Exceptional service in the national interest

Sandia National Laboratories



Abstract Models for Exascale Architectures

Two Years of Codesign

S.D. Hammond

Scalable Computer Architectures, Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

What is this guy talking about?

Sandia National Laboratories

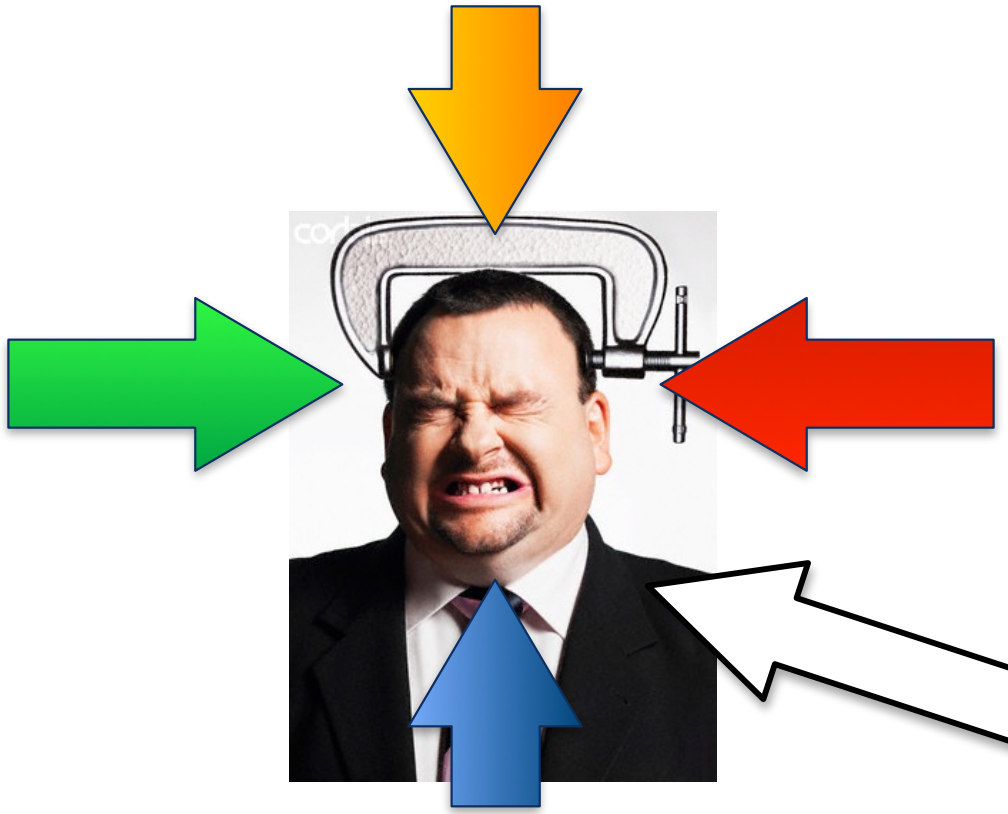
- Three years in code sign activities for the DOE ASCR and NNSA/ASC
  - Its been a painful process
- First working examples of NNSA mini-apps on:
  - Intel's Knights Ferry/Corner
  - NVIDIA Kepler K20-series
  - IBM POWER7+
  - AMD's APU (CPU+GPU)
  - ARM64
- Has been a <u>significant</u> undertaking for every platform

What do I do all day?

Sandia National Laboratories

System Software
(MPI, Runtimes, Compilers…)

Hardware
(IBM, NVIDIA, AMD, Intel, Cray, ARM, Micron…)



Algorithms
(Trilinos,SIERRA, LAMMPS…)

Mini Apps
(miniFE, LULESH, CoMD, miniMD, SNAP…)

I'm interested in your feedback and comments on Codesign

Acknowledgements

Sandia National Laboratories

http://www.cal-design.org/

- Lots of input from many partners, colleagues, industry, academia *etc*

- Particularly our Computer Architecture Laboratory (CAL) partners at Lawrence Berkeley.
  - Dave Donofrio, Dan Burke, Farzad Fatollahi-Fard, John Bachan and John Shalf

Wouldn’t it be great if…

- Our existing codes didn’t need to change
  - Would be cheaper and save many sleepless nights

