Beyond Performance

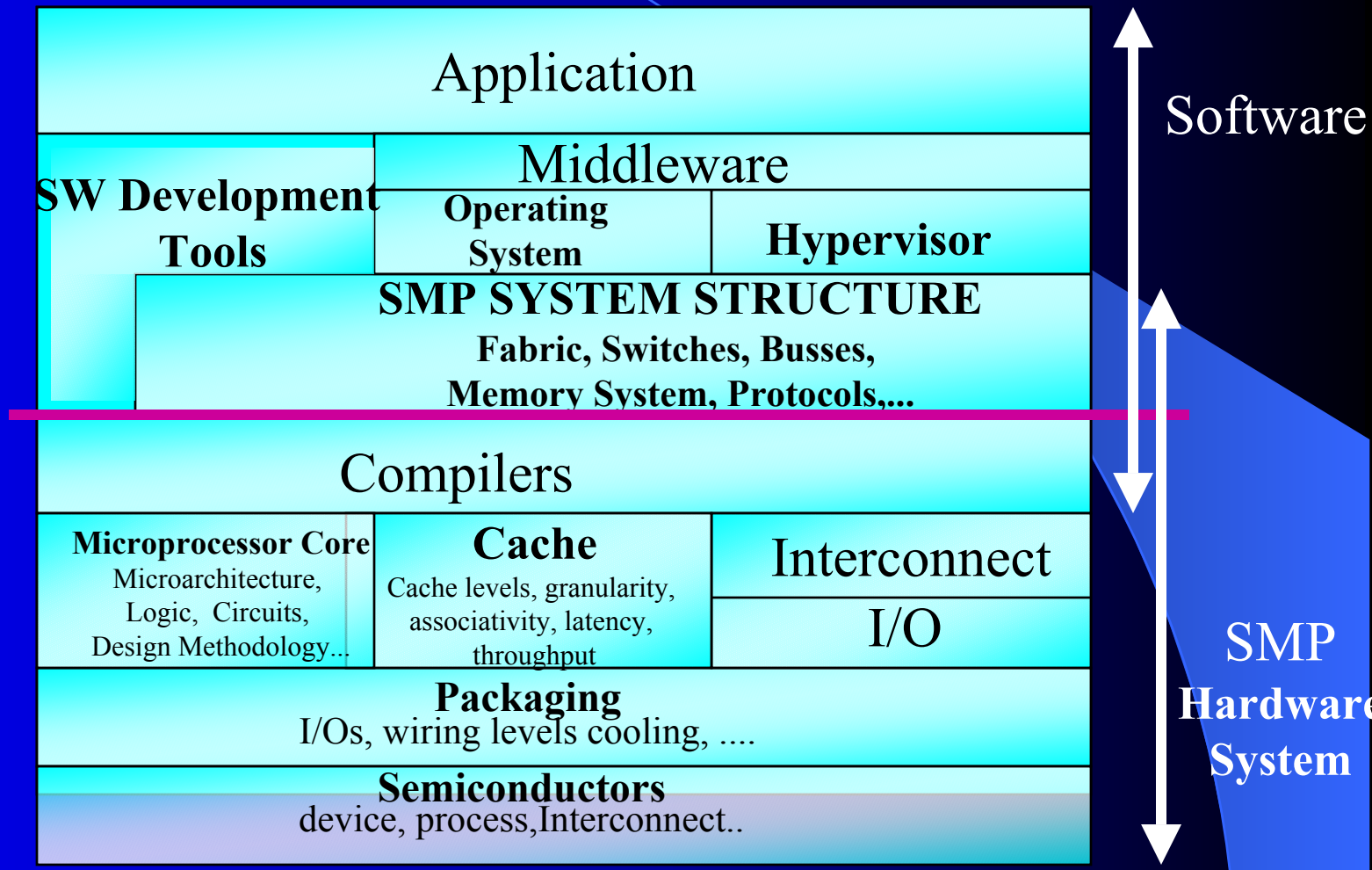
Some other challenges for System Design

Eric Kronstadt
IBM T.J. Watson Research Center
Yorktown Heights, NY

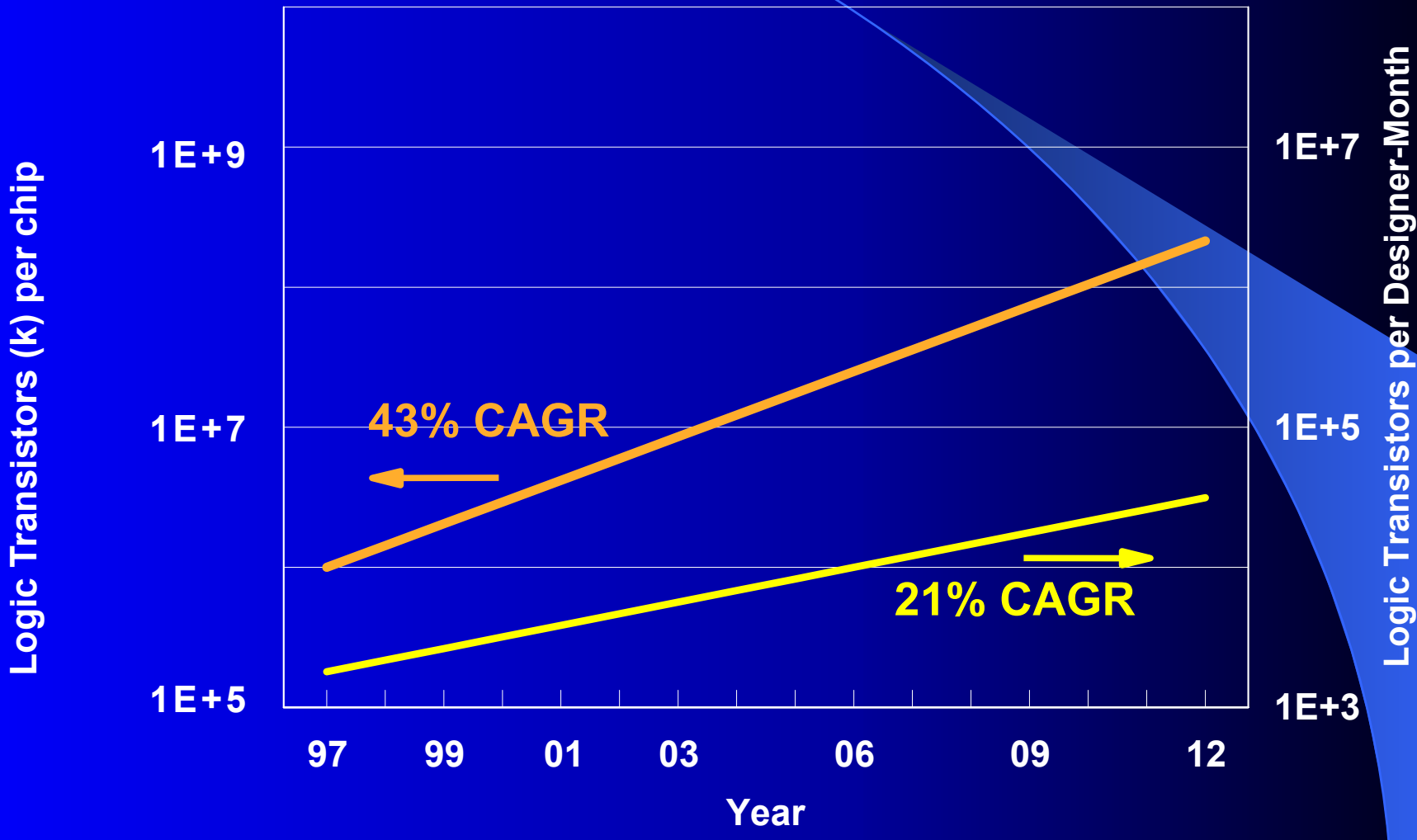
I. Design

System Performance Stack

Performance improvements will increasingly require system level optimization



Design Productivity Gap Growing



source: Sematech

Super Exponential Design Complexity

Thousands

# of Transistors

Billions

Functionality + Testability

Functionality + Testability + Wire Delay

Functionality + Testability + Wire Delay + Power Mgmt

Functionality + Testability + Wire Delay + Power Mgmt + Embedded Software

Functionality + Testability + Wire Delay + Power Mgmt +Embedded Software + Signal Integrity

