HPC Software Vision for Exascale Computing and Beyond

Dr. Robert W. Wisniewski
Chief Software Architect Extreme Scale Computing
Senior Principal Engineer, Intel Corporation

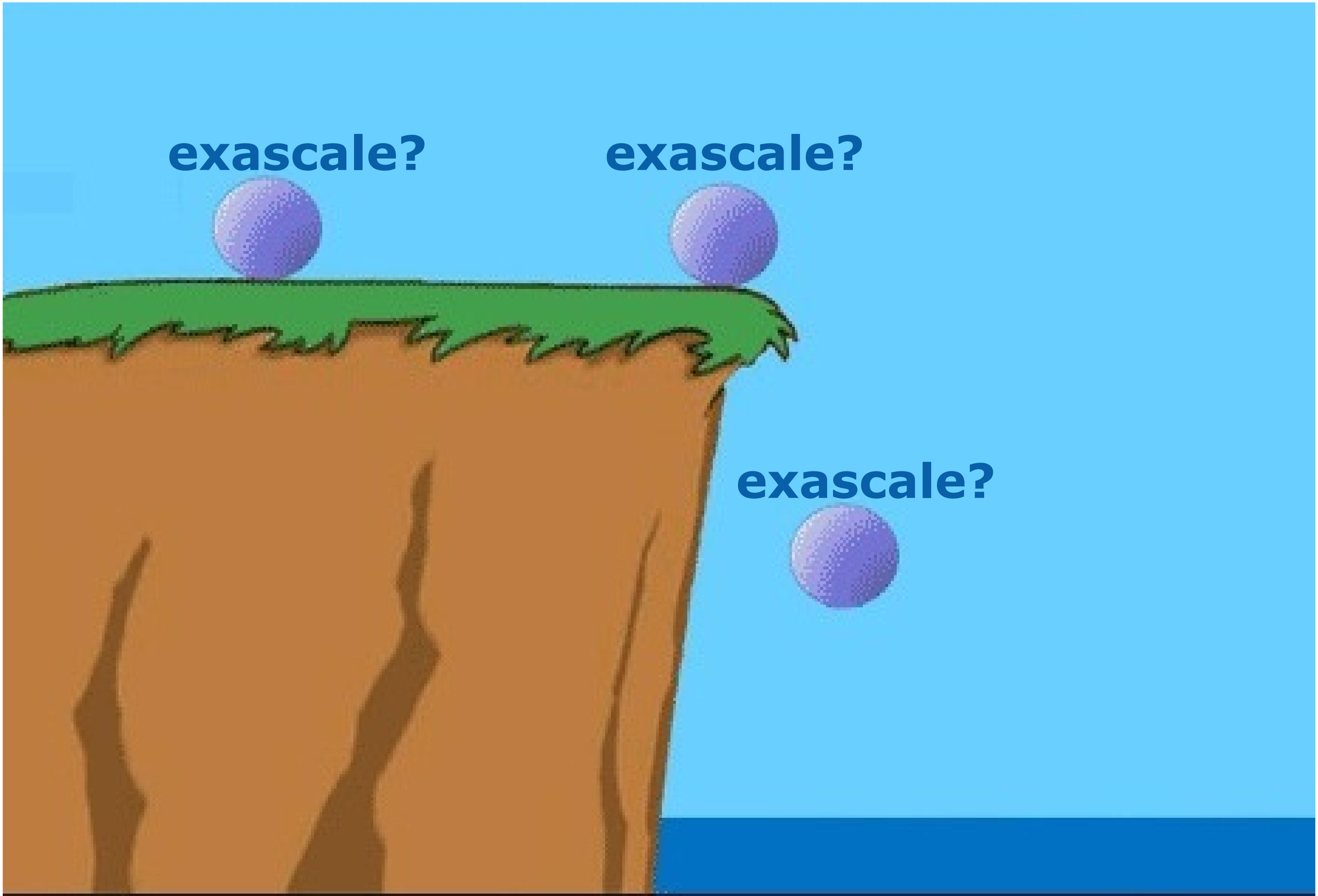
September 10, 2013

Outline

- What is exascale
- Exascale software tension resolved
- Exascale software approaches
- Exascale software components
- Conclusion

