

## 单 Linux 电信基站软件解决方案

自从 1991 年 Linux 第一次面世以来，任何人也不会想到这个不起眼的操作系统现在会应用得如此广泛。无论是从嵌入式领域的智能手机到大型计算服务器都可以找到它的身影，而且无论什么工作它都完成得非常出色，因为 Linux 的开放性决定了其自身可以根据使用领域而高度定制。作为一个开源软件，当前业界的工业标准可以最快地支持任何刚发布的硬件产品。同时，被世界各地的商业公司和志愿者一起参与开发和测试，它也变成了十分稳定的操作系统之一。

Linux 的设计原则决定了它的无所不能，但是对于特定领域它并不是最优的解决方案。同样对于嵌入式电信领域来说，为了追求满足电信领域的高实时性和高吞吐量要求，大家一般都采用传统的 RTOS+Linux+运行环境。

在传统架构中（参见图 1），数据面处理采用 RTOS+CPU/DSP 方式，主要处理底层协议数据业务，通过 RTOS 的高实时性及多核 CPU/DSP 来达到性能需求。Linux 自身的实时性较差，但是比较适合于高负载的复杂协议处理，因此一般用于完成用户控制平面协议的处理和 IP 转发等功能。这个方案在性能上可以保证电信产品的使用需求，但系统中存在多个处理器和多个不同操作系统平台，对于用户的维护和后续升级管理带来很大负担，同时也无法满足当前的低成本和低功耗要求。

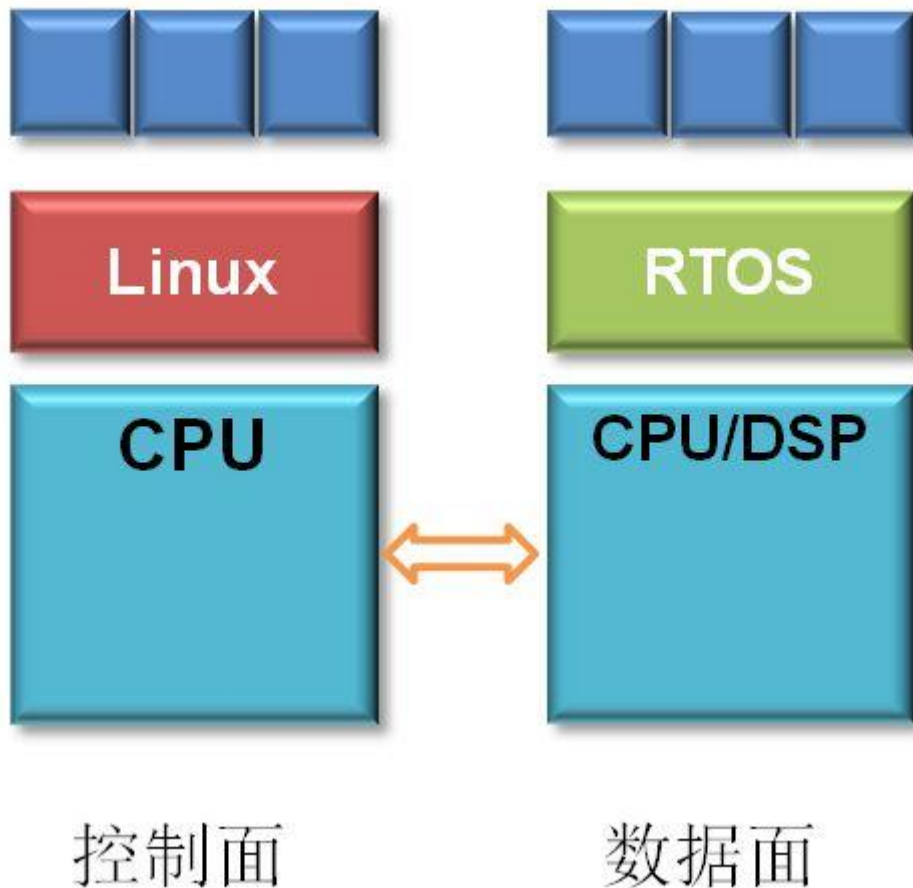


图 1:典型的电信基站系统结构图。

随着近年来处理器硬件技术的飞速发展，处理器的能力越来越强，多核平台已经广泛地应用在电信领域，尤其是在 ARM 公司将 Cortex A15 引入电信基站领域后，整合单芯片解决方案将成为业界主流。但是软件领域的发展相对滞后，尤其是如何在 SoC 单芯片硬件环境下保证用户已有成熟业务软件的继承使用，是每一个电信设备制造商关心的问题。

