IPDPS TCPP meeting, April 2010

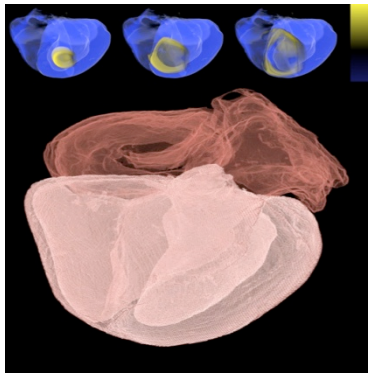
Exascale: Parallelism gone wild!

Craig Stunkel, IBM Research

Outline

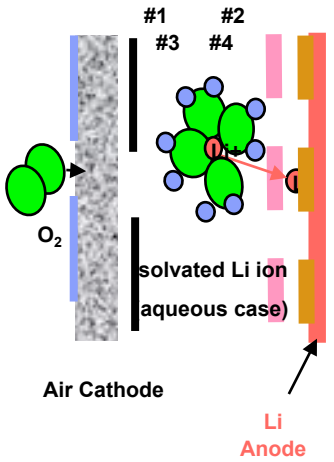
- **Why are we talking about Exascale?**
- **Why will it be fundamentally different?**
- **How will we attack the challenges?**
  - In particular, we will examine:
    - Power
    - Memory
    - Programming models
    - Reliability/Resiliency

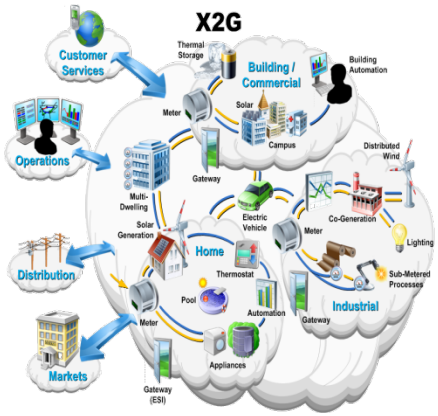
Examples of Applications that Need Exascale



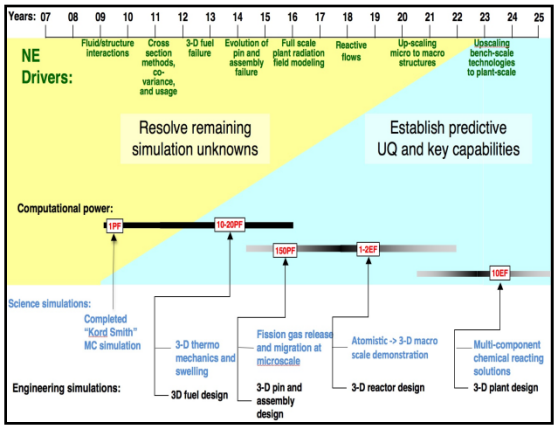
Whole Organ Simulation

Li/Air Batteries



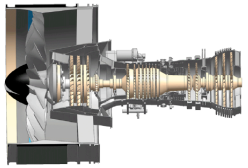


Nuclear Energy



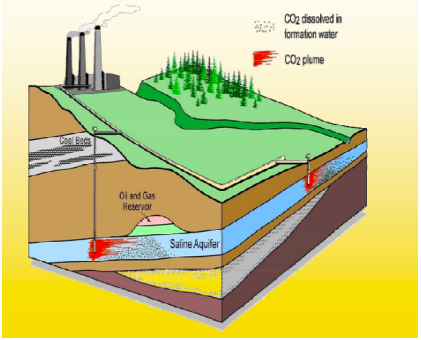
Pratt & Whitney on Intrepid
INCITE PI : Peter Bradley, Pratt & Whitney

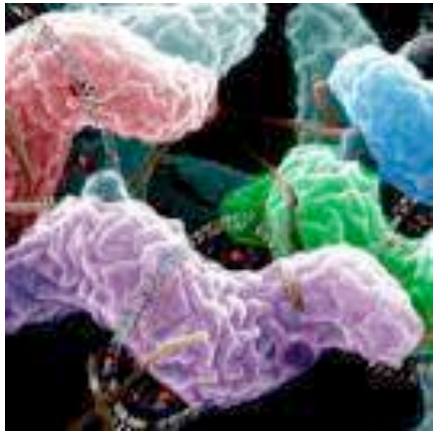
- INCITE 2006-2007 technologies now being applied to next generation low emission engines.
- Important simulations can now be done 3X faster
- A key enabler for the depth of understanding meet emissions goals