Wouldn’t it be great if…

Sandia National Laboratories

- Our existing codes didn’t need to change
  - Would be cheaper and save many sleepless nights
- If there was a consistent set of models we could program to?
  - Reduce the ambiguity and uncertainty
  - Enable reasoning about algorithms possibly before a programming model

So Let's Do It…

- Time for us to start thinking about **abstract models** of an Exascale machine

- **Abstract Machine Model** - a block diagram which describes:
  - Where you place your data structures
  - Where you run your kernels

- **Proxy Architecture** - an abstract model with parameters attached

To tell us your problems
To tell us why this won't work
To tell us what we missed

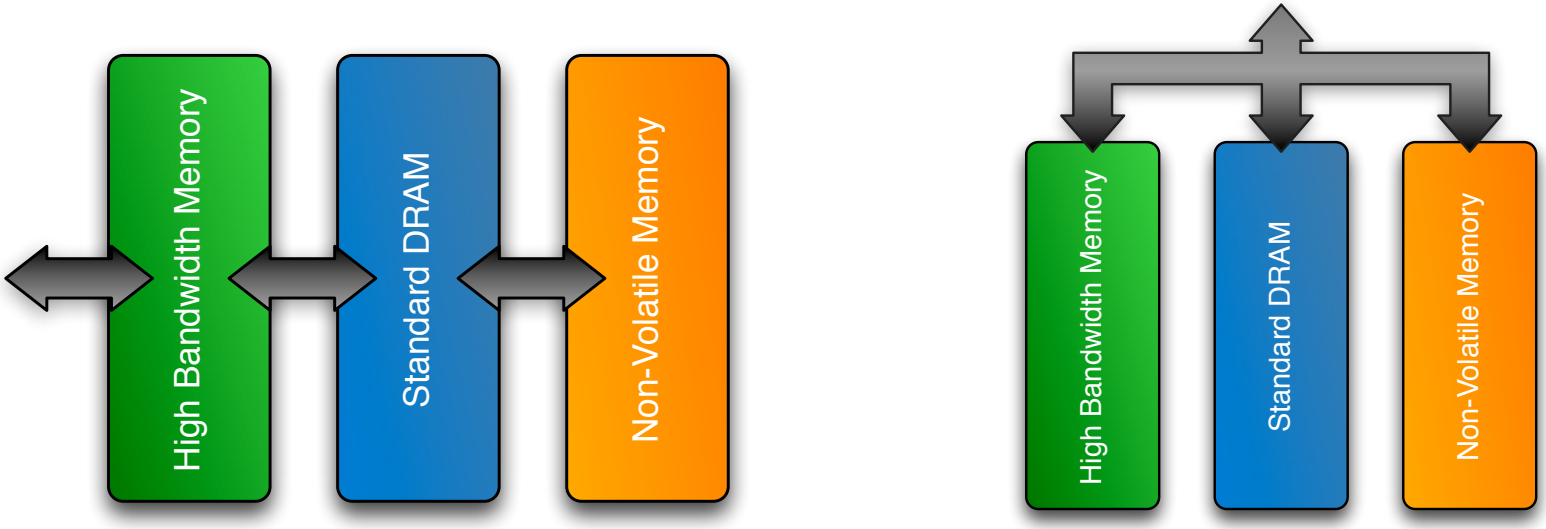
Abstract Machine Models

Sandia National Laboratories

- Want to keep it *really* simple as a first cut
  - Make Exascale class computing available for **everyone**
  - Improves generality and enables a clean(er?) mapping