最简单的解决方案是如图 2 将 Linux 和 RTOS 按照 AMP 方式独立运行在多核 CPU 的不同核上。这种方案可以保证高实时能力，同时可以避免用户应用程序的修改。但是该方案最大的问题是两个操作系统要求在硬件资源使用上独占，需要在使用前静态分配好，系统运行后则无法动态分配，没有办法做到资源共享。同一个硬件环境下同时运行两个独立的系统也带来了运行安全方面的隐患。综上所述，AMP 方案前期设计简单，方便于先前成熟软件集成，但是后期维护成本较高，同时造成了硬件资源的浪费。

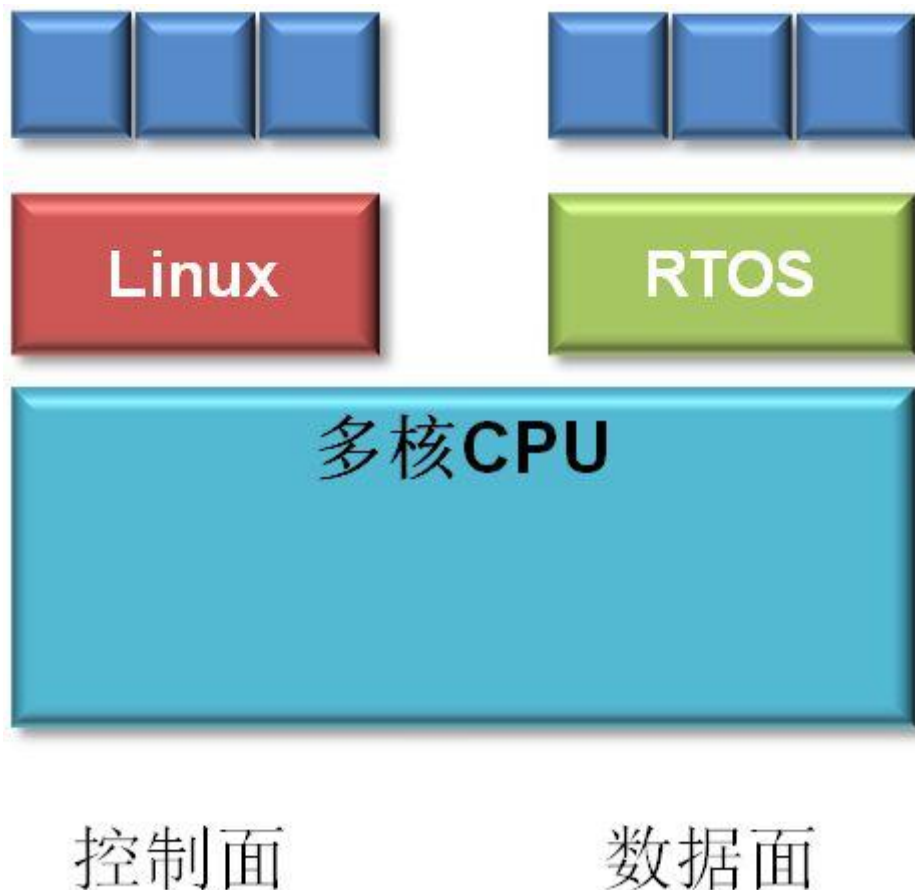


图 2:AMP 解决方案架构图。

Linux KVM 是基于 Linux 内核的虚拟机制，通过 KVM 可以在其上虚拟出原有 RTOS 所需要的运行环境（参见图 3）。例如我们可以将原有业务软件以及原有的 RTOS 一同移植到 KVM 的虚拟环境中，这种方式对业务应用软件不需要任何修改。同时，整个系统硬件资源也由 Linux 统一分配，相对 AMP 方案避免了硬件资源的浪费。KVM 机制可以做到系统的安全隔离，保证了电信系统的运行安全。但是该方案存在实时性较差（尤其是运行在 Linux 内核上 RTOS 的实时性），很难应用于电信协议等时隙要求较高的场合。

