其中一心跳线出现故障时,心跳信号还可通过另一种方式进行传输,这样可避免

由于单根心跳线故障而出现两台服务器争用共享磁盘的可怕现象。死区时间为 2 s,若备用节点 2 s 内未收到主节点的心跳信号,则立即接管主节点的服务资源,这样可确保主备系统在几秒钟内完成冗余切换。死区时间对系统冗余切换非常重要,故需要正确调整时间以适应系统和网络环境。若死区时间配置得太低,则在网络处理任务非常繁重或 CPU 利用率非常高的系统上,可能会引起心跳丢失或听不到,引发“脑裂”问题;但死区时间配置也不能太高,若太高,主、备服务器切换时间较长,影响系统工作。

authkeys 文件用于设定 Heartbeat 的认证方式,共有 crc、md5 和 sha1 三种认证方式。三种认证方式的安全性依次提高,但占用的系统资源也依次增加。反应堆系统有单独的内部网络,故采用安全级别比较低的 crc 认证方式即可。

haresources 资源文件用于指定系统的主节点、集群 IP 以及启动的服务等集群资源, EPICS IOC 启动服务脚本在此被调用。

2.3　Autosave 实现参数自动保存

冗余最关键的技术之一即是信息同步技术。基于 Heartbeat 的 IOC 冗余虽可在主服务器出现故障时,实现由备服务器接管主服务器资源并重新启动 IOC,但重新启动的 IOC 以初始值进行启动,不能实现系统的数据同步。为保障 IOC 重新启动时可恢复系统的运行参数,采用 Autosave 模块实现 IOC 各记录参数的自动保存。整个模块功能如下。

① 参数配置文件 auto_positions. req。需要自动保存和重启恢复的参数全部在 auto_positions. req 中定义。前面提到的记录 rootHost:a. val 需在此定义。

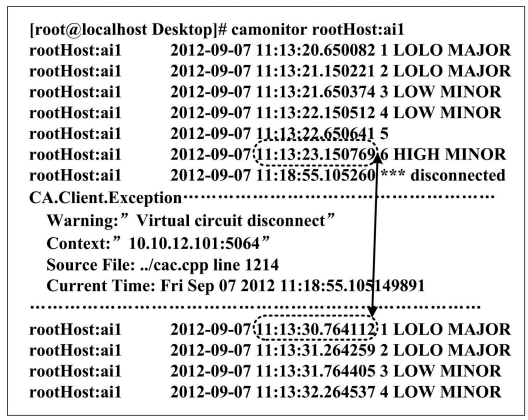
② 参数配置文件 auto_settings. req。需要自动保存的参数放置在参数配置文件 auto_settings. req 中定义。

③ 保存文件 auto_positions. sav 和 auto_settings. sav。IOC 第一次启动时自动创建 auto_positions. sav,用于存放参数值。当 IOC 正常工作时,Autosave 模块按照 IOC 启动脚本中设定的时间间隔自动更新 auto_positions. sav 和 auto_settings. sav 中的参数值。

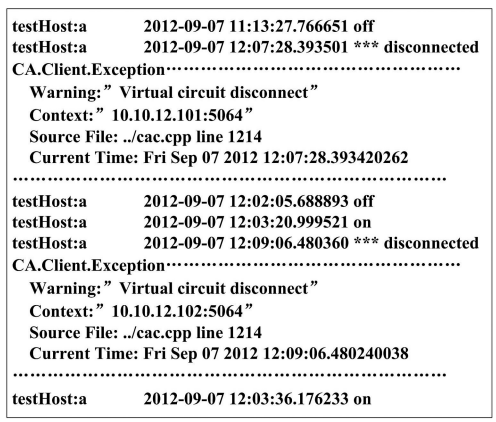
④ 启动恢复配置。在 IOC 的启动脚本中设定 auto_positions. sav 的恢复方式为 PASS0,当 IOC 启动时,Autosave 模块自动恢复 auto_positions. sav 中保存的参数值。

3　系统测试

系统测试在两台服务器上进行,即主备两台服务器,测试界面如图 3 所示。首先对主、备节点故障进行测试。在主、备服务器上依次执行心跳启动命令“/etc/init. d/heartbeat start”,启动 Heartbeat 心跳监测服务,然后在主节点 node1 上执行“/etc/init. d/heartbeat stop”,模拟主节点出现故障。客户端通过 camonitor 进行监测。



(a) IOC冗余切换时间测试



(b) IOC数据信息同步测试

图 3　IOC 冗余切换测试界面

Fig. 3　The switchover interface of IOC redundancy

由图 3(a)可知,EPICS IOC 首先在 node1 上被启动,监测的记录值也来自 node1;当在 node1 上执行“/etc/init. d/heartbeat stop”后,CA Client 与 node1 断开,几秒钟后与 node2 建立连接,继续监测 rootHost:ai1 的值。从图 3(a)可看出,系统断开时间大概有 6 ~ 7 s。这其中包括 node1 执行命令“/etc/init. d/heartbeat stop”,所用时间 3 s 多;接下来的 2 s 为 node2 允许未收到 node1 心跳信号的最长时间;剩下的时间 node2 接管 node1 的资源服务。

图 3(b)所示为 IOC 数据信息同步功能测试,通过监测 testHost:a 记录值进行 IOC 数据信息同步测试。首先 node1 和 node2 同时启动 heartbeat,此时 node1 上启动的 IOC 记录 testHost:a. val 初始值为 off,在 node1 上执行“/etc/init. d/heartbeat stop”命令,模拟 node1 出

(下转第 79 页)