OS 论文阅读（1）

刘炜
2014210878

June 7, 2015

1 Dune:Safe User-level Access to Privileged CPU Features [1]

1.1 思路

先讨论 DUNE 这篇文章，主要是因为 IX 的 dataplane 可以认为是在 DUNE 的基础上做的一个 application，因此先说明 DUNE 的设计思想。

DUNE 的设计是为了给应用提供安全地使用特权 CPU 属性的方法。与虚拟机不同的是，虚拟机中的系统和原系统是相互隔离的，不具有相同的上下文。而 ExoKernel 要求替换整个的 OS 栈来实现一个新的 library OS，工程量太大。而 DUNE 的设计只需要在 Kernel 中增加一个相应的 DUNE 模块（约 2500LOC）负责配置和管理硬件虚拟化，运行在 host mode 中，在应用中加入 libdune（约 6000LOC）帮助应用利用硬件虚拟化的资源，运行在 guest mode，来完成整个系统。

以垃圾回收为例，guest os 中的运行着的 guest page table 与 host os 中的 host page table 不同，因此可以供应用安全地使用，进而提高垃圾回收的效率。对于应用沙箱和权限隔离的例子，也有很好的应用。

总的来说，DUNE 可以看做是一个比 Exokernel 更为轻量级，而比虚拟机，guest os 有与 host os 更紧密的关系，有更多可配置的选项，将应用跑在 guest os 来应用硬件虚拟化的系统。

1.2 讨论

鉴于 guest os 和 host os 没有严格的分区隔离，如果用来做应用沙箱，利用 guest os 的 syscall 从沙箱逃逸似乎比从虚拟机中逃逸的难度要小了，如果系统大规模部署实践，这方面可以深入探索一下。

1.3 体会

这个工作如果部署之后的效果很好的话，有机会跟虚拟机技术一样大规模商业化。

2 IX: A Protected Dataplane Operating System for High Throughput and Low Latency [2]

2.1 思路

作者的出发点依然是硬件资源的计算性能很快，但系统由于是几十年前的设计，跟现在的使用情形不同，所以成为性能瓶颈。作者的方法主要也是利用硬件虚拟化来使得应用可以更加直接的访问到硬件资源。思路和另一篇 best paper《arrakis》非常像，以下主要描述二者的不同。

Arrakis 系统想法在于 bypass kernel，它的想法比较直接，要让应用访问到硬件资源，那么就直接用硬件虚拟化的虚拟设备拿给应用使用，而将之前 kernel 的工作移到硬件中或者移到应用中。总的来说，它的安全保证和管理方面相对要弱一些，但性能会有比较显著的提高。

而 IX 的工作是在 DUNE 的基础上做的，IX 可以认为是 DUNE 的一个应用实例。在 DUNE 的演讲提问中，有人问到 DUNE 是否能用到对 IO 设备上去?IX 就是一个最好的回答。IX 并不是 bypass kernel，而是在 guest os 中运行应用，而 guest os 可以利用很多硬件虚拟化的资源，因此达到提高 I/O 任务处理速度的目的。这样会有更好的管理与控制，在安全方面也可以比直接绕过 kernel 的方法做得更好一些。另外 IX 的工作是专门针对网络协议栈的，而 Arrakis 还做了存储控制设备的虚拟化相关的工作。

2.2 讨论

从两篇 best paper 有相同数据集的 memcache 的性能对比上来说，二者是非常接近的，IX 略微好一点。但我个人觉得，从总体长远上来说，Arrakis 的性能会比 IX 更好，毕竟 bypass kernel 的方法更加直接。但综合安全，控制等角度上看，IX 是一个看上去更加实用的系统。

2.3 体会

之前读 Arrakis 的文章，对于相关工作的了解还是不够，以至于对其的认识不够全面，还需要更多的阅读的实践才能更加领会学习文章中没有说到的本质性内容。

References

[1] A. Belay et.al, "Dune:Safe User-level Access to Privileged CPU Features", In OSDI'2012.

[2] A. Belay et.al, "IX: A Protected Dataplane Operating System for High Throughput and Low Latency", In OSDI'2014