What is exascale

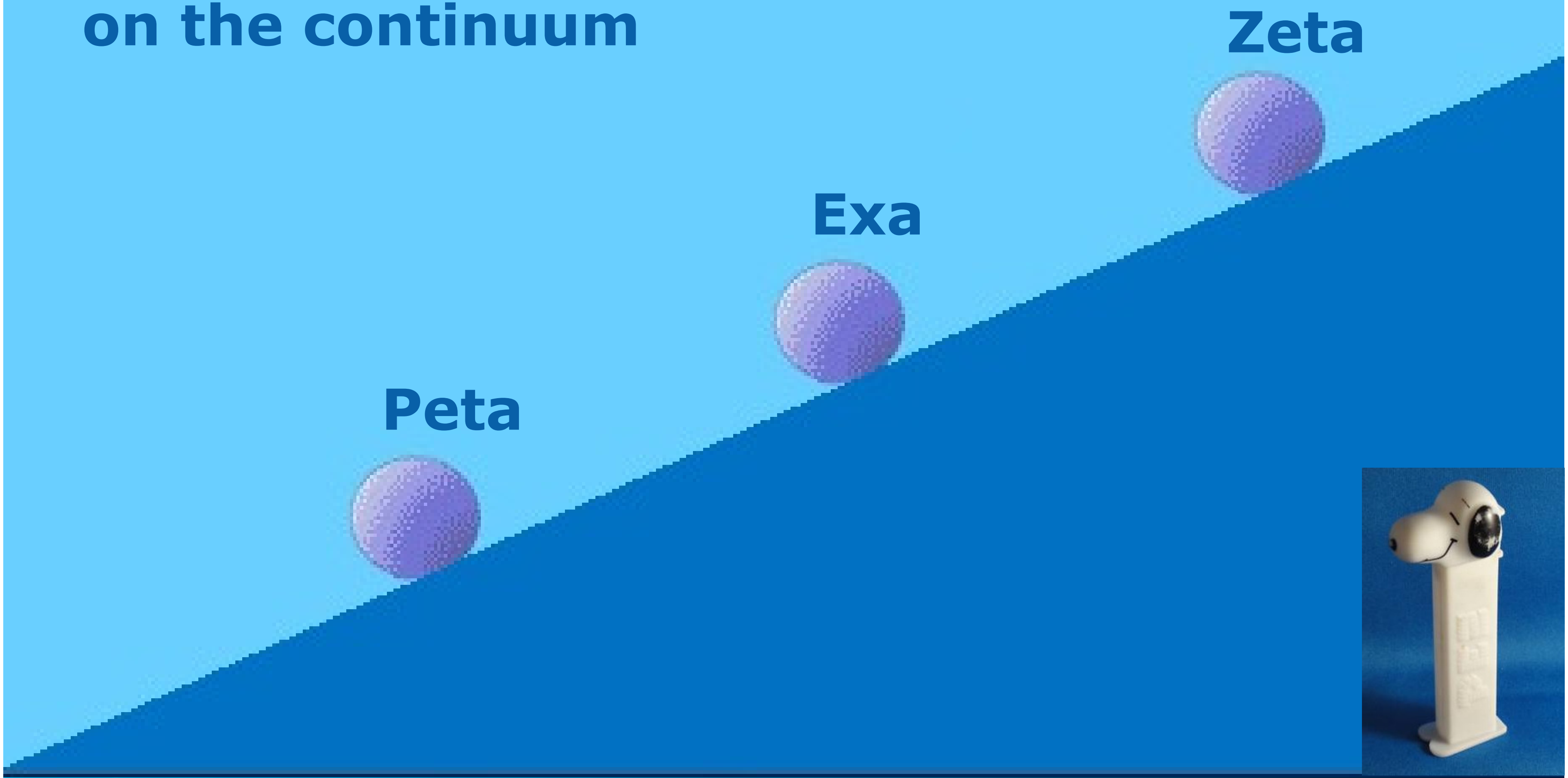
?


exascale?
exascale?
exascale?