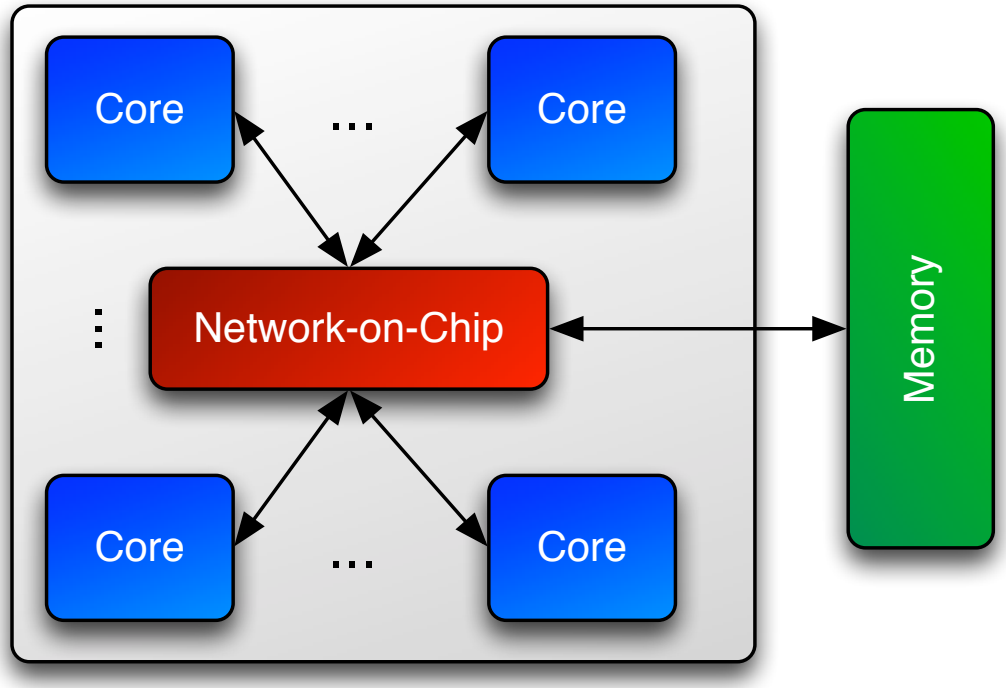
- Build out of some basic blocks:
  - The concept of a (latency optimized) **core**
    - Can be SMT, SIMD, out-of-/in-order
  - A **memory** block
    - Can be multiple technologies or levels (inc. remote node)
  - **Accelerator** block
    - Typically SIMT style, throughout optimized
  - **Network-on-chip**

First Stop: Memory



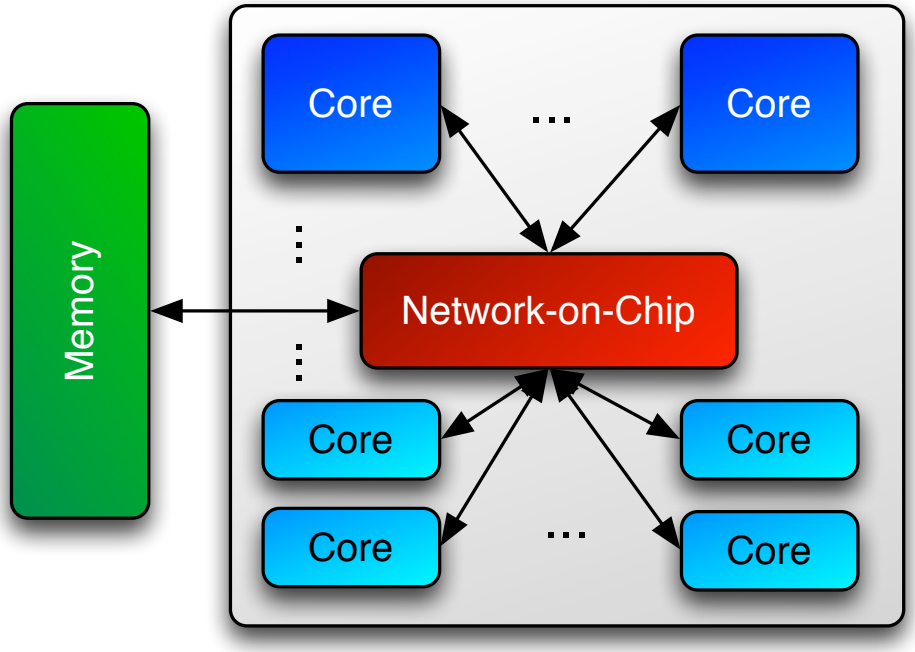
- Multi-levels of storage - fastest, fast, real slow
- Three approaches:
  - Cached-style
  - Explicitly managed
  - Blend both