Functionality + Testability + Wire Delay + Power Mgmt +Embedded Software + Signal Integrity + Hybrid Chips

Functionality + Testability + Wire Delay + Power Mgmt +Embedded Software + Signal Integrity + Hybrid Chips + RF

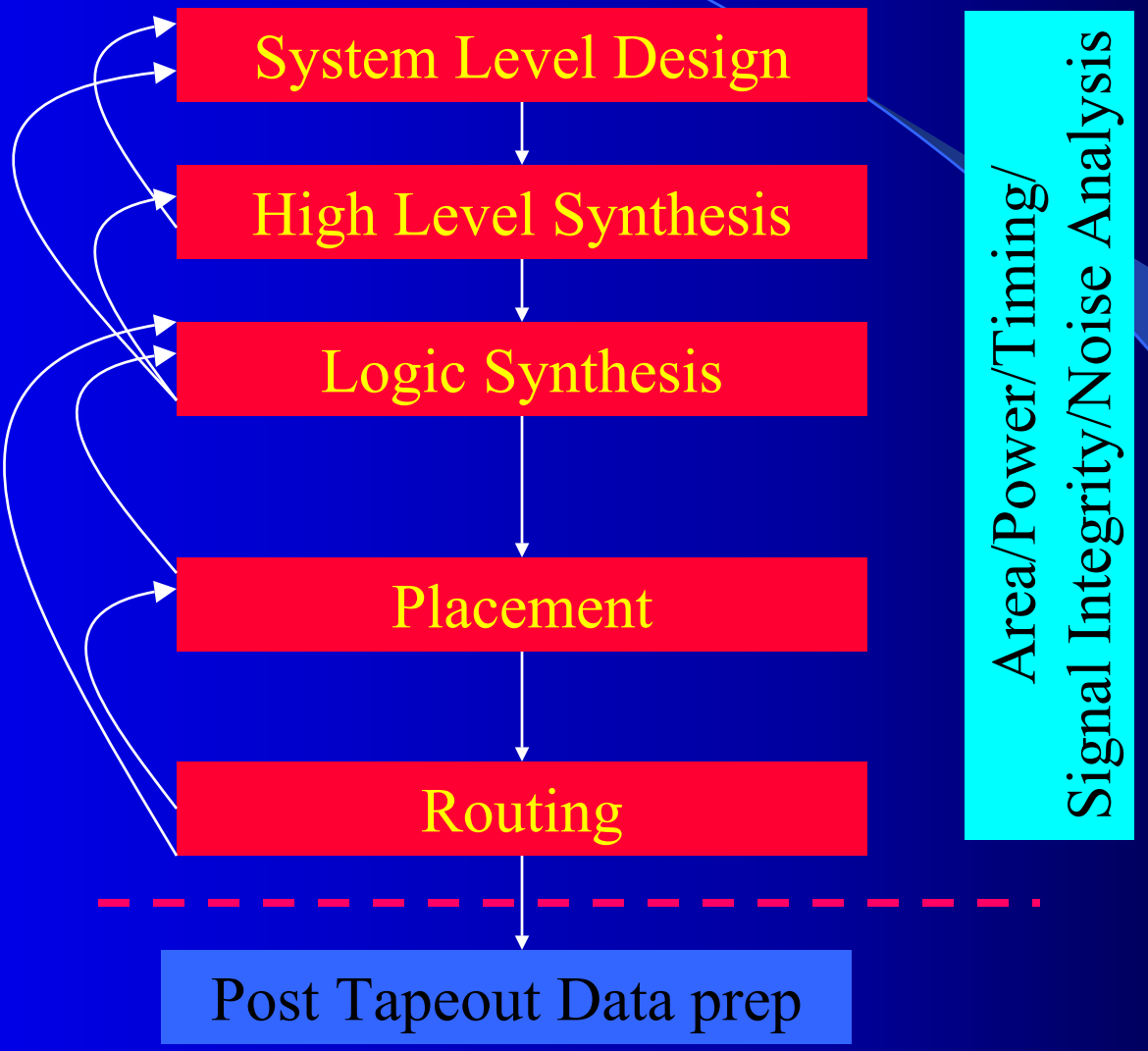
Functionality + Testability + Wire Delay + Power Mgmt +Embedded Software + Signal Integrity + Hybrid Chips + RF + Packaging