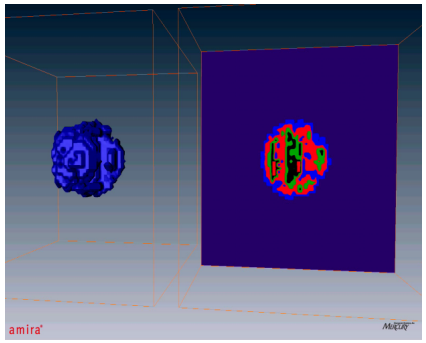
Low Emission Engine Design

CO2 Sequestration





Life Sciences: Sequencing

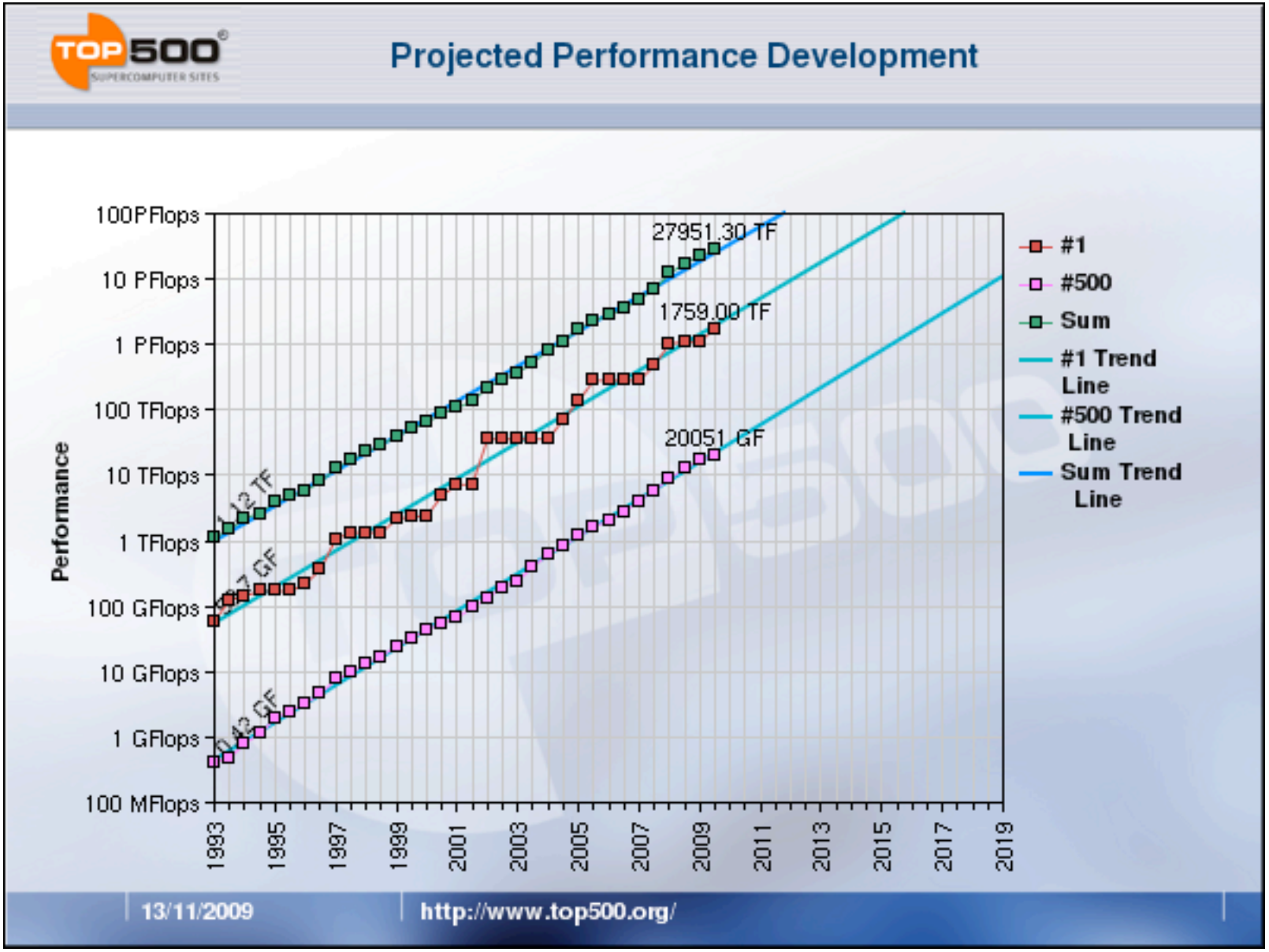


Tumor Modeling

Beyond Petascale, applications will be materially transformed

- Climate: **Improve our understanding of complex biogeochemical cycles that underpin global economic systems functions and control the sustainability of life on Earth**
- Energy: **Develop and optimize new pathways for renewable energy production ….**
- Biology: **Enhance our understanding of the roles and functions of microbial life on Earth and adapt these capabilities for human use …**
- Socioeconomics: **Develop integrated modeling environments for coupling the wealth of observational data and complex models to economic, energy, and resource models that incorporate the human dynamic, enabling large scale global change analysis**

*** "Modeling and simulation at the exascale for energy and the environment", DoE Office of Science Report, 2007.**


TOP500
SUPERCOMPUTER SITES
Projected Performance Development
100 PFlops
10 PFlops
1 PFlops
100 TFlops
10 TFlops
1 TFlops
100 GFlops
10 GFlops
1 GFlops
100 MFlops
Performance
27951.30 TF
1759.00 TF
20051 GF
1.12 TF
59.7 GF
70.43 GF
1993
1995
1997
1999
2001
2003
2005
2007
2009
2011
2013
2015
2017
2019
#1
#500
Sum
#1 Trend Line
#500 Trend Line
Sum Trend Line
13/11/2009
http://www.top500.org/

