**Agenda**

* Alternate approach based on VFIO

**Next meeting** – Tuesday, 12/17/13 9:00AM PST

**Verbs Compatibility – see Christoph Lameter’s slides VFIO-Verbs.odp, VFIO-Verbs.pdf**

Not a proposal, just some thoughts that may ultimately form the basis for a proposal.

Basic proposal – put everything into user space, and use the VFIO facilities provided in the kernel to allow mapping of features into user space.

VFIO is predicated on availability of an IOMMU in the northbridge (or equivalent). Cray observes that past experience shows poor performance using the standard IOMMU.

Effectively, the idea is to move the device driver into a privileged user-space daemon. Security and isolation protections would require that there be a privileged entity in user space that is capable of programming the TLB.

Intel posits that the translation mechanism could be handled at the provider layer, opaque to the application.

Second issue is concern over the proliferation of function vectors. Alternative is to statically link the application code directly to the provider, thus eliminating some function indirection which in theory should improve code path lengths. Redhat observes that this creates difficulty in maintaining the code. Intel suggests an analysis of the difference in code paths on the hypothesis that the increase in code paths is not significant.

Redhat response to VFIO – skeptical as to whether performance will be better or worse. Redhat experience with IOMMUs, which are usually ignored. When the IOMMU is engaged, it has a drastic impact on performance. You obviously can’t hand control of the physical device to the user. The only way to make this performant would be for the provider to nail down large pieces of memory and hand out chunks as needed.

Possible next steps – deeper dive into VFIO and how it works. Probably defer a final decision on this

**Next meeting:**

Tuesday, 12/17/13 9-10AM PST

Dial-in info: 1-888-875-9370 or 1-916-356-2663 bridge 5 PC 960895963

* Review strawman categorization of applications into classes