HPC System Software Vision for Exascale Computing and Beyond

Charm++

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

April 29, 2014

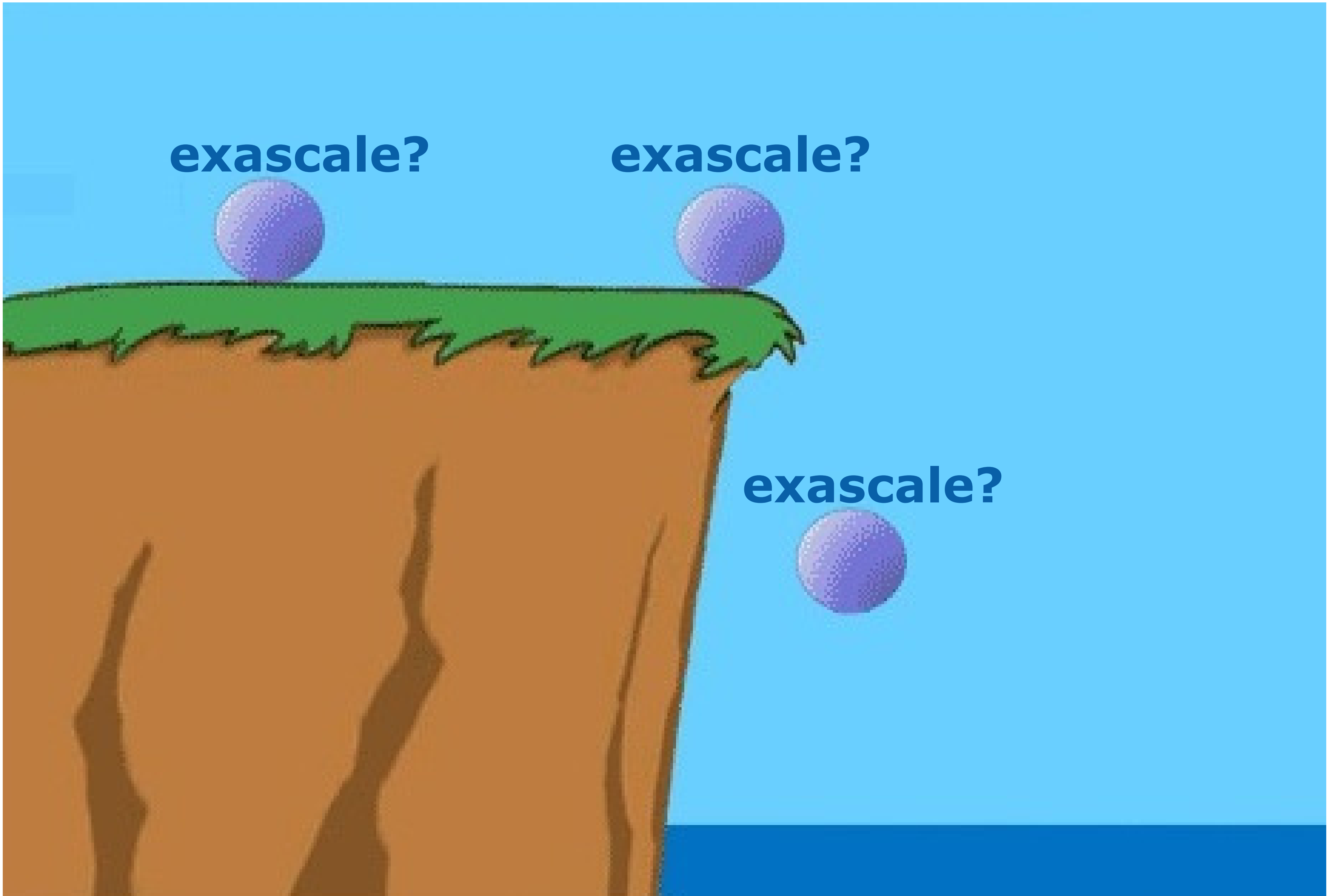
Legal Disclaimer

Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Outline

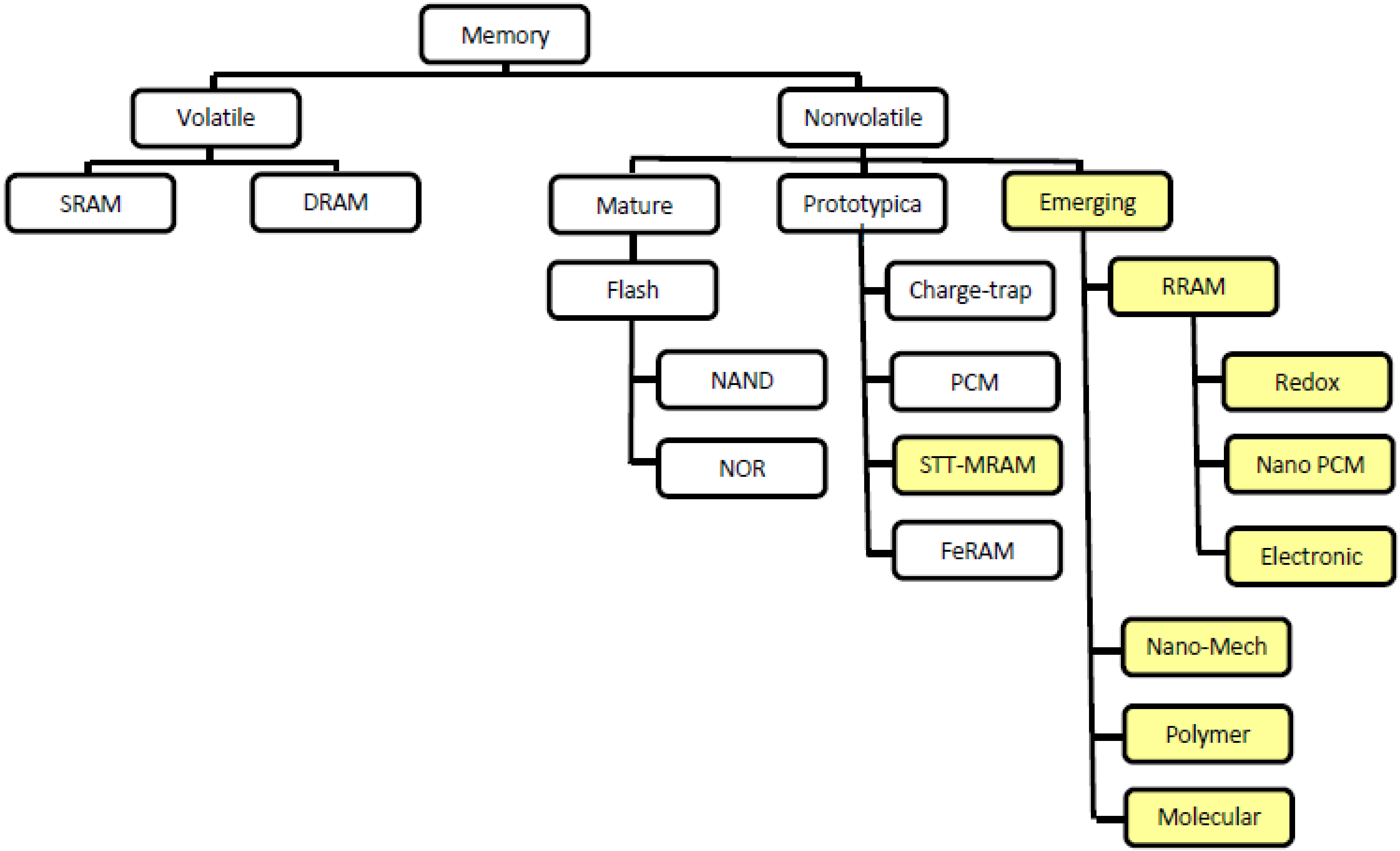
- What is exascale
- Relevant hardware trends
  - Change in model for HPC
  - New opportunity in big data though
- PEZ: not exascale: extreme-scale system software approaches
- Software components
- Conclusion


exascale?
exascale?
exascale?

What is Exascale

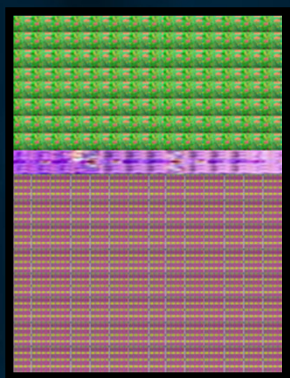
- Today
  - Tianhe-2 (Milkyway-2)
    - 54 Peak PF
    - 125 racks, 17.8 MW, 48K Phis, 3.1M cores
  - K computer 10PF 800 racks, Sequoia 20 PF 100 racks
- Exascale
  - 10^18 operations per second
  - Biggest challenges: Power, Scalability, Reliability