What is Exascale

- Today
  - Tianhe-2 (Milkyway-2)
    - 54 Peak PF
    - 125 racks, 17.8 MW, 48K Phis, 3.1M cores
  - K computer 10PF 800 racks, Sequoia 20 PF 100 racks
- Exascale
  - 10^18 operations per second
  - Biggest challenges: Power, Scalability, Reliability
    - Approximate straight-line projections yield:
      - 350M Watts
      - 100M computing threads
      - Each OS instance needs to stay up 50,000 years
  - Exascale software approaches need to address challenges

Exascale is only a point on the continuum



Extreme-Scale Software Challenge

?

?

OR

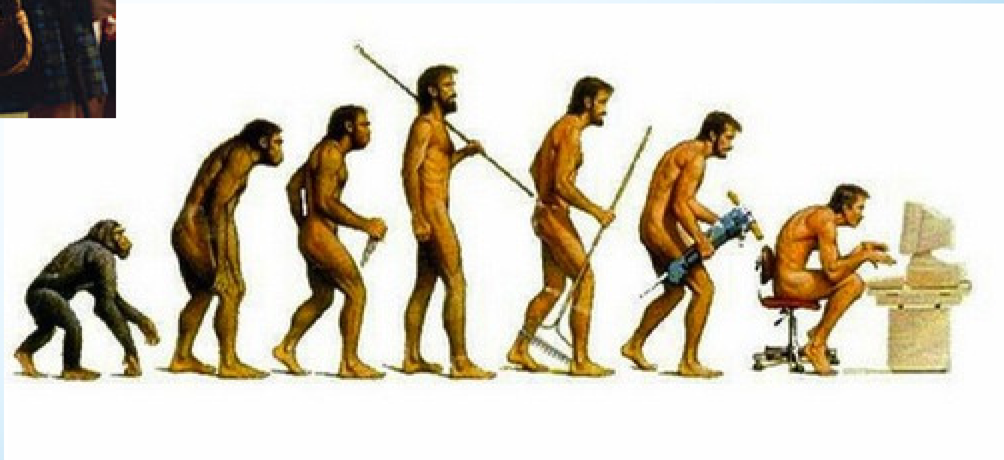
When investigations began

- Challenges too great with current SW
- Need all new OS, compiler, language…

Others advocated

- Enhance capability of existing
- Hard, drive evolutionary approach

Revolutionary versus Evolutionary



- Which one ?

Revolutionary

Imagine vendors telling their customers throw out everything you've done over the last 20+ years

Evolutionary

But there are serious challenges
getting to exascale

The Real Extreme-Scale Software Challenge

- The real challenge in moving software to extreme scale, and therefore the real solution, will be figuring out how to incorporate and support existing computation paradigms in an evolutionary model while **simultaneously** supporting new revolutionary paradigms.