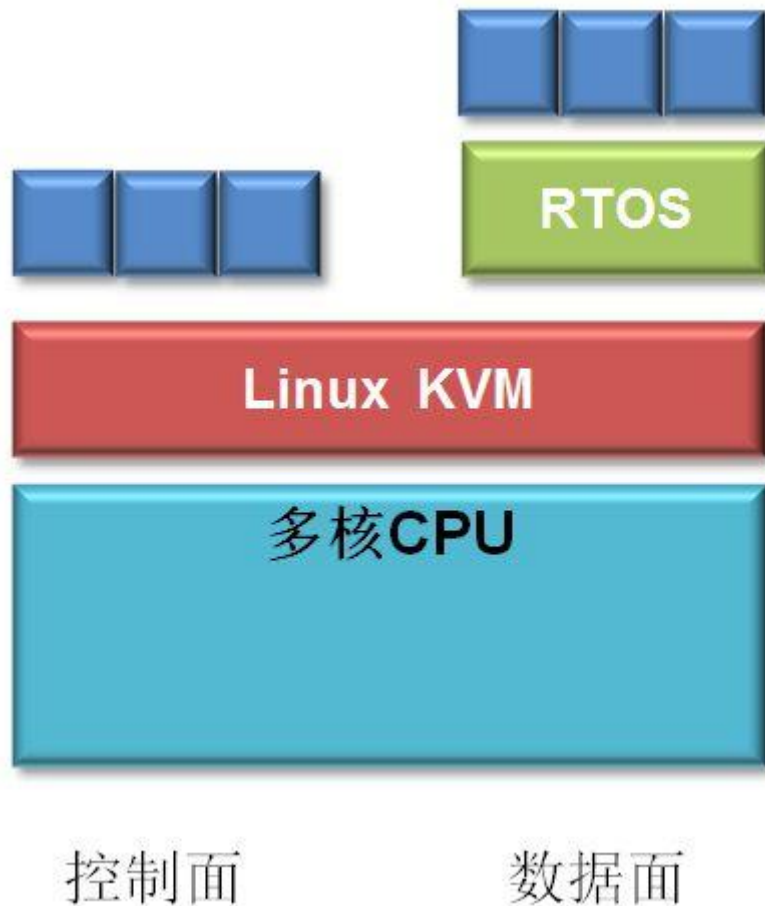


图 3:Linux KVM 解决方案。

如何在 Linux 基础上提供更加高效的实时处理能力，同时又简化多操作系统并存带来的软件复杂性？为此，Enea（宜能）公司在用户空间实现了 LWRT（Light-weight Run-time Threading, 轻量级运行环境）。LWRT 是针对多核使用场景设计的，它基于 Linux 用户空间的标准 Pthread 作为运行载体，并在此基础上实现了可确定的调度方式、信号机制和内存管理。通过这种方式，LWRT 内部的线程切换和信号处理不再依靠 Linux 内核调度机制，而是完全在用户态的 LWRT 内部实现。这在减少系统开销的同时，也实现了严格意义上的实时切换。图 4 描述了基于 LWRT 电信基站软件架构。