    - Approximate straight-line projections yield:
      - 350M Watts
      - 100M computing threads
      - Each OS instance needs to stay up 50,000 years
  - Biggest change: I/O
    - Bits can no longer traverse from spinning disk to registers and back
  - Software approaches need to address these challenges

Taxonomy of Emerging Memory Technologies



Courtesy of James Hutchby SRC

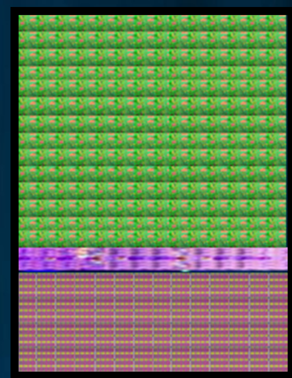
Option 1: Large Die With>10B Transistors



More cache
Fewer cores
"Everything integrated"

✓ Enables on-package memory

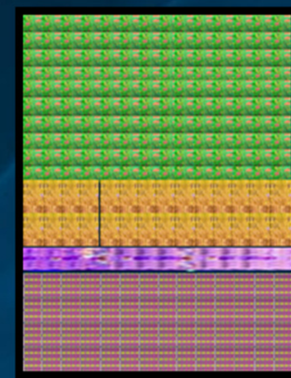
? Cache size beyond a certain threshold not utilized by the programmer"



More cores
Enough cache for HPC
"Everything integrated"

✓ High FLOPS count on a die

? Enough on-package memory becomes difficult to implement
Extreme performance levels result in problematic off-package memory usage