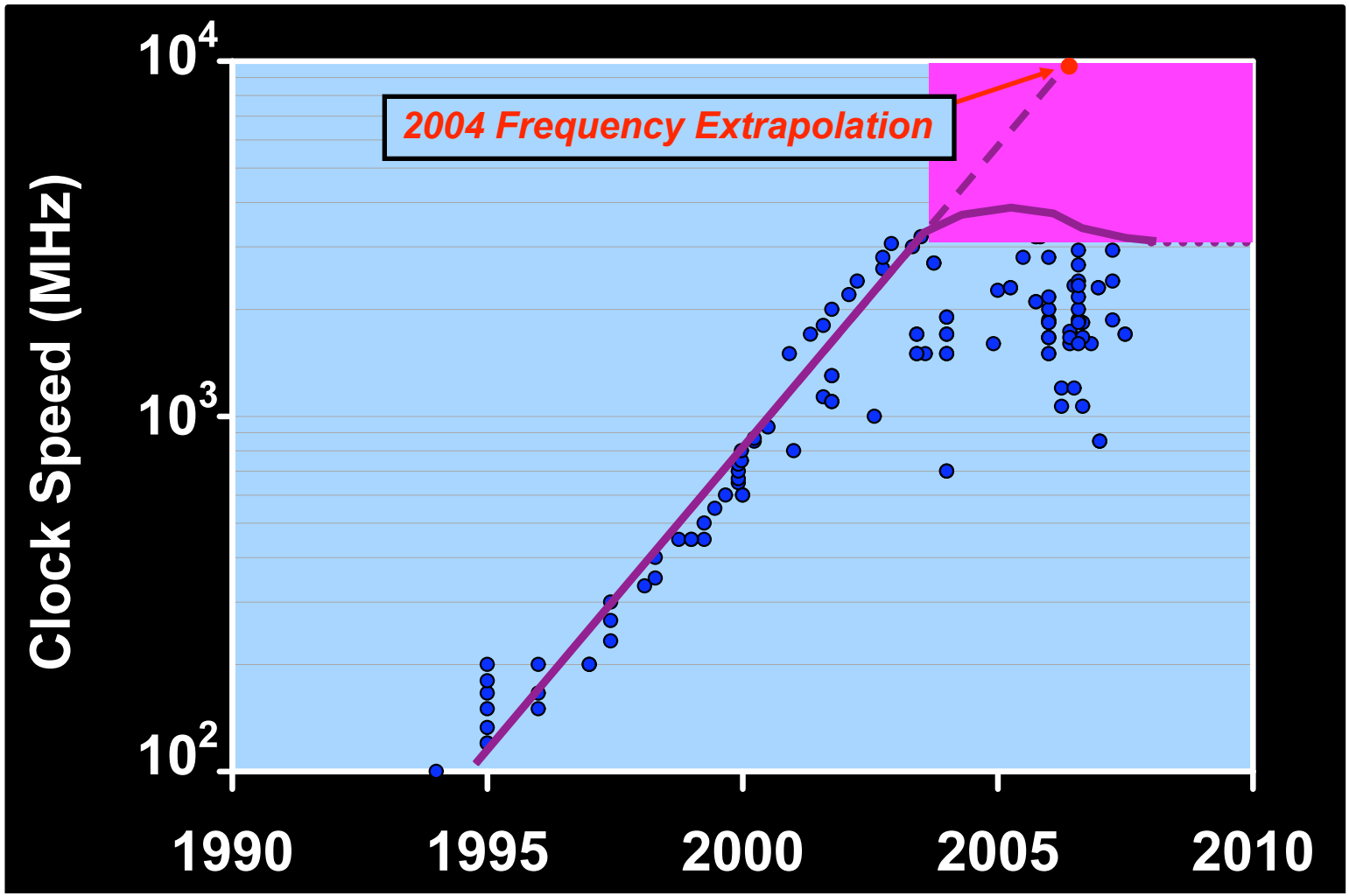
Are we on track to Exascale machines?

- **Some IBM supercomputer sample points:**
- **2008, Los Alamos National Lab: Roadrunner was the first peak Petaflops system**
- **2011, U. of Illinois: Blue Waters will be around 10 Petaflops peak?**
  - NSF “Track 1”, provides a sustained Petaflops system
- **2012, LLNL: Sequoia system, 20 Petaflops peak**

- **So far the Top500 trend (10x every 3.6 years) is continuing**
- **What could possibly go wrong before Exaflops?**