Functionality + Testability + Wire Delay + Power Mgmt +Embedded Software Signal Integrity + Hybrid Chips + RF + Packaging + Mgmt of Physical Limits

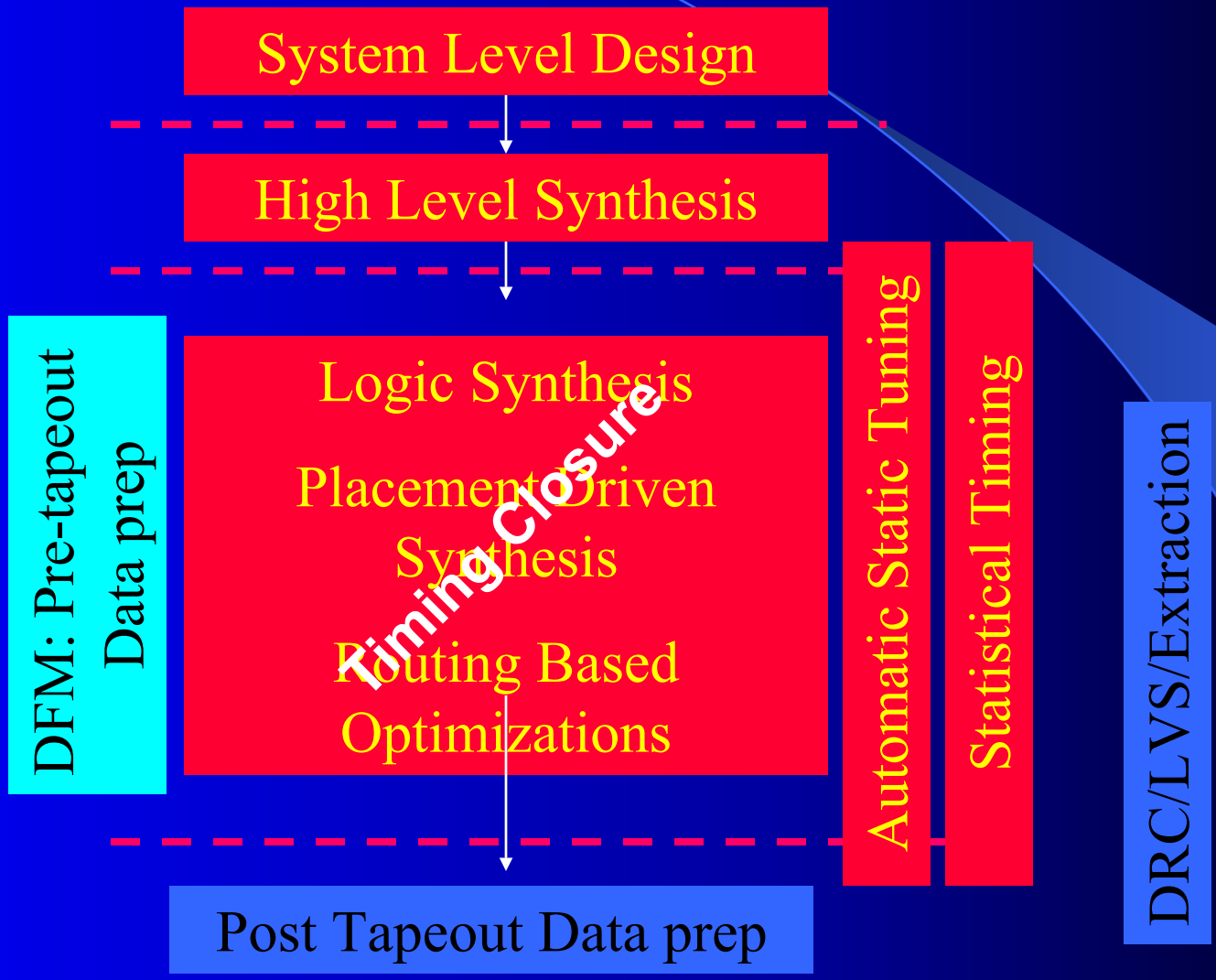
Exponentially growing number of elements (devices & wires)