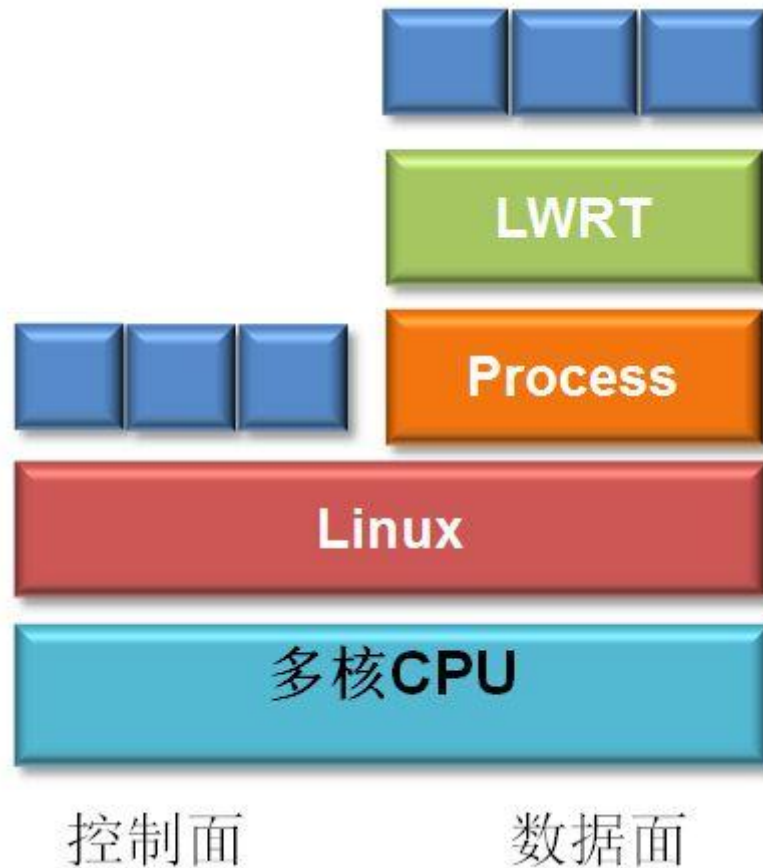


图 4:LWRT 多核解决方案。

为了保证整个 LWRT 的实时性，我们需要将其承载的容器 Linux Pthread 绑定到一个物理内核上，通过这种方式使 LWRT 像单独运行在这个物理内核的传统 RTOS 一样。同时，我们在 LWRT 上设计了用户态中断处理模型，将整个中断处理过程放在用户态进行。在 Linux 内核只需要在上半部中向 LWRT 发出中断信号，而所有的下半部处理完全实现在用户态的 LWRT 中。整个中断处理不再有内核互斥锁的影响，从而保证了高实时处理能力。

先进的设计保证了高效的性能，在一个实际测试环境中对于一个简单的“信号触发-线程切换-信号接收”的线程切换模型，LWRT 可以在不超过 200 个周期内完成，且该数据不会随着 CPU 占用率增高而受到影响，而在同一硬件条件下，采用 Linux 标准线程完成同样的切换动作需要大约 3500 个时钟周期，测试解决见图 5。整个切换过程达到了高实时性要求。Linux 上将不再只是运行控制面代码，用户完全可以将数据面处理迁移到 Linux 上。基于这种方案我们将不再需要 RTOS，而是将 RTOS 的功能基于 LWRT 来提供。

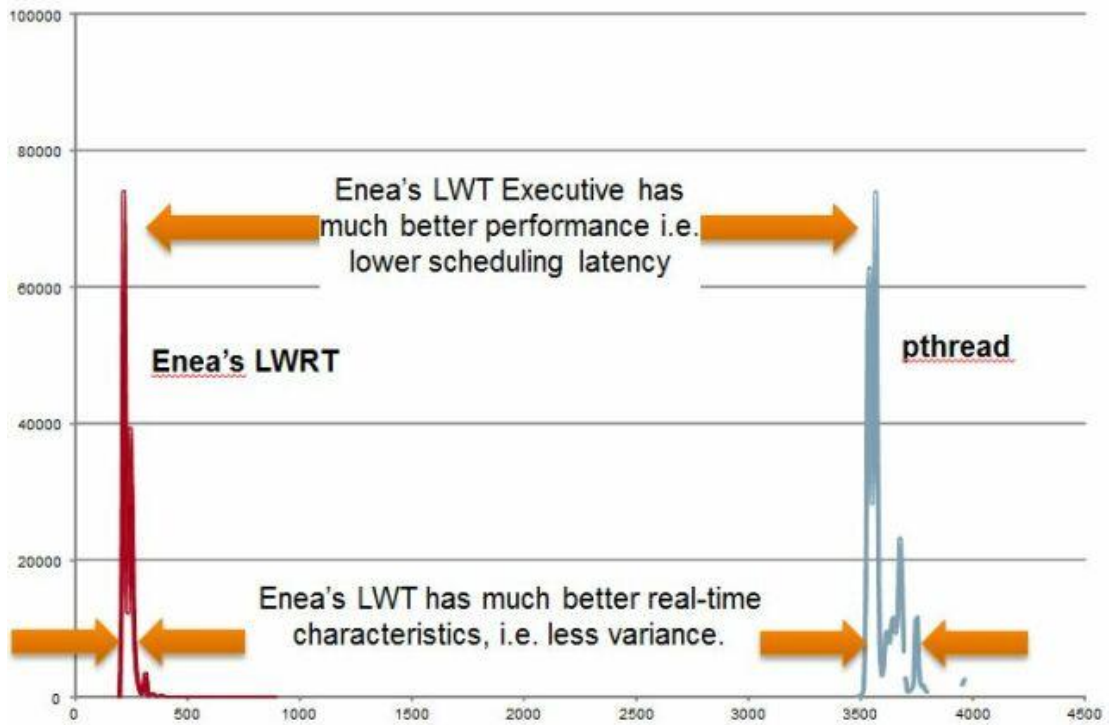


图 5:LWRT 线程切换和 Pthread 线程切换对比。

目前整个电信领域向 IP 化发展，针对于 IP 数据处理，Enea 实现了 PAX（数据包加速基础）解决方案。PAX 同样运行在 Linux 的用户态。

它实现了基于图形方式现实的 IP 报文分析处理流程，在 PAX 中可以完成 IP 转发、IPSec、流量控制和 ARP/WLAN 网桥，以及数据包检查 DPI 等 IP 通讯中最消耗资源的处理。针对于数据面的这些处理过程不再依赖于原有 Linux 内核 IP 协议栈实现，而是由 PAX 及其相关的硬件加速器一起完成。这样便将原有 Linux IP 转发能力提高数倍，而非常适合于可扩展、高吞吐量和低时延的 IP 协议处理，尤其适用于基于 IP 的无线基站、控制器、智能防火墙、入侵检测系统和防病毒扫描网关等产品中。同时，PAX 提供的图像化追踪(tracing)和分析(profiling)功能可以方便用户找出系统瓶颈所在，进行针对性优化。