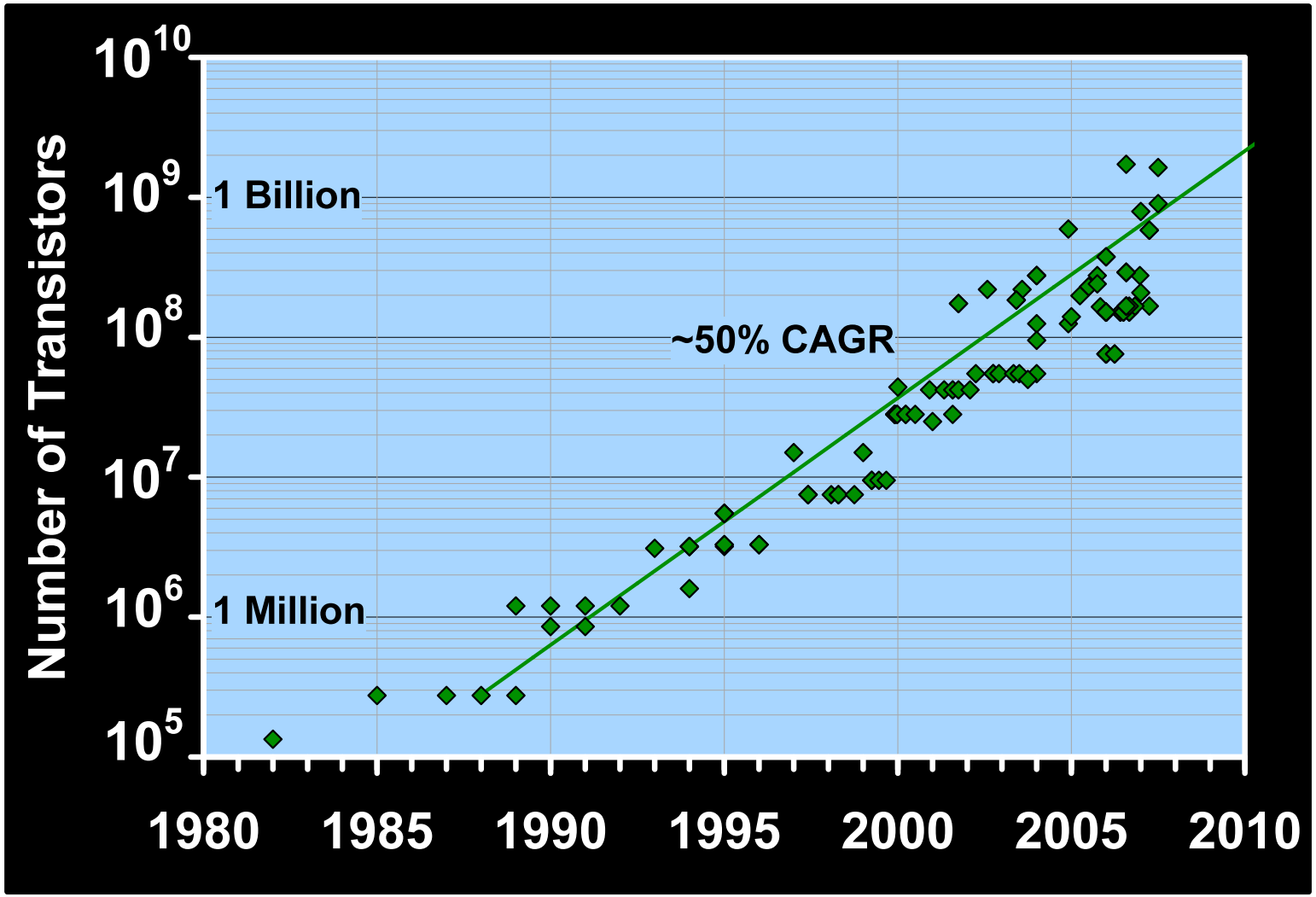
Microprocessor Clock Speed Trends

Managing power dissipation is limiting clock speed increases



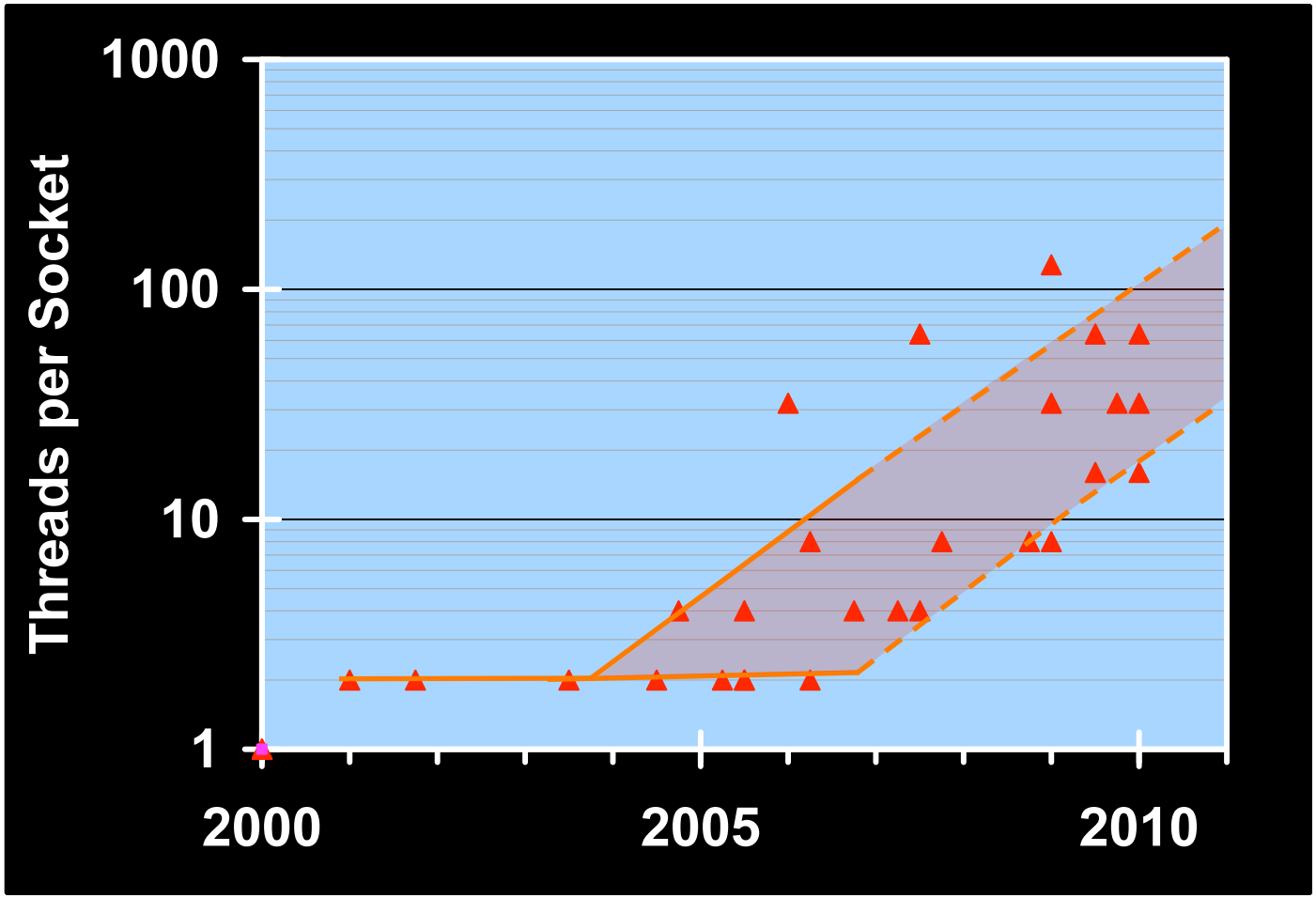
Microprocessor Transistor Trend

Moore's (original) Law alive: transistors still increasing exponentially



Server Microprocessors Thread Growth

We are in a new era of massively multi-threaded computing



Exascale requires much lower power/energy

- **Even for Petascale, energy costs have become a significant portion of TCO**
- **#1 Top500 system consumes 7 MW**
  - 0.25 Gigaflops/Watt
- **For Exascale, 20-25 MW is upper end of comfort**
  - Anything more is a TCO problem for labs
  - And a potential facilities issue

Exascale requires much lower power/energy

- **For Exascale, 20-25 MW is upper end of comfort**
- **For 1 Exaflops, this limits us to 25 pJ/flop**
  - Equivalently, this requires ≥40 Gigaflops/Watt
- **Today's best supercomputer efficiency:**
  - ~ 0.5 – 0.7 Gigaflops/Watt
- **Two orders of magnitude improvement required!**
  - Far more aggressive than commercial roadmaps