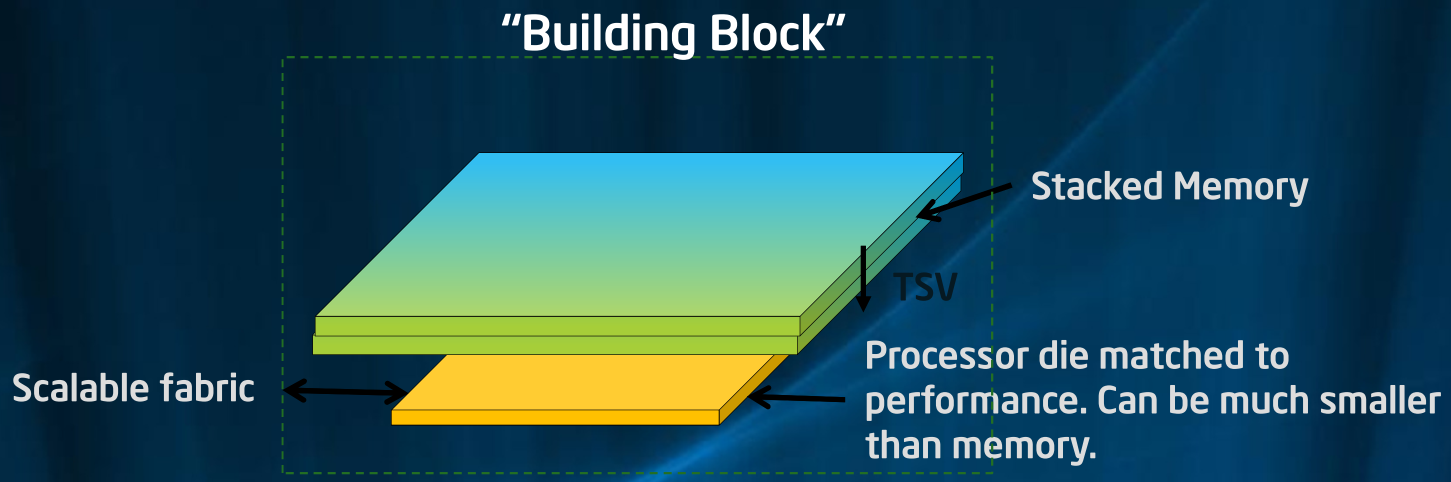


Flavor of cores
Enough cache for HPC
"Everything integrated"

✓ "Powerful" cores for ST performance
"Smaller" cores for highly parallel

? Enough on-package memory becomes difficult to implement.
Extreme performance levels result in problematic off-package memory usage

Option 2: Cost Effective Die That Supports On-package Memory



- Broad Usage: With the right memory capacity per building block, it can address a large portion of the HPC market
- Cost: Building blocks can replace the compute and DRAM in a node *(at the right price point)*
- Scalability: Configure building block as memory or memory+compute
- Power: Better thermal solution with disaggregated compute blocks

The Possibilities
With the "Building Block" Approach

	At Exascale	Evolved
Cost	1	1
Memory capacity (in-package)	2 TB	300 GB
Memory capacity (outside package)	Assume none	2TB (DDR4/5)
Number of cores	8000	1000
Memory Bandwidth (In-package)	50 TB/s	5 TB/s
Memory Bandwidth (outside-package)	Assume none	400 GB/s
Performance peak	512TF	64TF