Homogeneous Many Core



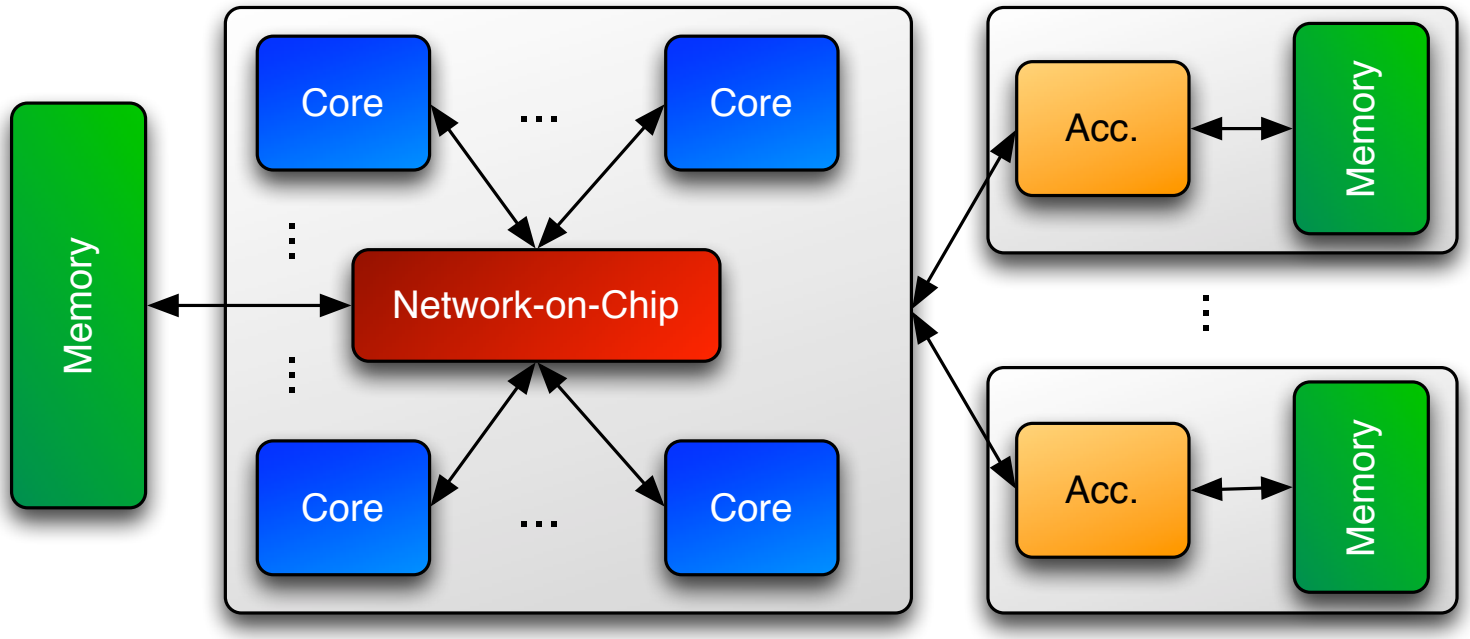
- Cores are approx. equal in performance
  - Subject to DVFS, thermal slowdowns, process variations
- Think of Sandy Bridge, POWER7+, Xeon Phi, some ARM cores