A surprising advantage of low power

- **Lower-power processors permit more ops/rack!**
  - Even though more processor chips are required
  - Less variation in heat flux permits more densely packed components
  - Result: more ops/ft^2

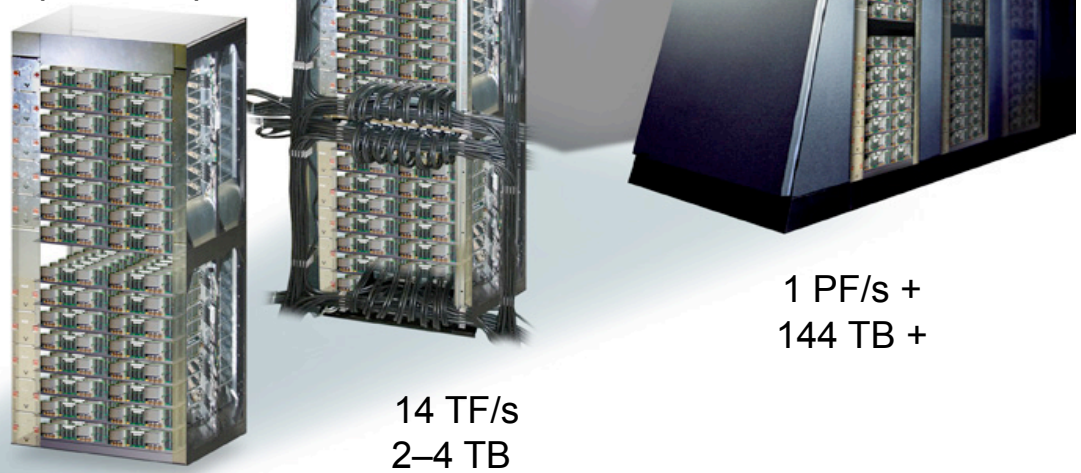
Blue Gene/P

Space-saving, power-efficient packaging

System
1 to 72 or more Racks

Cabled 8x8x16

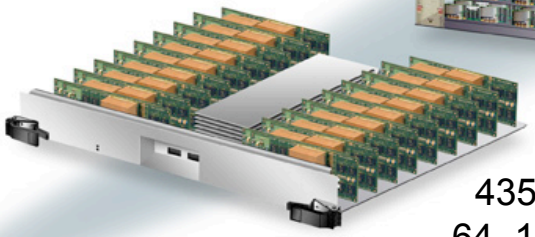
Rack
32 Node Cards
1024 chips, 4096 procs



1 PF/s +
144 TB +

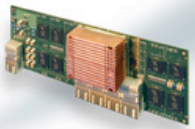
14 TF/s
2–4 TB

Node Card
(32 chips 4x4x2)
32 compute, 0-2 IO cards



435 GF/s
64–128 GB

Compute Card
1 chip, 20 DRAMs



13.6 GF/s
2–4 GB DDR
Supports 4-way SMP

Chip
4 processors

13.6 GF/s
8 MB EDRAM

A perspective on Blue Gene/L

How do we increase power efficiency O(100)?

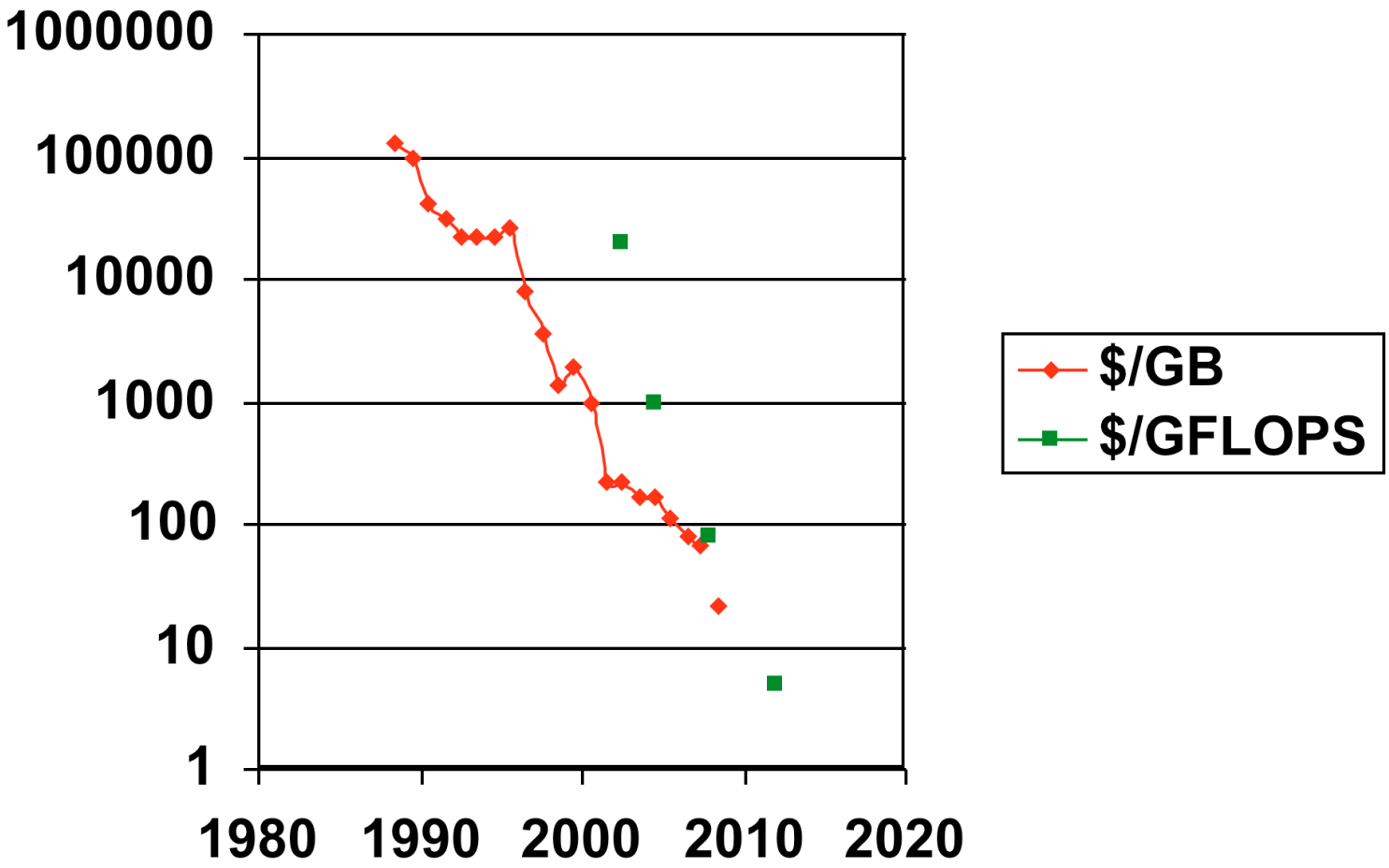
- **Crank down voltage**
- **Smaller devices with each new silicon generation**
- **Run cooler**
- **Circuit innovation**
- **Closer integration (memory, I/O, optics)**