AND

Moving to Extreme Scale

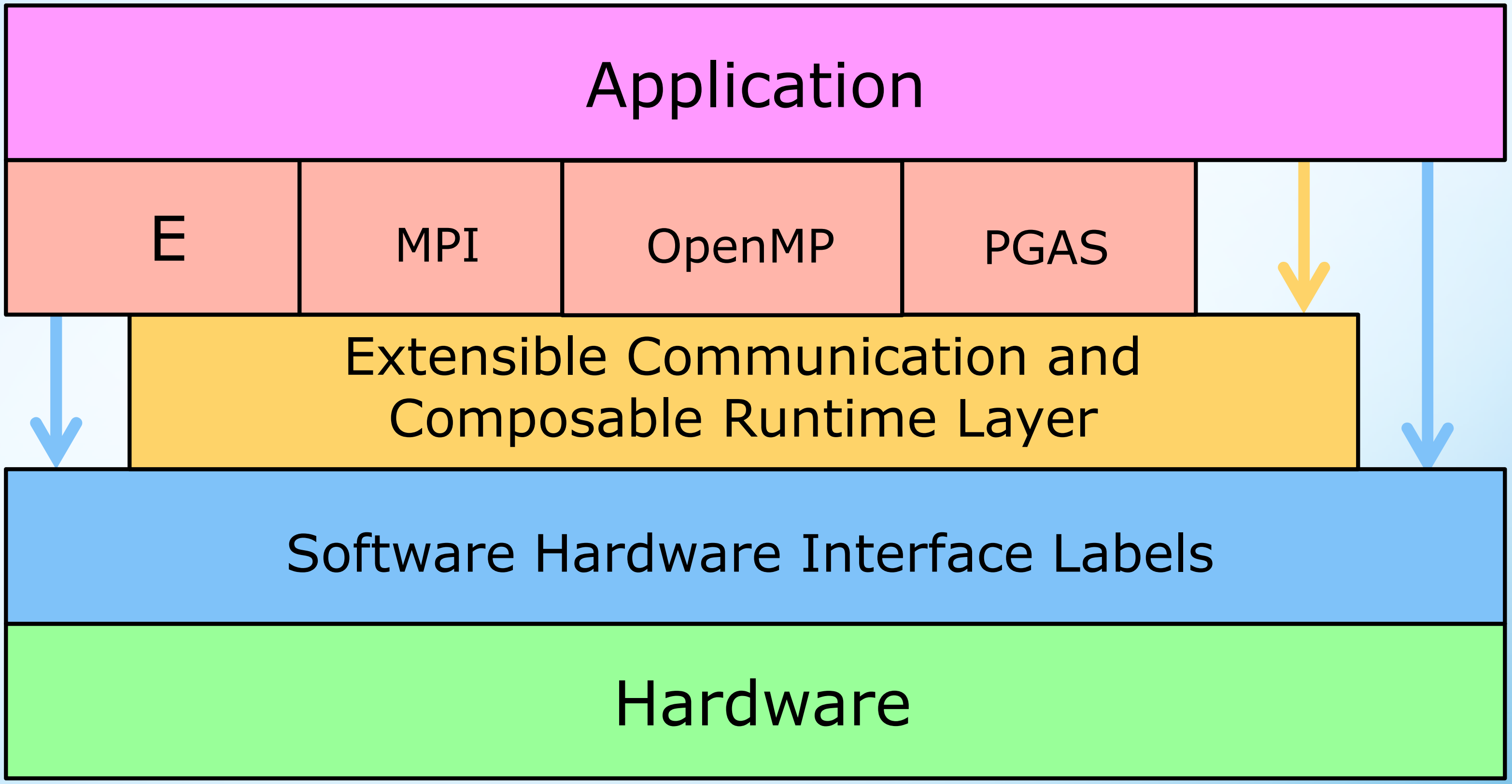
- Support evolutionary and revolutionary models