Exponentially growing number of Intrinsic Design Challenges

Traditional Design Flow



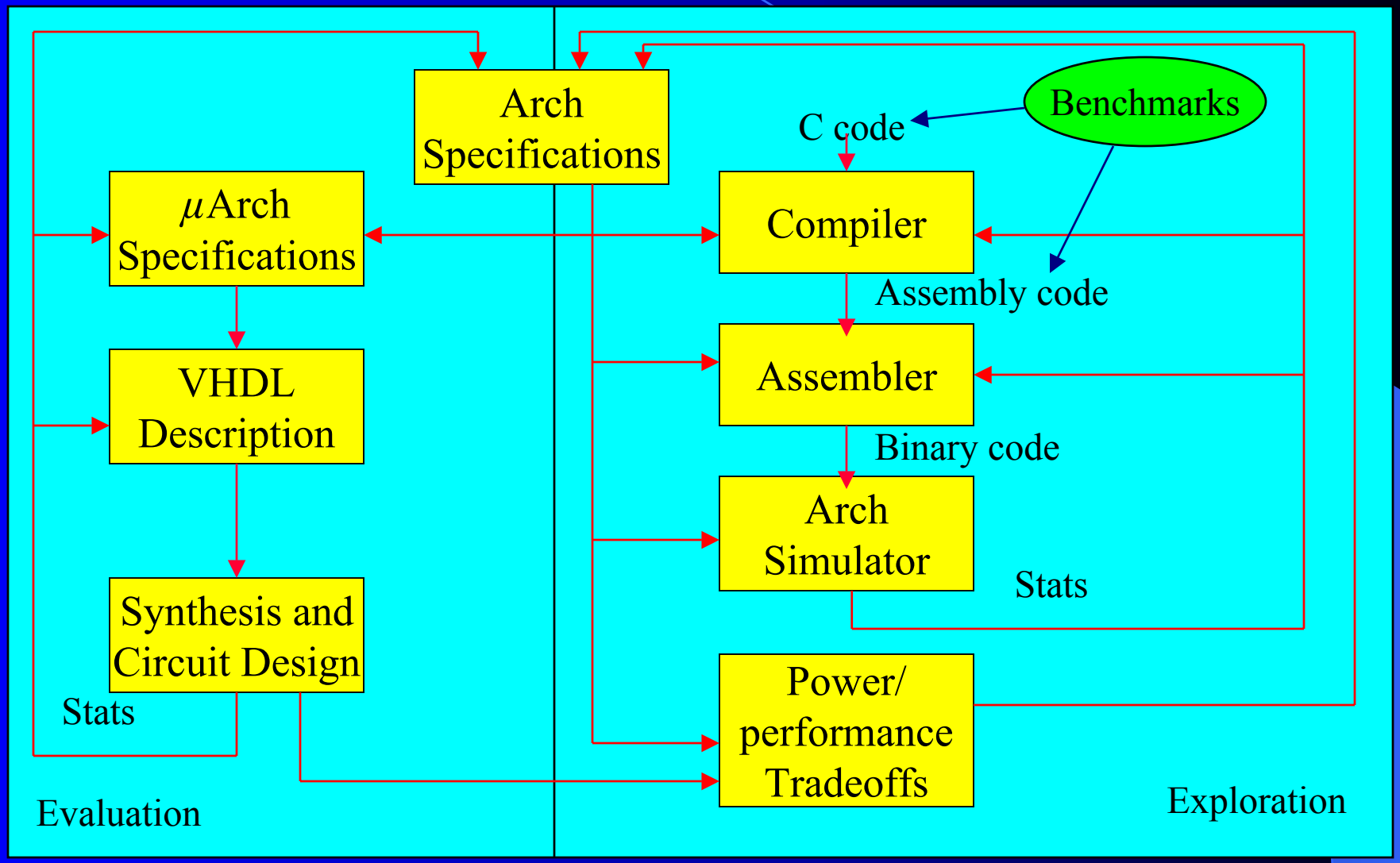
Future Trends and Shifts



Co-Design Craze

- Hardware - Software Co-Design
  - **Driven by SOC opportunities**
- Circuits - Technology Co-Design
  - **Feedback from Circuit Designers to Semiconductor Process developers has always been a good practice**
  - **Becoming a necessity - Driven by need to go beyond scaling laws**
- Technology – Circuit - Microarchitecture (Architecture?) Co-Design
  - **Driven by need for lower power – Power Aware Microarchitecture**

"Technology - Circuit - Microarchitecture - Architecture – Compiler" Co-Design



Applying “feedback” to fundamentals

- **Example: Power Management**
  - Already, chips have temperature sensors that feedback information to voltage or frequency controllers
  - **Opportunity:**
    - **Feedback of off chip environmental conditions**
- **Example: Cycle time**
  - Dynamic Clock variations, noise and across chip line variations require (static) require margin in cycle time