=16M cores/
64M threads

Synthetic data for illustration only

1) On-package memory has 8-10x the bandwidth compared to external memory

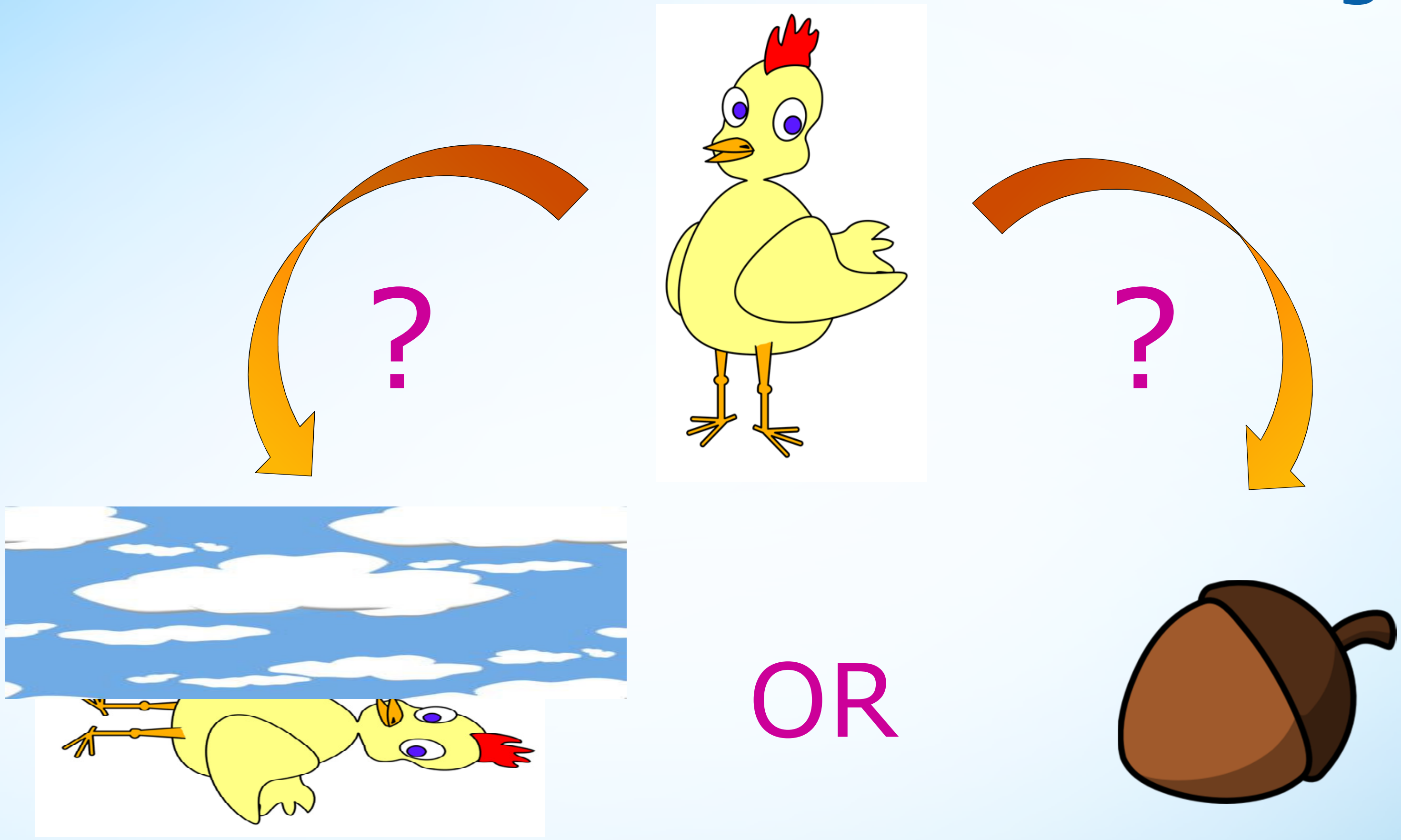
2) At iso cost and memory capacity, on-package memory enables 8-10x additional compute to be placed under the memory

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
  - Biggest change: I/O
    - Bits can no longer traverse from spinning disk to registers and back
  - Software approaches need to address these challenges

Exascale is only a point on the continuum

Extreme-Scale Software Challenge



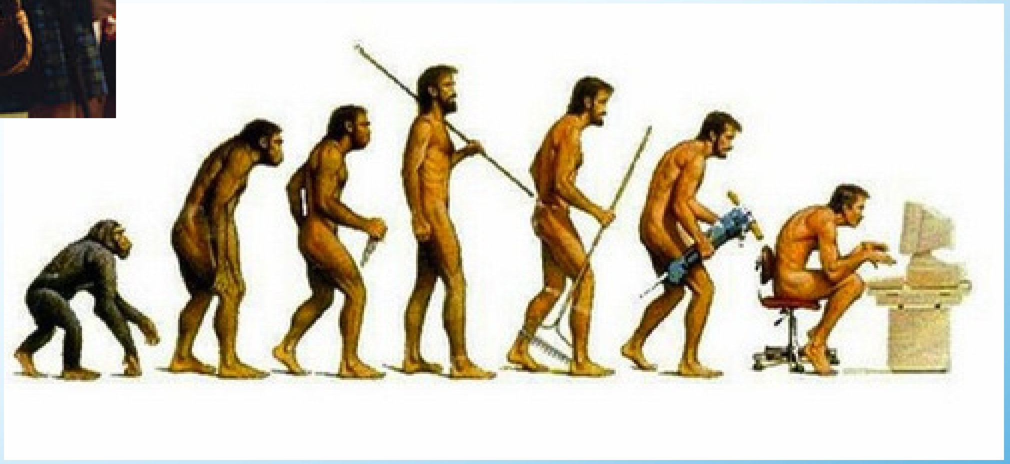
When investigations began

- 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. Leverage tremendous investment in Intel Architecture ecosystem.

Evolutionary

But there are serious challenges getting to exascale. Drive new innovations and invigorate the x86 ecosystem.