- Scale

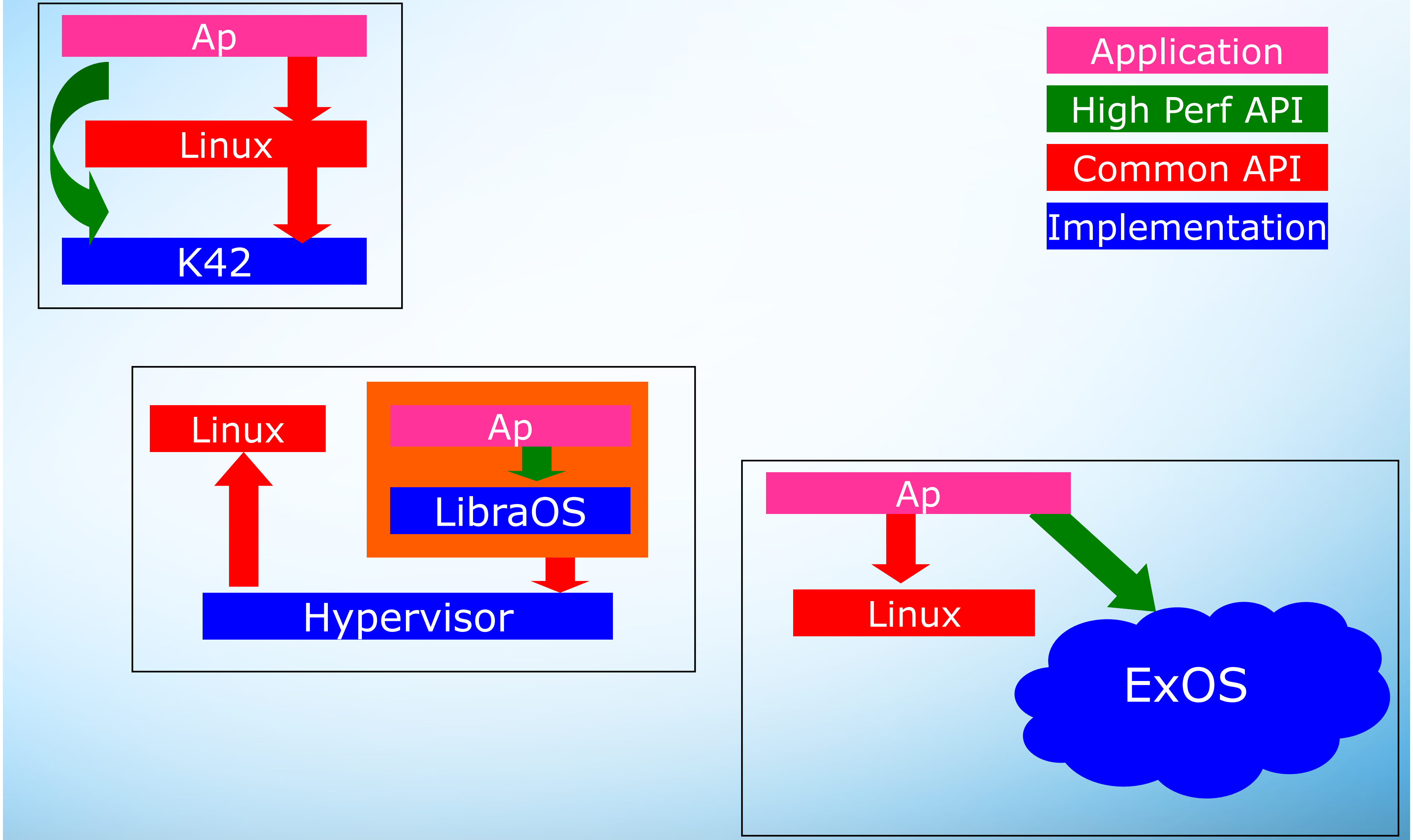
- Be resilient

- Be power aware

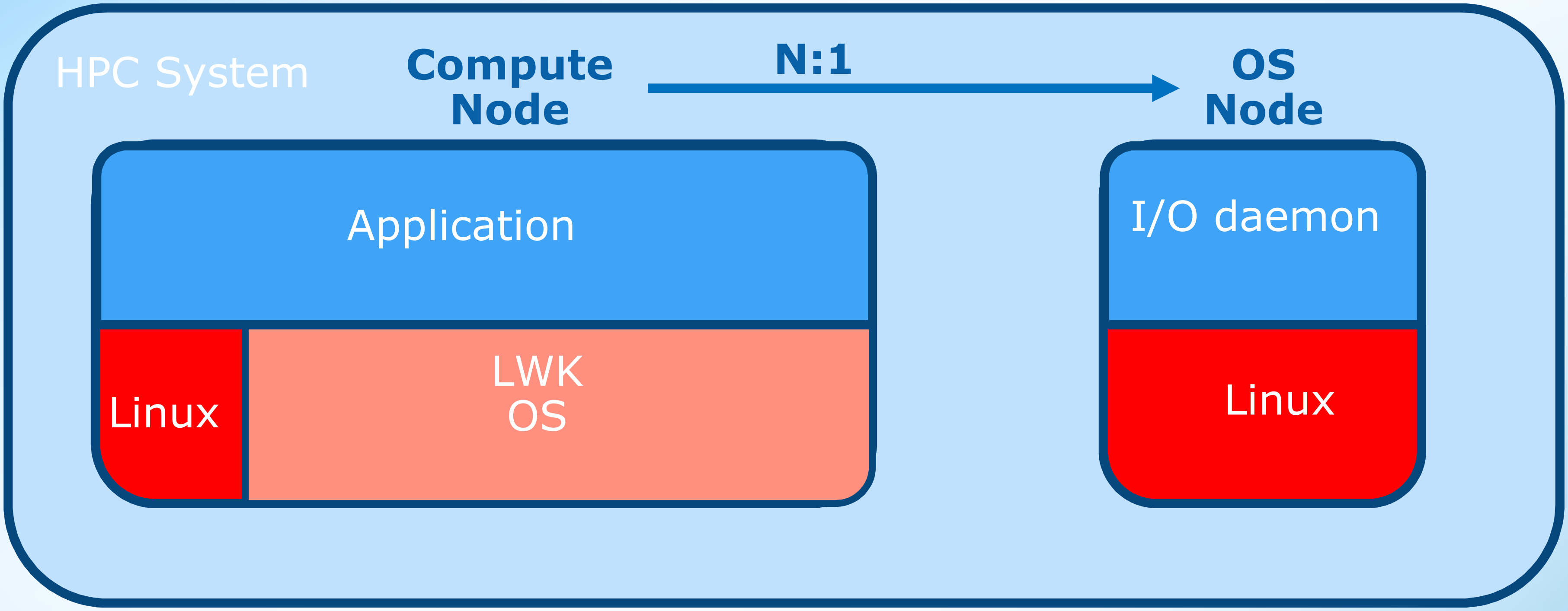
Communication Example


Application
E
MPI
OpenMP
PGAS
Extensible Communication and Composable Runtime Layer
Software Hardware Interface Labels
Hardware

