

从 Solaris 平台到 Linux 平台的软件移植研究

A Study of Porting Application from Solaris to Linux

李 昕 左 明 (江苏徐州中国矿业大学计算机学院 221008)

摘要:随着 Linux 操作系统的逐渐普及,许多 IT 公司试图尝试把它们的应用程序和开发环境移植到 Linux 上去。Solaris 环境下用 JAVA,C,C++ 等所开发的应用程序,并不能和所有的系统都兼容。本文主要探讨了从 Solaris 平台到 Linux 平台的软件移植工作流程及相关注意事项,最终使 Linux 成为另一种可选的运行部署平台。

关键词:Linux Solaris 移植

1 为什么要移植到 Linux?

1.1 灵活性

Linux 源代码可以免费获得。Linux 能够运行在多种处理器和配置上,从大型机到 PowerPC、SPARC(TM)、Alpha、PA-RISC 和 Intel 处理器,甚至到个人数字助理和嵌入式系统。Linux 还有完整的 TCP/IP 协议栈,包括带有路由和防火墙功能的 IP 版本 6。最新版本的 Linux 在内核支持 HTTP 服务器,实现了高性能、逻辑卷管理和群集功能,而这使它能够处理以前需要超级计算机才能进行的大型计算和搜索。Linux 还支持很多种文件系统,包括具有可恢复性的日志记录文件系统、具有可靠性的独立磁盘冗余阵列(Redundant Array of Independent Disks,RAID),还有很多种可选的与其他系统兼容的文件系统。

1.2 安全性

Linux 安全性是从内核开始构建的,Linux 的网络安全性是它特别出色的地方,而且 Linux 本身就内建了大范围的路由和防火墙功能。其源代码由遍布世界各地的程序员集体开发,任何 bug 都会被很快发现。

1.3 扩展性

Linux 在支持对称多处理器(symmetric multiprocessor,SMP)系统的方面,可以稳定地支持四路 SMP 系统,在 Web 服务工作负载的八路 SMP 配置上也取得了很好的成绩。在处理更大工作负载时,还可以采用 Linux 的集群技术(clustering techniques)。电影《泰坦尼克号》的图象制作就是采用了数百台 Linux 的集群计算技术。

1.4 可维护性

随着 Linux 的普及,越来越多的公司开始对 Linux 提供多方面的技术支持。而在各大大专院校内,有 Linux 使用经验的人也越来越多。各大 Linux 相关技术网站,新闻组,也为 Linux 爱好者提供了技术交流与讨论的平台,所以 Linux 的专业技术人才将会越来越多。

总之,Linux 的伸缩性、可靠性、稳定性、扩展性、安全性以及对配置的要求比较低,成本低,已经面向企业级应用准备好;专业服务队伍齐全,市场增长快;Linux 的技术和操作特性使它适合很多应用程序,这些都使移植很有吸引力。

2 从整体上需考虑的因素

为了能顺利进行移植,我们首先需从整体上考虑以下若干因素:

(1) 移植到哪个 Linux 目标平台,如硬件平台(intel, sparc……);使用哪个 Linux 分发版和内核版本,如 Red Hat、TurboLinux、Caldera、SuSE,还是其他的?

(2) Linux 目标平台必须支持所有硬件需求吗?举例来说,是否要依赖第三方网卡?是否需要支持网络和存储需求?

(3) Linux 目标平台上是否具备所有所需的第三方包(如类库)、中间件软件、应用程序服务器工具以及应用程序开发工具?

(4) 在移植到目标平台的过程中是否要作出某些改变?举例来说,应用程序是否要改为使用其他数据库系统?

(5) 32 位的 Solaris 应用程序是否需要移植到 64 位的 Linux 平台上?

(6) 理解应用程序体系结构将使移植过程容易一些。举例来说,它是客户机/服务器模型还是 n 层应用程序组件?需要决定先移植哪个软件组件。

3 移植工作流程

首先,明确一点,一般的 Java 应用程序由于其跨平台性很少或根本不依赖于底层机器硬件,因为不管 Java 虚拟机(JVM)运行在什么操作系统上,它都将接受相同的字节码,所以可以选择在一个操作系统上编译 Java 源文件,然后在另一个操作系统上运行编译得到的类文件,所以在移植