- **But with general-purpose core architectures, we still can't get there**

Core architecture trends that combat power

- **Trend #1: Multi-threaded multi-core processors**
  - Maintain or reduce frequency while replicating cores
- **Trend #2: Wider SIMD units**
- **Trend #3: Special (compute) cores**
  - Power and density advantage for applicable workloads
  - But can't handle all application requirements
- **Result: Heterogeneous multi-core**

Processor versus DRAM costs



Memory costs

- **Memory costs are already a significant portion of system costs**
- **Hypothetical 2018 system decision-making process:**
  - How much memory can I afford?
  - OK, now throw in all the cores you can (for free)

Memory costs: back of the envelope

- **There is (some) limit on the max system cost**
  - This will determine the total amount of DRAM
- **For an Exaflops system, one projection:**
  - Try to maintain historical 1 B/F of DRAM capacity
  - Assume: 8 Gb chips in 2018 @ $1 each
  - ⇨ $1 Billion for DRAM (a bit unlikely ☺)
- **We must live with less DRAM per core unless and until DRAM alternatives become reality**

Getting to Exascale: parallelism gone wild!

- **1 Exaflops is 10^9 Gigaflops**
- **For 3 GHz operation (perhaps optimistic)**
  - ⇨167 Million FP units!
- **Implemented via a heterogeneous multi-threaded multi-core system**
- **Imagine cores with beefy SIMD units containing 8 FPUs**
  - This still requires over 20 Million cores

