

Which is Easier to Program?

Companion Accelerator Node
Many-core Self-hosted Node
Multi-core Node

Systems to be delivered in the next 4-5 years



- **Companion accelerator Node sharing memory with host**
 - Coral systems TB Delivered by IBM/Nvidia
 - AMD APU systems
- **Many-core Node**
 - Trinity and Cori systems TB Delivered next year
- **Multi-core Node**
 - Haswell/Broadwell/Skylake
 - ARM

High Level Comparisons



	Companion Accelerator	Many-Core	Multi-Core
Number of threads required	1000s	100s	10s
Number of MPI Tasks/Node	1-4	4-30	4-32
SIMD Length (64-Bit)	8-32	8	4-8
Memory Hierarchy	Virtual/User Controlled??	User Controlled	NUMA
Cache Architecture	Insufficient	Could be an issue	Well understood
Scalar Performance	Very Poor or Host – implies memory movement	Poor	Good

Programming issues with Companion Accelerator

- **Must do tremendous amount of threading – cannot be MPI**
- **Must SIMDize more code – Can we say Vectorization**
- **Scalar code on the accelerator is very slow, on the host requires memory movement.**
- **How does the Memory manager work? Can the user control data movement? Virtual Memory??**
- **Amount of registers/cache per MIMD processor is too small**

Who Said this:

We don't use Virtual Memory, you can't fake what you don't have

Seymour Cray

Memory Management must be user controlled

Programming issues with Many-core System



- **Must do threading – all MPI will not work well, it will work**
- **Must not have scalar code in important areas – must vectorize as much as possible**
- **Ideally have parallel, Vectorizable loops**
- **Managing Memory Hierarchy will be a challenge**
- **Cache optimization within Memory Hierarchy will be a challenge**

Programming issues with Multi-core System

- **All MPI will still work –**
 - The main reason MPI/OpenMP on the Node does not perform well is that the OpenMP is poorly implemented. Most of the implementers do not want it to win.
- **Higher level caches become larger – while some optimization can be performed, many take what they get**
- **Vectors become more important**

Which is Easier to Program?

ANSWER

- 1) Multi-core Systems; but, you'll pay more for power
- 2) If you want a performance portable application, none are