过程中无需进行任何修改;移植工作主要集中于.h、.c、.cpp、makefile 编译等方面的调查和修改。

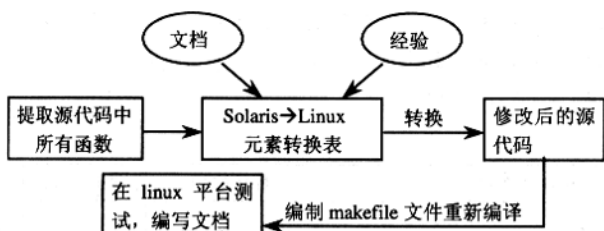


图 1 给出一个移植工作的大致流程

3.1 整理出 Solaris 平台下源代码中所有的函数

Solaris 平台下软件的源代码中往往包含多种文件,如 Java, c, c++ 等。其中由于 Java 语言的跨平台性,所以不在移植考虑的范围之内。而对于 c, c++ 文件,其中部分函数由于平台的差异,其对应参数,返回值,函数名甚至存在与否则不尽相同,所以第一步,要整理出源代码中的所有函数后作互换性调查。可从数百、数千个 c, c++ 文件中提取出所有的函数是一项烦琐的任务。我们需要自己编写一款从源文件中提取函数的工具。具体思路如图 2 所示:

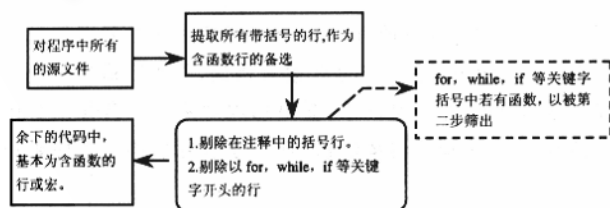


图 2 从源文件提取函数的大致流程

这里,我们可以把结果存为以逗号分隔的形式,如文件名,行号,含函数的行的文本文件,接着改写文件后缀名为 csv(逗号 s 分隔文件),用 excel 打开后,将按列存放文件名,行号,含函数的行,以便于下一步的分析。

3.2 对整理出的函数作互换性调查

对于我们所整理出的函数,可以大致分为以下五类:

- (1) 由项目开发自己定义的函数。
- (2) Solaris 系统函数。
- (3) 如果软件是跨非 Linux 平台,则应有其他平台的系统函数,如 win32api。
- (4) 第三方接口包。
- (5) 宏定义。

这里我们应当关注的是 Solaris 系统函数,因为一般情况下,其他四类函数是无须替换的。为了顺利的把软件从

Solaris 平台移植到 Linux,我们需要对项目出现的 Solaris 平台下的每一个函数作细致的调查。可以利用 Linux 系统和 Solaris 系统的 man 帮助页,会发现此二平台下的函数往往有以下差异:

- ① 系统函数的返回值,参数顺序,数量,参数类型不同。
- ② 在 Solaris 平台下存在的系统函数在 Linux 平台上却找不到。
- ③ 系统函数所需的头文件不同。

之所以会有这些差异,我们应该先从 Linux 和 Solaris 系统自身来研究:

3.2.1 Linux 和 Solaris 系统平台的差异

(1) 系统调用和 C 库。Solaris 内核提供逻辑卷支持、文件的 ACL(access control list,访问控制列表)管理和系统审计日志功能。对于这种功能,可用的 Linux API 与 Solaris 上的 API 有所不同。Linux 上缺省条件下不支持的还有 STREAMS。我们需要对使用 STREAMS 进行联网的应用程序作出修改,才能使用 POSIX 套接字作为替代。

(2) C++ 库。Solaris Forte 6 C++ Compiler 包括三个类库: -compat=4 的复数、经典 io 流和标准 C++ 库(它包括标准复数和 io 流)。GCC 只包括标准的 C++ 库。如果被移植的应用程序使用 -compat=4 复数库或经典 io 流库,那么就需要修改应用程序,以使用标准 C++ 库中包含的这些库的标准版。

(3) 数学库。Solaris 数学库包括 ASCII 编码十进制数的转换函数。Linux 数学库中没有这种功能。

(4) 线程 /LWP(Light Weight Process,轻量级进程)支持。Solaris 支持 POSIX 线程和特定于 Solaris 的线程模型。如果被移植的应用程序使用特定于 Solaris 的线程 API,那么最好将应用程序修改为使用 POSIX 线程。Linux 上支持 POSIX 线程,而且它更具可移植性。