Heterogeneous Many Core



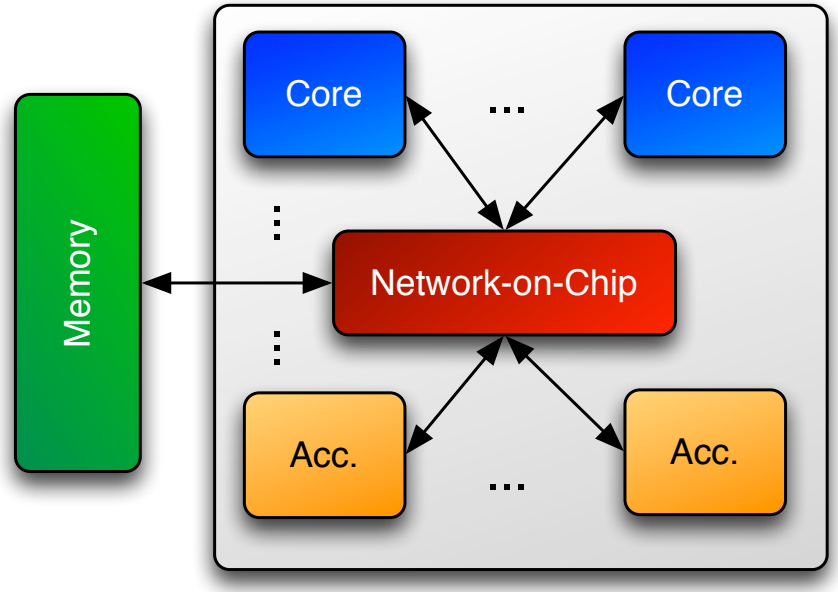
- Multiple core variants, blend lightweight and heavyweight latency optimized cores
  - Shared ISA?
- Think of ARM big.LITTLE concept designs

Discrete Accelerators



- Add discrete accelerators with local memories
  - Discrete in terms of the silicon package
- Designs include NVIDIA/ATI-GPUs or a PIM concept
- Pressure around kernels and data structure allocation

Integrated Accelerator

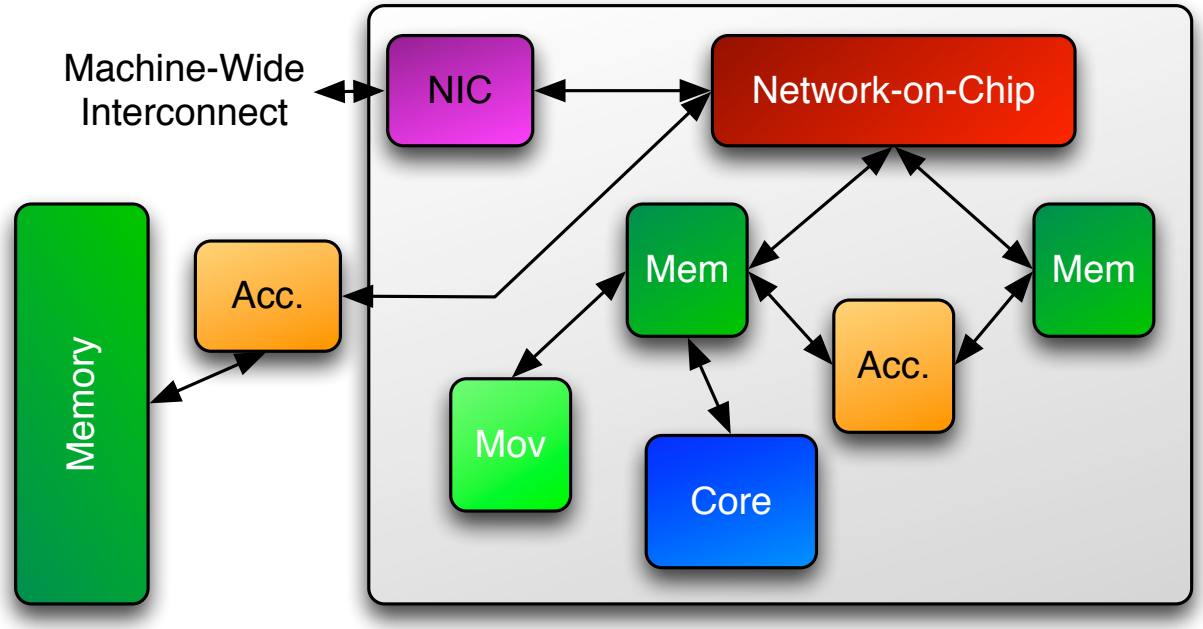


- Integrate the accelerators into the package
  - Give up cores to gain blend and shared resources
  - Shared memory controllers
- Design is close to AMD's Fusion APU and Intel's IRIS products
- Possible for a SoC future design?