The Real Extreme-Scale Software Challenge

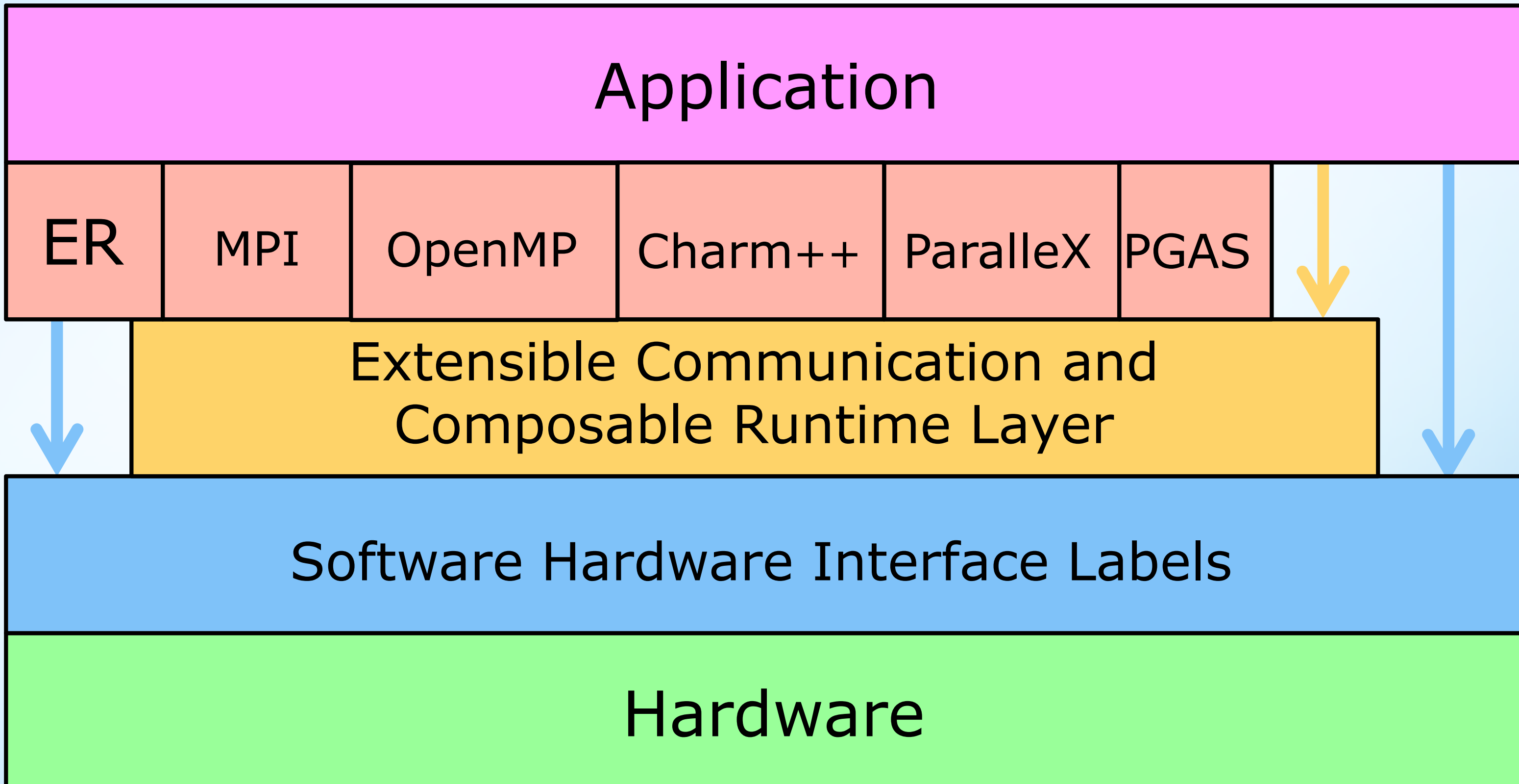
- The real challenge in moving software to extreme scale, and therefore the real solution, will be figuring out how to incorporate and support existing computation paradigms in an evolutionary model while **simultaneously** supporting new revolutionary paradigms.