(5) 进程管理: /proc 文件系统。/proc 伪文件系统为访问内核数据结构提供了一种便捷的方法。在 Solaris 上, /proc 包含关于活动进程和线程的信息,并提供控制这些进程和线程的接口。Solaris 上的 /proc 控制接口一般由调试者用来跟踪程序执行情况。在 Linux 上, /proc 并不提供进程控制接口。要控制进程以便在 Linux 上进行调试,请使用 ptrace() API 代替 /proc。

3.2.2 对二平台系统函数的一些互换性说明

总的来说,大多数系统函数在 Linux 和 Solaris 上都是通用的,如 closedir, closelog, ctime, dlclose, dlerror, execlp, execlv, execvp, fclose, fdopen 等等。

对于所要替换的函数,我们以以下三种情况作为标记:

- (1) 追加。追加是指在 Linux 下的某系统函数所需的

头文件比其对应的在 Solaris 平台下的系统函数的头文件多,所以当把 Solaris 平台下的源代码移植到 Linux 平台下时,我们要添加所缺的头文件。

(2) 修改。修改是指 Linux 与 Solaris 平台下系统函数的返回值,参数顺序,数量,参数类型,头文件不同,当把 Solaris 平台下的源代码移植到 Linux 平台下时,要进行相

应的修改。

(3) 替换。替换则指在 Solaris 平台下所存在的系统函数,在 Linux 平台下不存在或函数名不一致。这时,需要我们在 Linux 平台下用相应的函数或自己写的功能一致的函数替换。如图 3 所示:

| 函数名 | linux 平台 | Solaris 平台 | 替换原因 | 替换方式 |
|-----------------|--|---|--------|------|
| Mmap | # include <sys/mman.h> # include <unistd.h> | # include <sys/mman.h> | 头文件不一致 | 追加 |
| reboot | int reboot(int flag) | int reboot(int howto, char * bootargs); | 参数不一致 | 修改 |
| sigsend kill | int sigsend(ldtype_t ldtype, id_t id, int sig); | int kill(pid_t pid, int sig) | 函数名不一致 | 替换 |

图 3 二平台函数差异举例

| Solaris | Linux |
|--|---------------------------------------|
| CC = /opt/SUNWspro/bin/cc | CC = /usr/bin/gcc |
| CXX = /opt/SUNWspro/bin/CC | CXX = /usr/bin/g++ |
| OTHERLIBDIR3 = /opt/SUNWspro/WS6U2/lib | OTHERLIBDIR3 = /usr/lib |
| LIBDIRS = -L/opt/SUNWspro/lib | LIBDIRS = -L/usr/java/jdk1.3.1_08/lib |

图 4 二平台编译环境差异举例

3.3 对 Solaris 下的 makefile 文件进行修改

由于 Solaris 与 Linux 平台下编译器的差异,相应的编译选项也要作适当的修改。主要有以下两个方面:

- (1) 编译器和库文件或其他文件的路径,如图 4:
- (2) 其他一些常用编译选项的修改,如图 5:

| Solaris | Linux |
|---------|----------|
| -mt | -pthread |
| -KPIC | -fPIC |
| -xO2 | -O2 |

图 5 二平台编译环境差异举例

当把源代码各个目录下的 makefile 文件修改完毕后,首先针对每一个 makefile 文件单独编译测试,通常,要重复好几次才能够编译出没有问题的代码。要确保使用了 -Wall 选项,以捕获所有警告消息,最后再进行整体调试。

4 结束语

把软件从 Solaris 平台向 Linux 平台上移植,不但可以

极大地为用户降低总拥有成本,而且可使系统性能得到显著提高。而移植过程是个不断修改源代码,不断在目标平台编译测试的过程,在此过程中,会更加加深对两种平台体系结构的理解,获得丰富的移植经验,从而使以后的移植工作更加顺利。

参考文献

- 1 曙福等, Linux C 高级程序员指南 [M], 国防工业出版社, 2001。
- 2 萨贝尔等, Solaris 技术指南 [M], 机械工业出版社。
- 3 Malcom Zung . A roadmap to moving your applications over to Linux [EB/OL]. <http://www-900.ibm.com/developerWorks/cn/linux/porting/Solaris/roadmap/Index.shtml>, 2001.
- 4 Lee Cheng, . Technical guide for porting applications from Solaris to Linux[EB/OL]. http://www-900.ibm.com/developerWorks/cn/linux/porting/Solaris/guide1/Index_eng.shtml, 2001.