Operating System Example



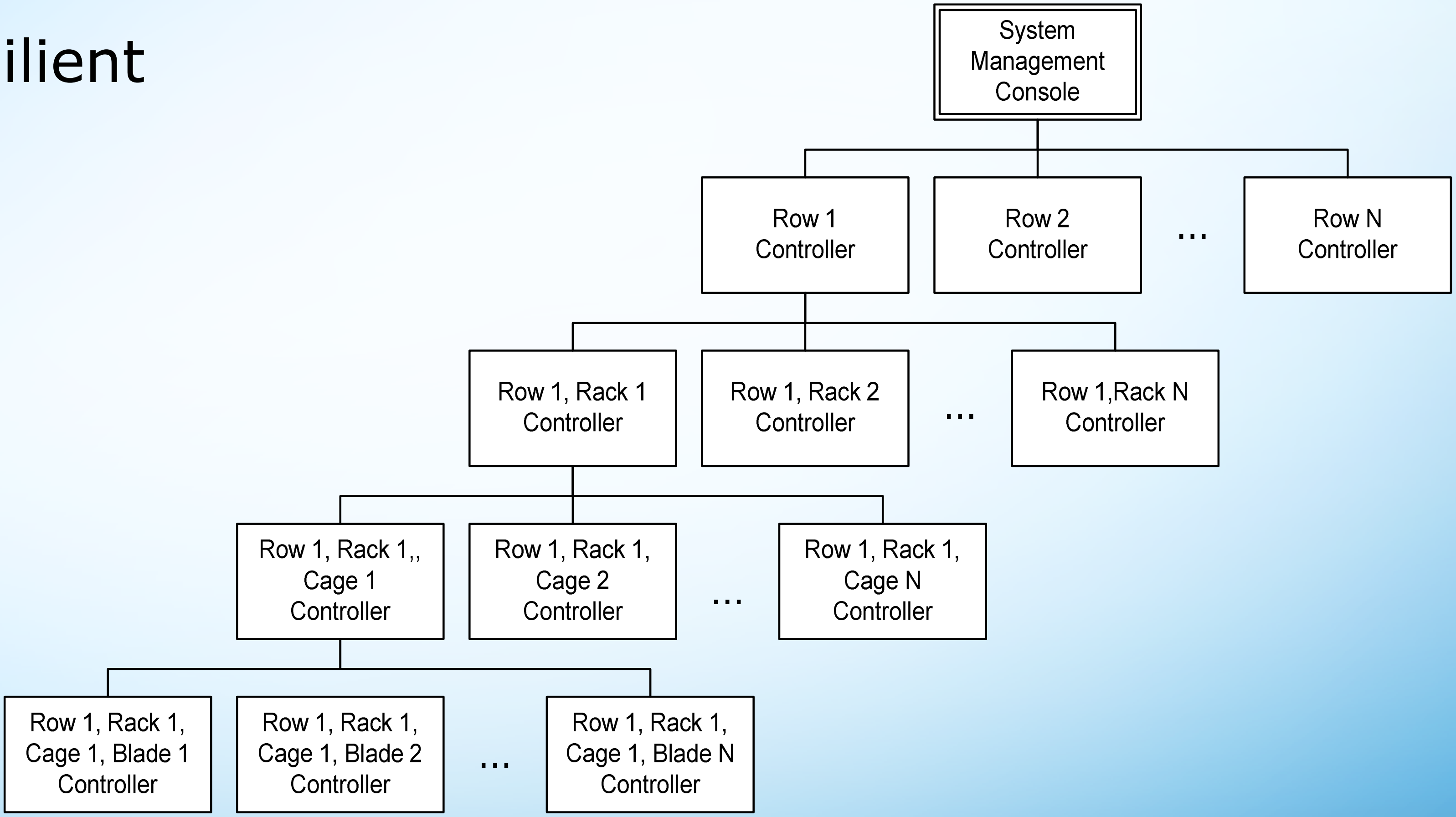
OS Compute Node View



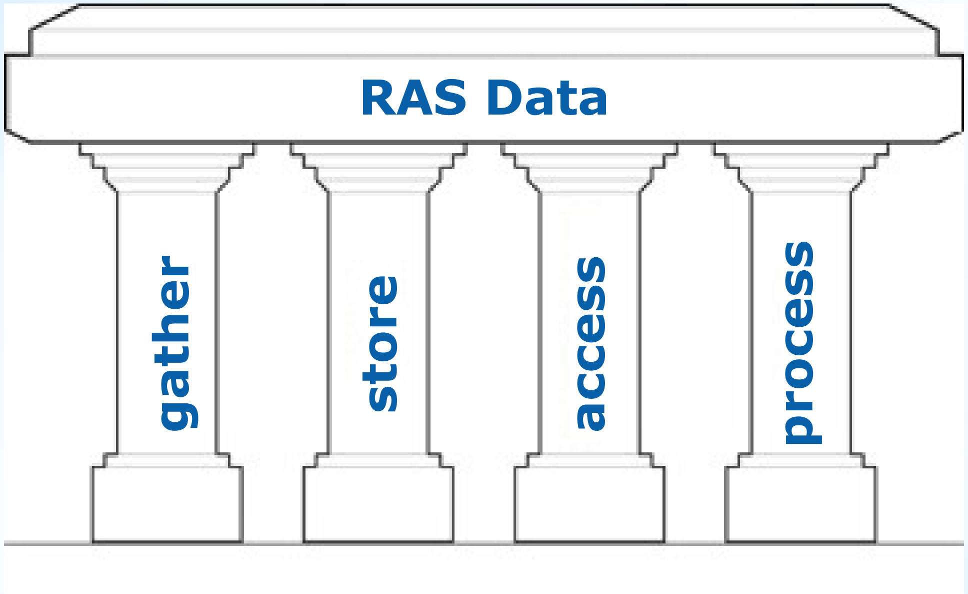
- CNOS that fully supports Linux API and ABI
- Nimble to support new technology effectively
- Move to hierarchy of OS offload for scalability
- Support fine-grained threading and asynchronous requests
- Provide support for and be amenable to running on differentiated cores

System Management

- Provide single comprehensive view of system
- Hierarchical and scalable
- Resilient


System Management Console
Row 1 Controller
Row 2 Controller
…
Row N Controller
Row 1, Rack 1 Controller
Row 1, Rack 2 Controller
…
Row 1,Rack N Controller
Row 1, Rack 1,, Cage 1 Controller
Row 1, Rack 1, Cage 2 Controller
…
Row 1, Rack 1, Cage N Controller
Row 1, Rack 1, Cage 1, Blade 1 Controller
Row 1, Rack 1, Cage 1, Blade 2 Controller
…
Row 1, Rack 1, Cage 1, Blade N Controller

Scalable RAS Infrastructure



- Four Pillars of RAS
  - Gather: As extensive as possible, consistent format
  - Store: Database for searching and associating
  - Access: Real-time pub-sub access by all components
  - Process: Agents aggregate, trigger, notify, filter, etc.

Conclusion

- We will get to extreme scale by figuring out how to incorporate existing computation paradigms in an evolutionary model while **simultaneously** supporting new revolutionary paradigms
  - Support evolutionary and revolutionary models
  - Scale
  - Be resilient
  - Be power aware

AND