AND

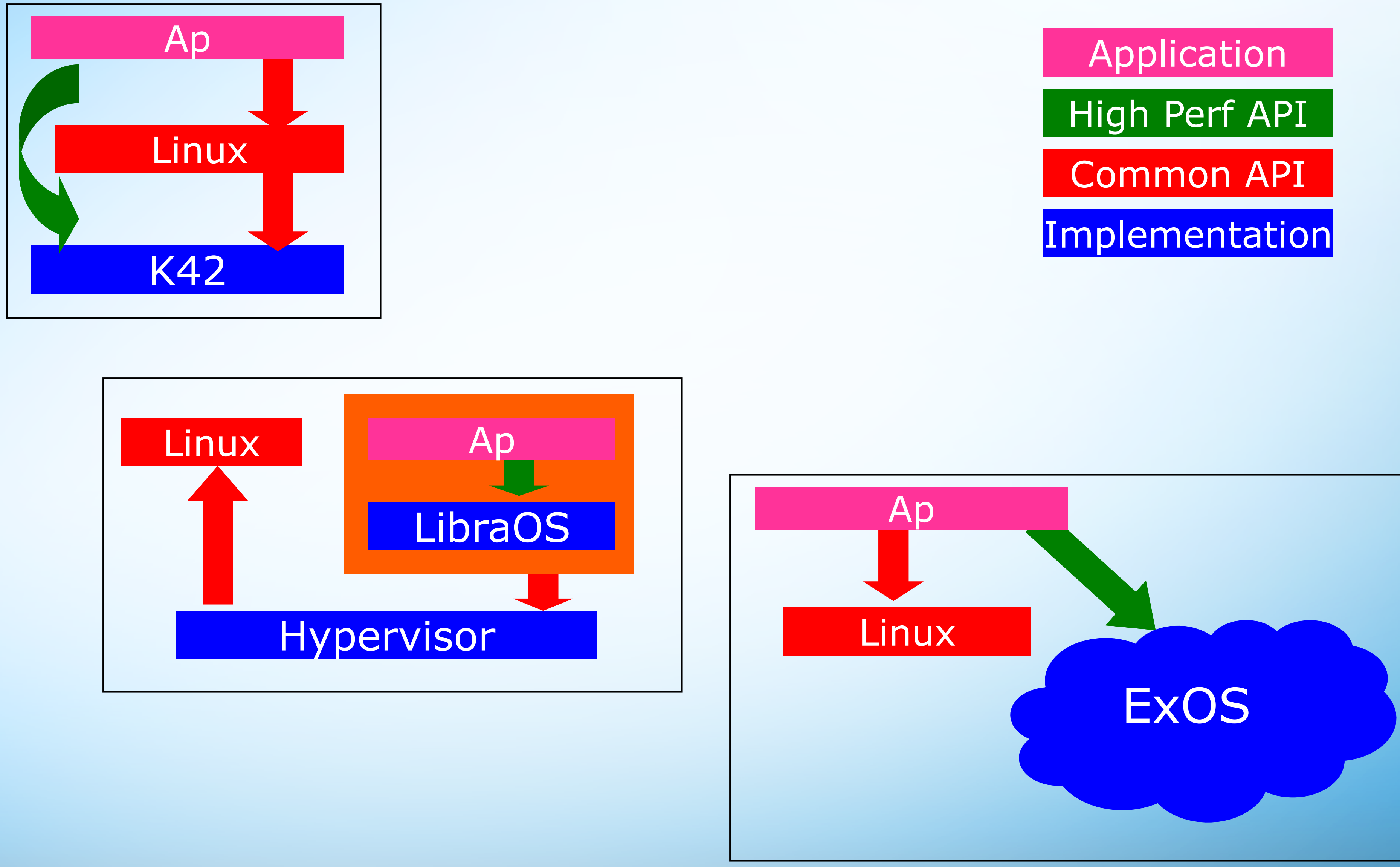
Moving to Extreme Scale

- Support evolutionary and revolutionary models
- Scale
- Be resilient
- Be power aware