  - With multi-Ghz frequencies this margin is becoming a non-trivial part of cycle times
  - **Opportunity**
    - **High frequency I/O links already have dynamic frequency feedback**
    - **Could we dynamically feed back environmental variations to control clock behavior, and reduce design margin**

II. The Memory Wall

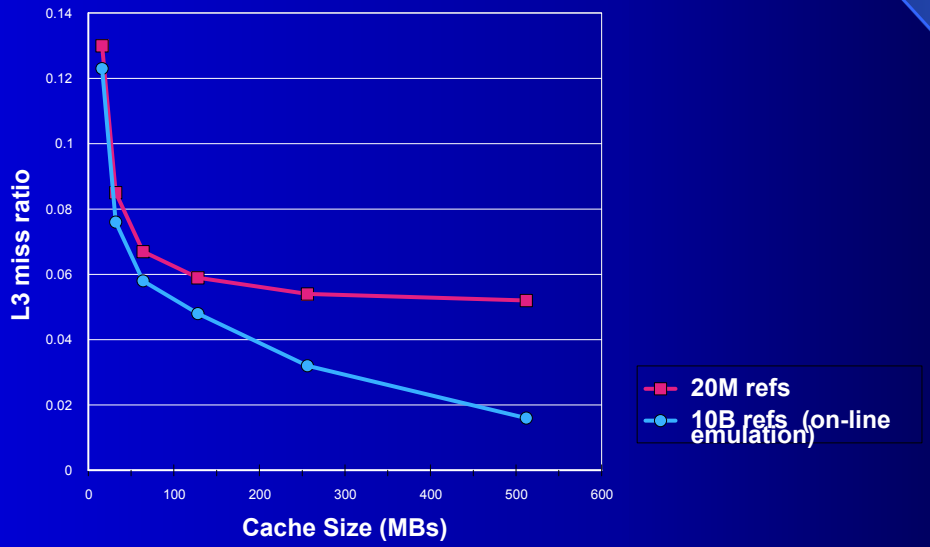
The Memory Wall

A pre-historic problem that won't go away soon!

- **Memory access times recently quasi-static (~200 n)**
  - Processor speedup expected ~ 60% p.a.; DRAM speedup ~10% p.a.
  - Increasing number of processor cycles as processor speeds have increased by an order of magnitude
  - Implication is significant increase in CPI
- **Finite L2 cache CPI typically greater than 2x infinite cache performance**
- **For multiprocessors, inter-cache latencies increase this degradation to 3x or more for 4 processors and up**
  - Perfect cache performance worsens because of locking and synchronization
    - **Greater path length than on a uniprocessor**
    - **Synchronization operations are slow**
  - Even single thread performance worsens as the number of processors increases