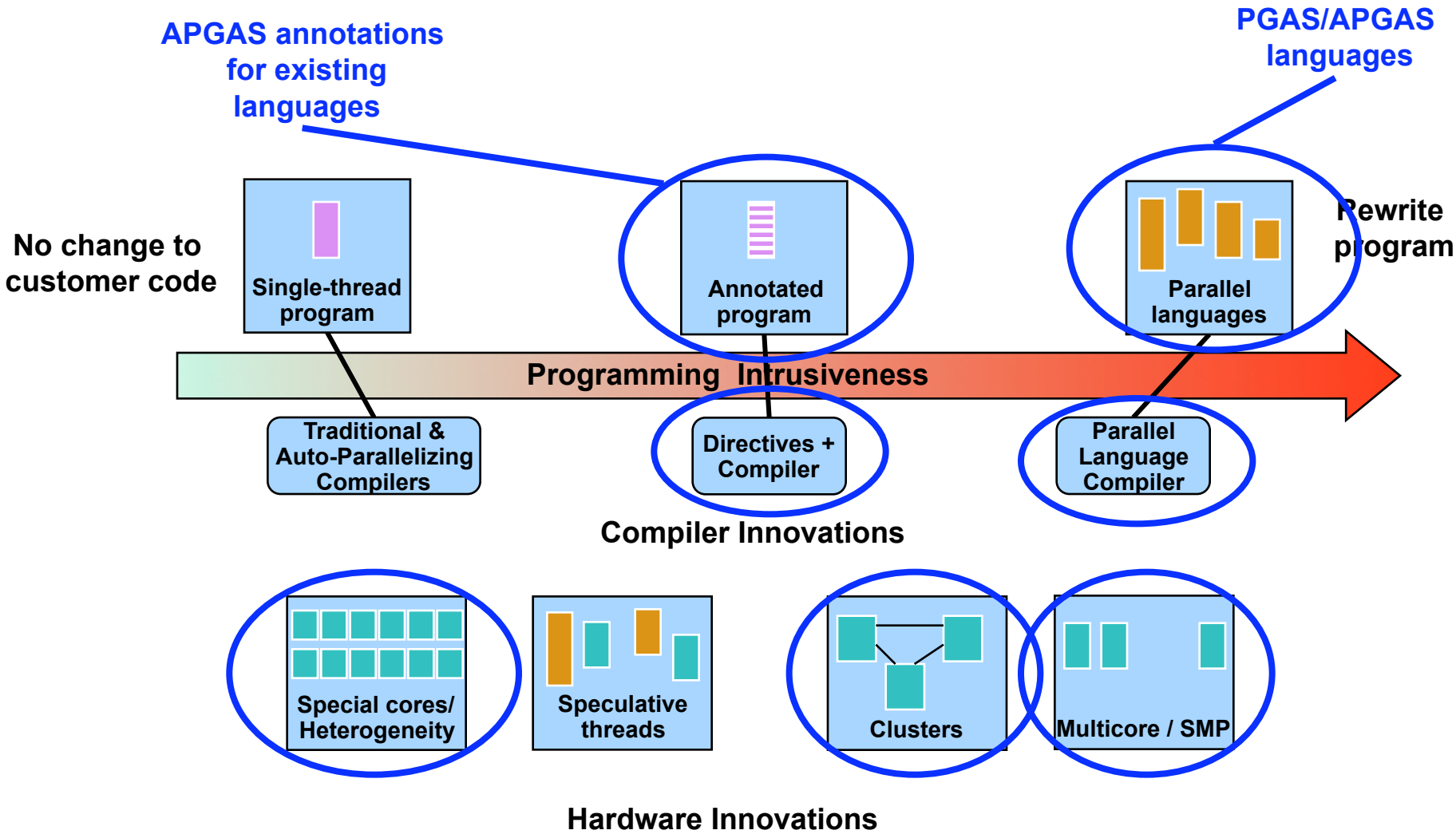
Petascale

Exascale

Programming issues

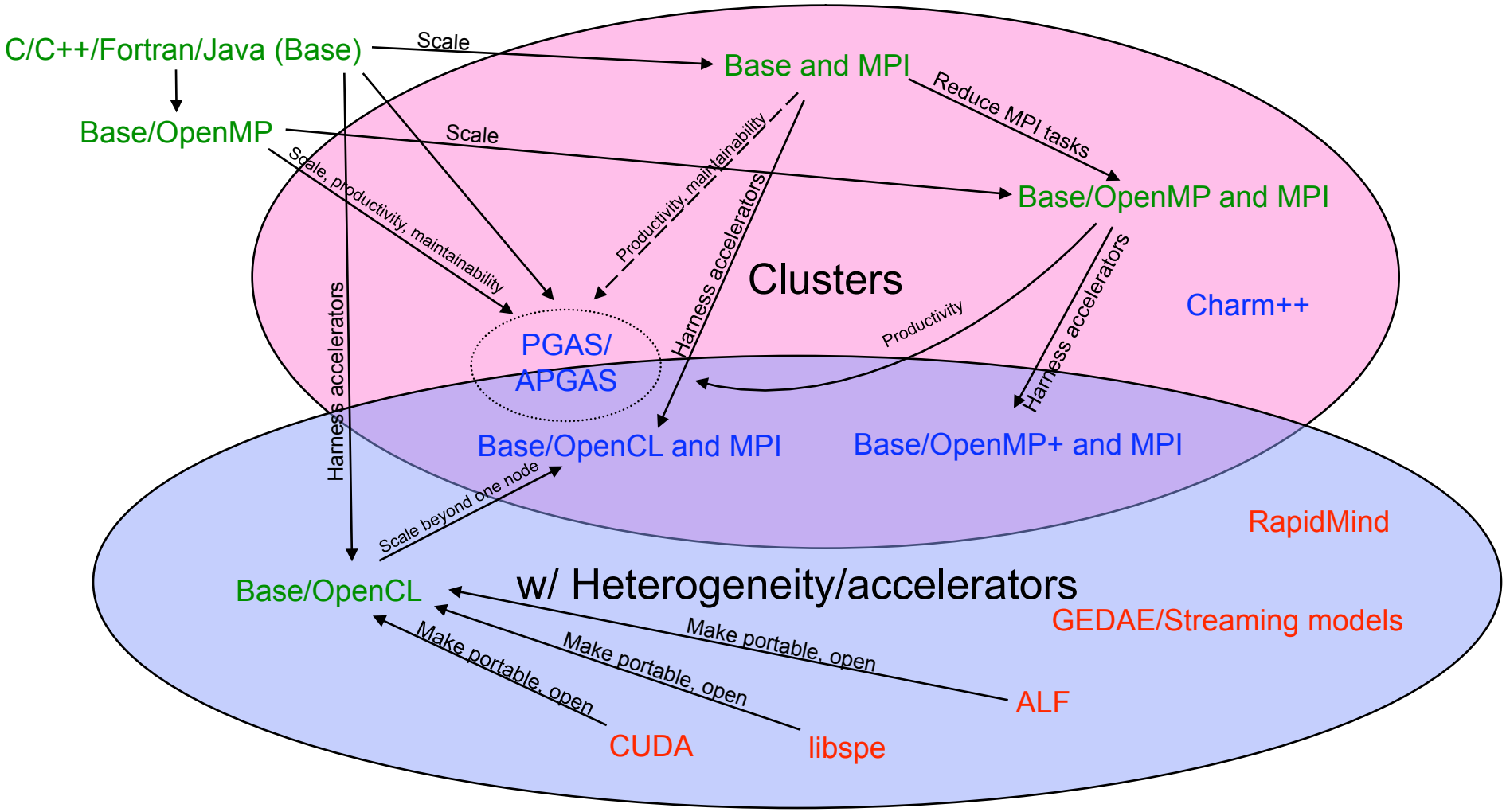
- **Many cores per node**
  - Hybrid programming models to exploit node shared memory?
    - E.g., OpenMP on node, MPI between
  - New models?
    - E.g., Transactional Memory, thread-level speculation
  - Heterogeneous (including simpler) cores
    - Not all cores will be able to support MPI
- **At the system level:**
  - Global addressing (PGAS and APGAS languages)?
- **Limited memory per core**
  - Will often require new algorithms to scale