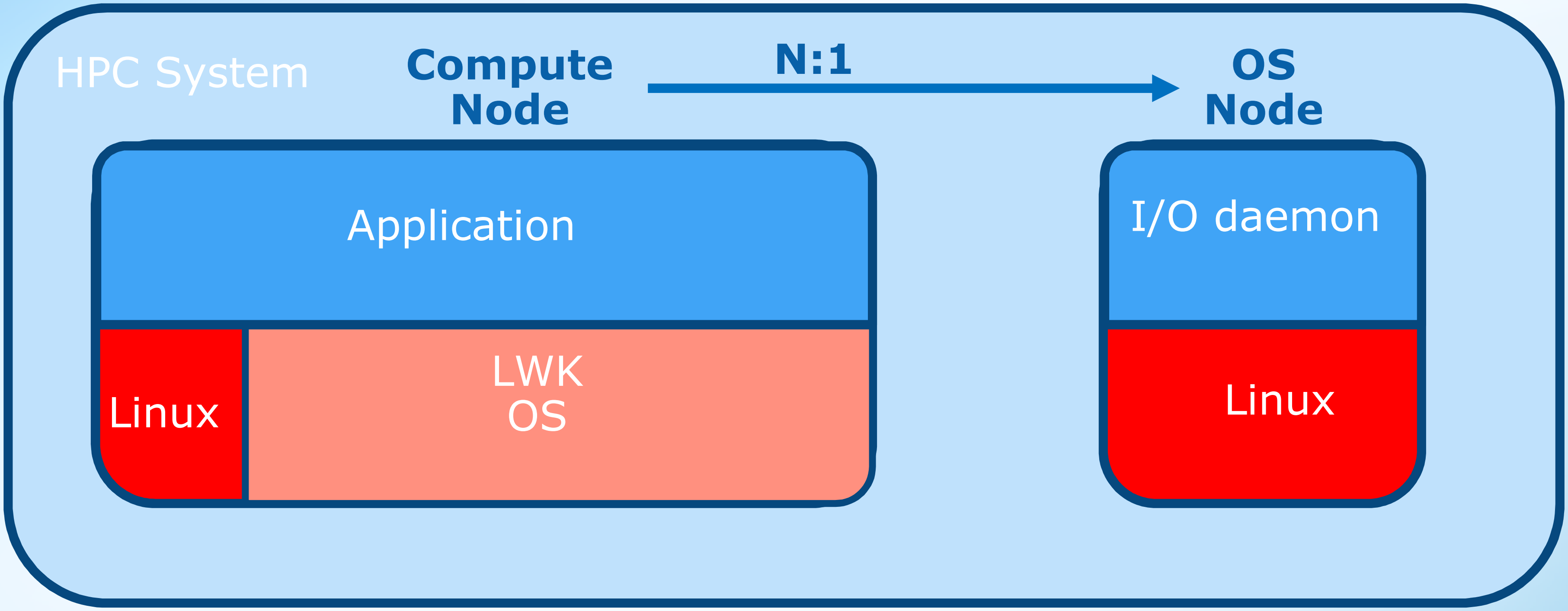
Communication Example



Operating System Example

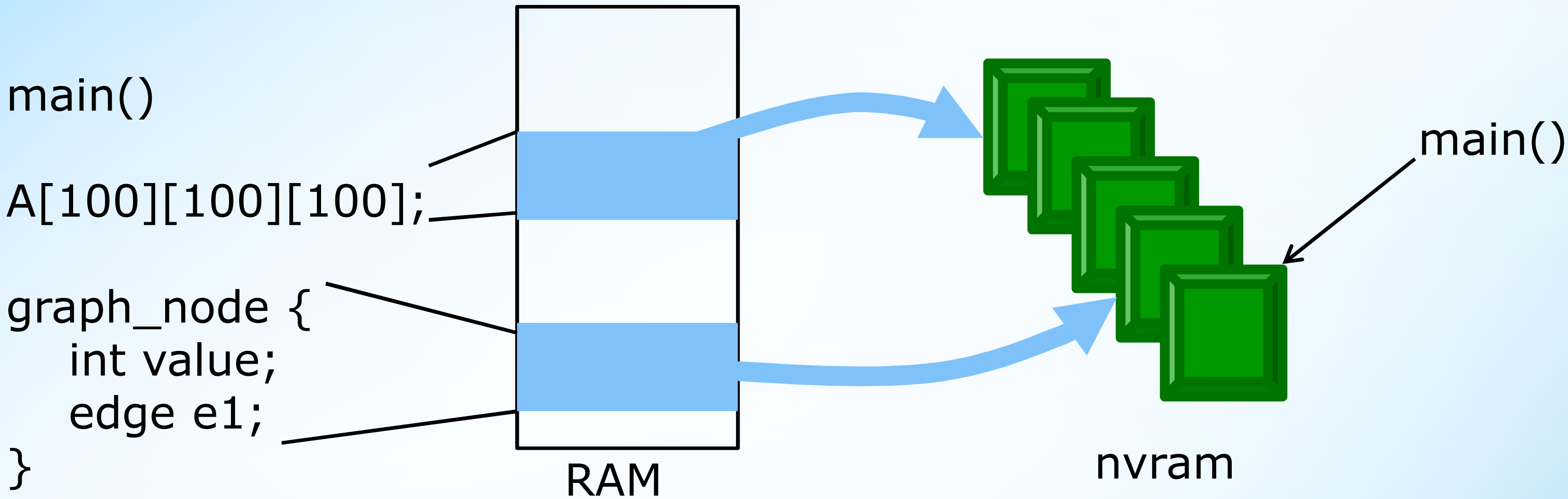


OS Compute Node View



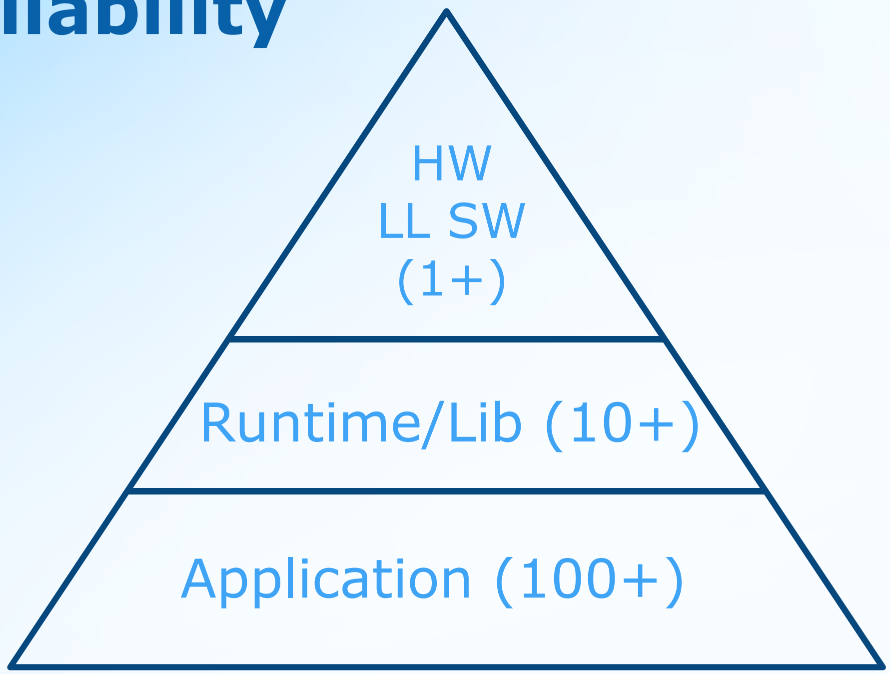
- CNOS that fully supports Linux API and ABI
- Nimble to support new technology effectively
- Move to hierarchy of OS offload for scalability
- Support fine-grained threading and asynchronous requests
- Provide support for and be amenable to running on differentiated cores

Data Management for Big Data
(Long-Term View Active Short-Term Work)



- Smooth representation between
  - Application data structure in memory
  - Representation and access to NVRAM
  - External access
  - Storage to disk
- Moving compute to data