Questions for You…

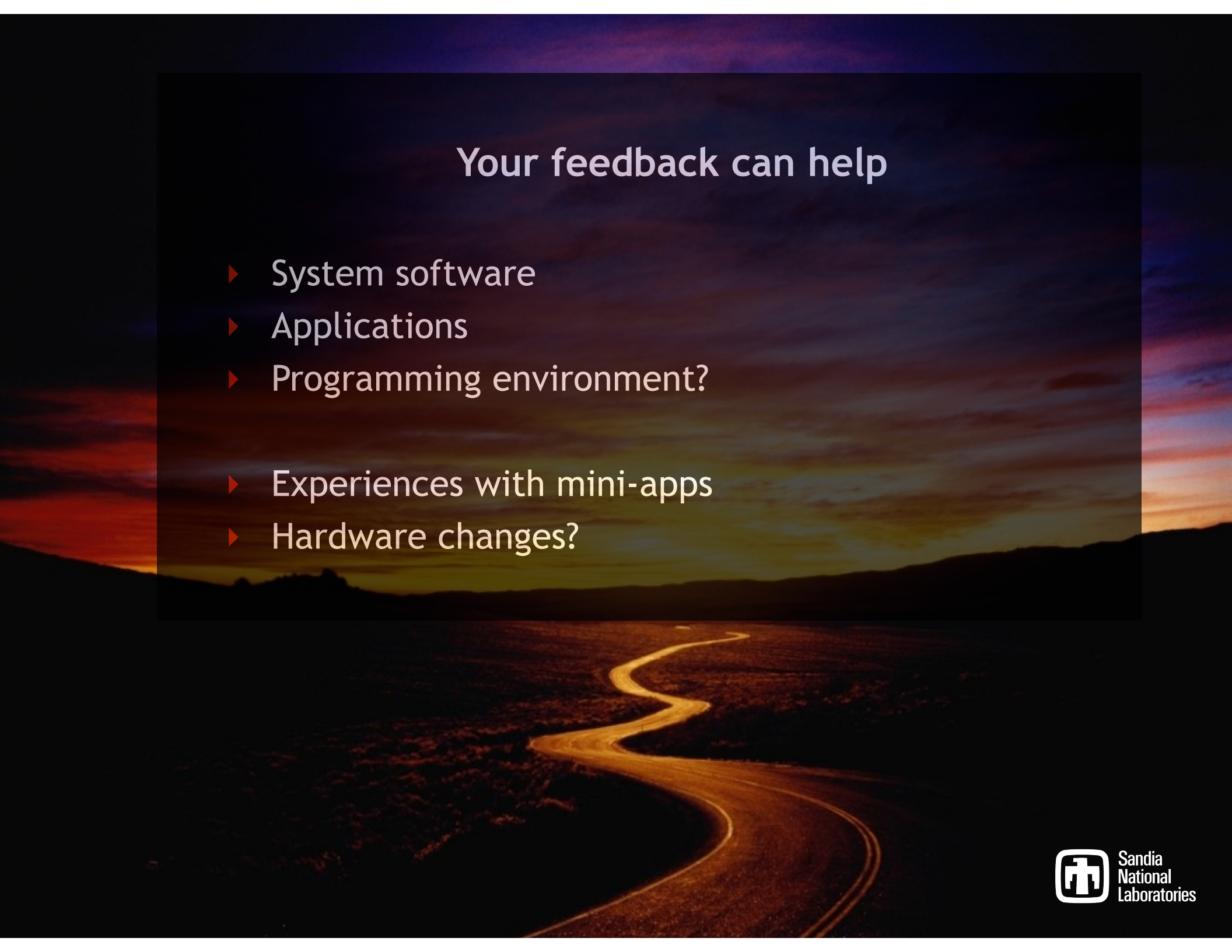
- Are these kind of models useful?
  - Its perfectly fine to say no (but tell us why)
  - Can you reason about algorithm design?
  - Remember - this is ***not*** for heroes
- Given these blocks what would you do for your application?
  - Go to the local Exascale computing store, fill up your shopping cart

Concept Design



Design by Dave Resnick,
Scalable Computer Architectures, Sandia National Labs, NM

- Concept design to blend analytics and conventional physics
- Heavily specialized core/accelerator types


Your feedback can help
System software
Applications
Programming environment?
Experiences with mini-apps
Hardware changes?
Sandia
National
Laboratories

Sandia National Laboratories

Exceptional service in the national interest

sdhammo@sandia.gov