Different approaches to exploit parallelism



Potential migration paths

Green: open, widely available
Blue: somewhere in between
Red: proprietary



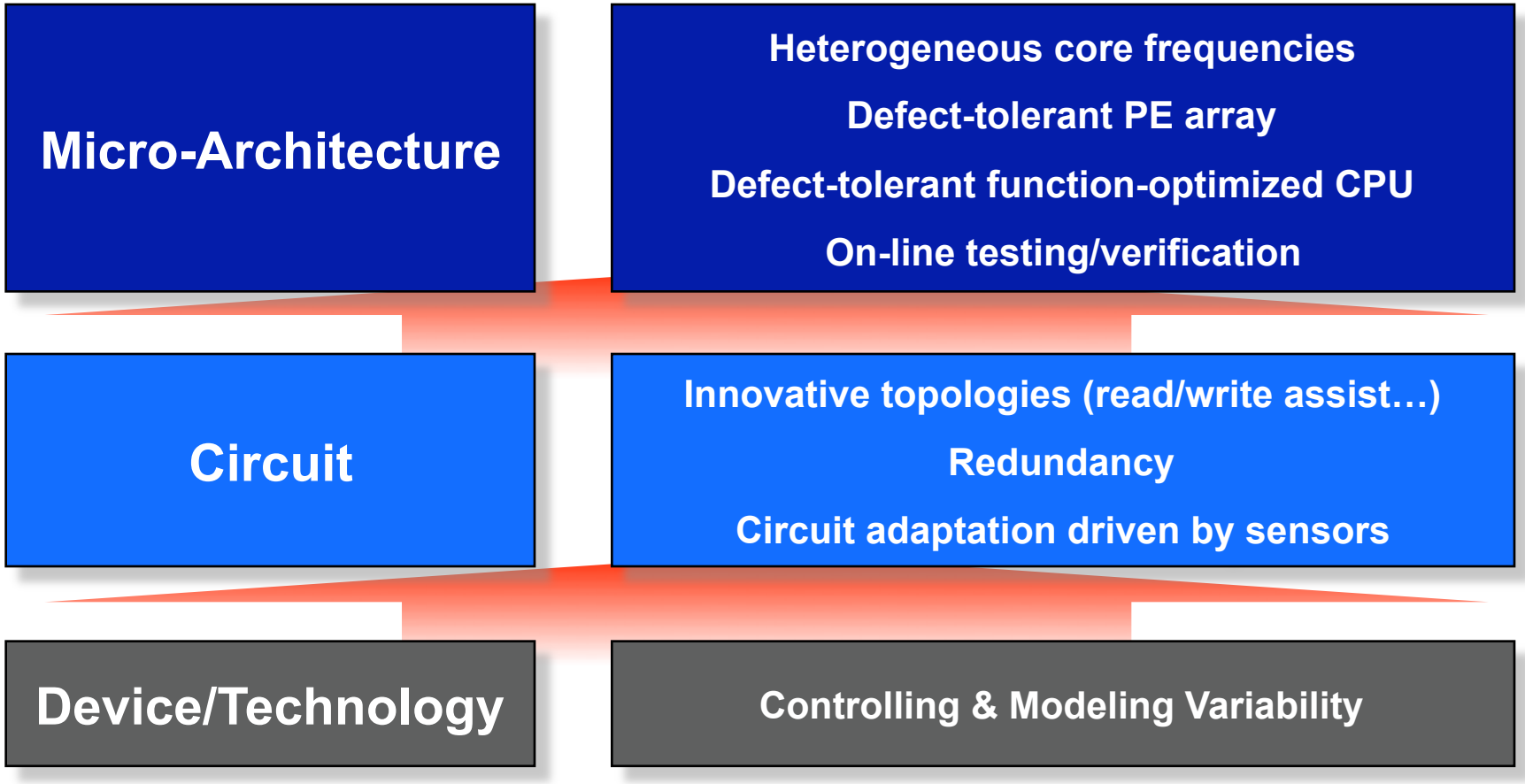
Reliability / Resiliency

- **From IESP: "The advantage of robustness on exascale platforms will eventually override concerns over computational efficiency"**
- **With each new CMOS generation, susceptibility to faults and errors is increasing:**
  - For 45 nm and beyond, soft errors in latches may become commonplace
- **Need changes in latch design (but requires more power)**
- **Need more error checking logic (oops, more power)**
- **Need means of locally saving recent state and rolling back inexpensively to recover on-the-fly**
- **Hard failures reduced by running cooler**

Shift Toward Design-for-Resilience

Resilient design techniques at all levels will be required to ensure functionality and fault tolerance

- **Architecture level solutions are indispensable to insure yield**
  - **Design resilience applied thru all levels of the design**



Reliability: silent (undetected) errors

- **How often are silent errors already occurring in high-end systems today?**
  - With Exascale systems we can compute the wrong answer 1000x faster than Petascale systems
- **Silent error rates are a far more serious concern for supercomputers than for typical systems**
  - Exascale systems will require systems to be built from the ground up for error detection and recovery
    - Including the processor chips
- **Fault-tolerant applications can help**

Some other issues we didn't cover

- **Interconnection networks**
- **Operating systems**
- **Debugging and monitoring**
- **Performance tools**
- **Algorithms**
- **Storage and file systems**
- **Compiler optimizations**
- **Scheduling**

Perspective on supercomputer trends

- **Vector systems gave way to killer micros**
- **Clusters of killer micros and SMPs have ruled for almost 20 years**
- **The ASCI program drove the innovation for these systems**
  - Leveraging commodity micros with interconnect, …
- **However, commodity killer micros aren't likely to be the answer for Exascale**
  - Back to the drawing board, with investment required from the ground up

A “Jeff Foxworthy” take on Exascale

- **If your system energy efficiency is >100 pJ/flop**
  - You might *not* have an Exascale system
- **If your algorithm doesn’t partition data well**
  - You might *not* have an Exascale algorithm
- **If your application is difficult to perfectly load-balance**
  - You might *not* have an Exascale application
- **If message-passing is the only means of providing parallelism for your application**
  - You might *not* have an Exascale application

Concluding thoughts

- **Getting to Exascale/Exaflops performance within 10 years will be tremendously challenging**
  - Power and cost constraints require significant innovation
  - Success not a foregone conclusion
- **Processor architecture and technology**
  - Low voltage many-core, SIMD, heterogeneity, fault tolerance
- **Memory and storage technology**
  - Closer integration, limited size, and Phase Change Memory
- **Programming models and tools**
  - Must deal with parallelism gone wild!
  - Hybrid programming models, PGAS languages
- **An exciting time for parallel processing research!**

Exascale