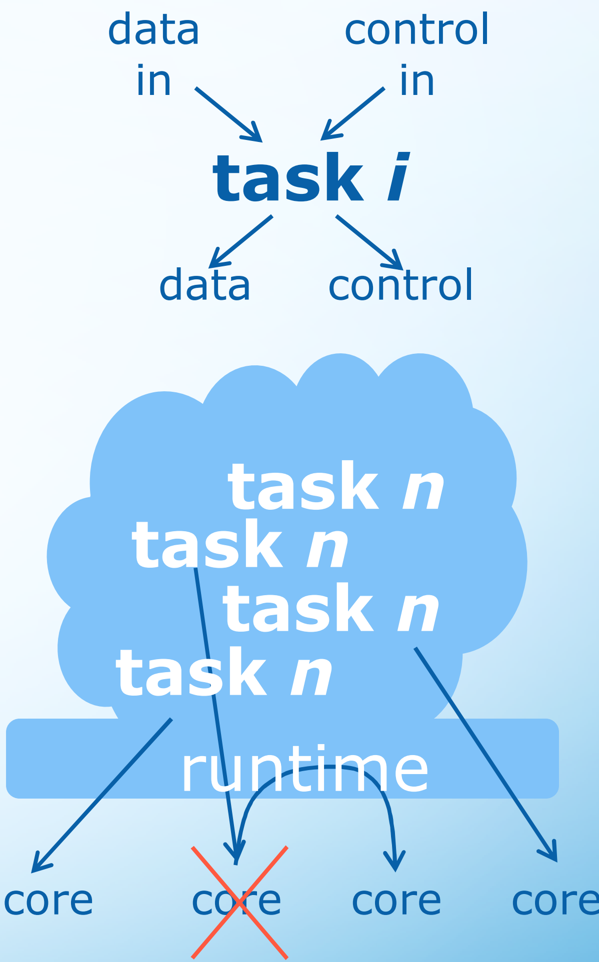
Reliability



Reliability Effort Pyramid

- Check pointing often more efficient if done in user application
- NWChem working on handling errors

Software Example



CnC execution frontiers

Runtime Key Areas for Contribution

- Resilience
- Asynchronous behavior
  - I/O
  - Communication
  - Execution
- Power
- Facilitate fine-grained threading
  - 100s cycles to coordinate

- Understand how a solution
  - Makes a scientific contribution
  - Helps applications across the spectrum
  - Fits into/valuable to current system software

Technical Computing Continues Its Rapid Growth

Governments & Research

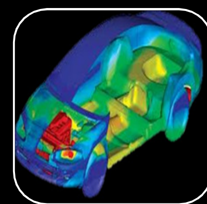
"My goal is simple. It is complete understanding of the universe, why it is as it is and why it exists at all"

Stephen Hawking

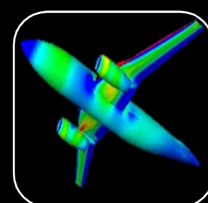
Fundamental Discovery to Gain Fundamental Insights

Commercial/Industrial

Better Products



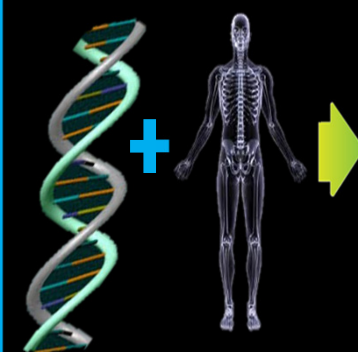
Faster Time to Market



Reduced R&D

Business Transformation

New Users – New Uses



Genomics Clinical Information

From Diagnosis to personal-ized treatment quickly

Big Data Analytics Enabling Data Driven Science

HPC: Transforming the world of data and information into KNOWLEDGE

Source: IDC: Worldwide Technical Computing Server 2013–2017 Forecast; Other brands, names, and images are the property of their respective owners.

Conclusion

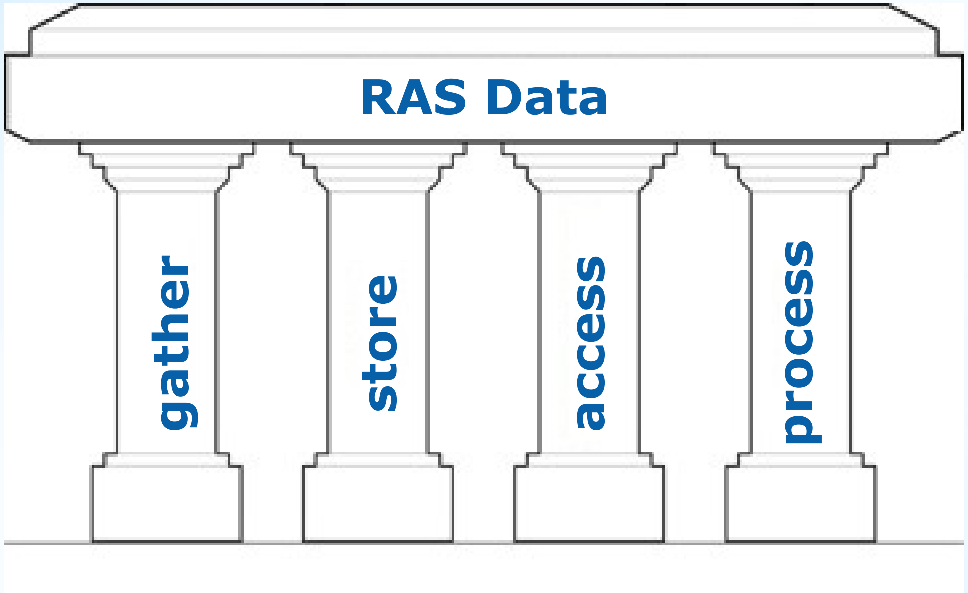
- We will get to extreme scale (PEZ) by figuring out how to incorporate existing computation paradigms in an evolutionary model while **simultaneously** supporting new revolutionary paradigms
  - Support evolutionary and revolutionary models
  - Scale
  - Be resilient
  - Be power aware

AND

Backup

- Backup

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.

System Management

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