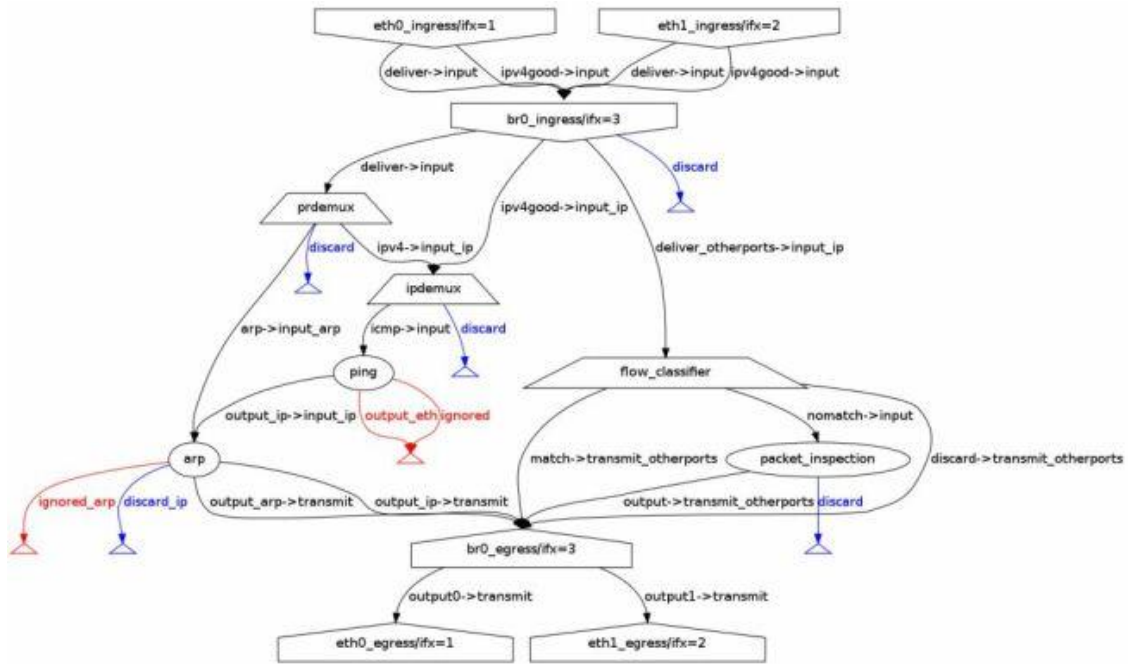


图 6:PAX 数据处理流程图。

电信级应用对产品的可靠性和稳定性的要求极高，为此 Enea 将多年成熟的软件开发流程引入了 Linux 的开发和维护中。这样保证了所有用户的可选软件包都是不同硬件平台下严格验证过的。作为开源社区成员，我们也密切监视开源社区的安全漏洞和 Bugfix 修复情况，并在全面验证后将其纳入 Enea Linux 产品中。为了更好地保护用户在软件领域的投资，在 Enea Linux 的升级版本中可以保证 Enea IPR 的向后兼容。这将确保对每一个硬件平台的 Linux 内核进行长期支持。另外，所有目标平台均采用 Linux 3.0.x 系列内核，并且每 18 个月更新一次内核版本。根据与客户达成的协议，Enea 也可更加频繁地更新内核版本。

在 Enea Linux 提供的开发工具中，用户依然可以找到熟悉的 Opitma 调试分析工具。这缩短了用户学习新开发工具的时间，同时也集成了 LTTng、oProfile、Gcov&gprof、valgrind 等常用开源分析工具和 Qemu 主机仿真环境。

Enea 作为一家全球领先的 3G 和 4G 基础架构操作系统解决方案供应商，拥有 40 年通信驱动产品解决方案研发经验，同时，Enea 也是高可靠性和性能要求极高的软件平台开发的全球引领者。Enea 在实时操作系统和高可靠性中间件方面，必将会将其具有的丰富经验分享给客户，解决客户面临的问题。并且在成就客户的同时，实现自己的价值也是 Enea 公司一直追求的不变承诺。