Traditional Approaches to break through the Memory Wall

- Larger caches, deeper cache structures
  - **Going from 64MB to 512MB, the miss ratio drops to a third for TPC-C**
  - **Later generations (1GHz+) would see even more benefit**



- Latency hiding via prefetching (h/w, s/w, both)
- Out of Order execution, speculative execution, hardware multithreading
- Special hardware (and new hw algorithms) to support coherency, partitioning, cache/memory replacement

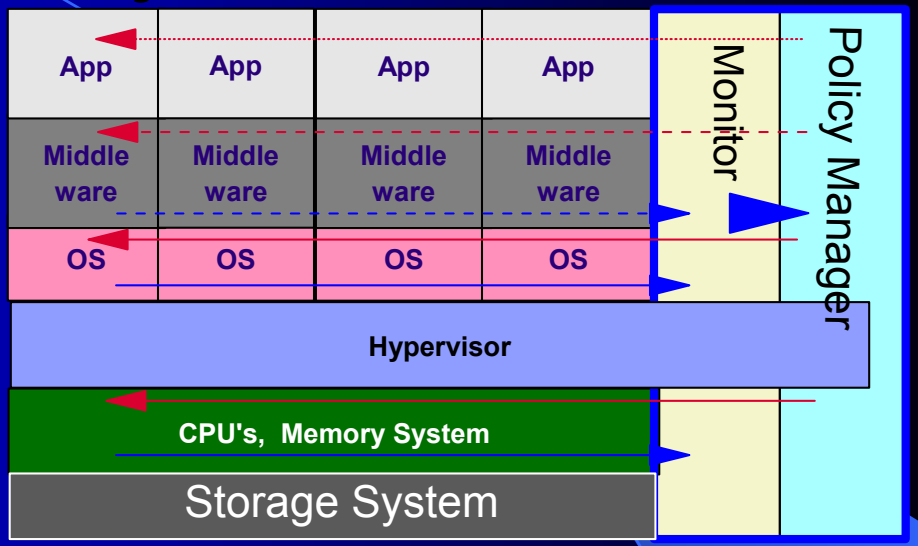
What else is needed

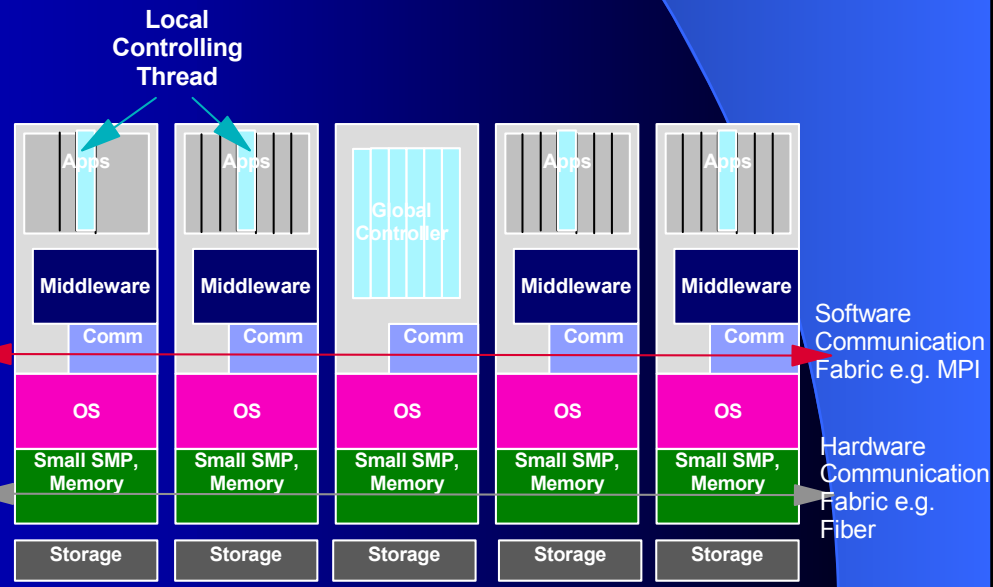
Low Hanging Fruit: CPI impact of scaling can be reduced by better task scheduling and cache affinity

- Other Opportunities:
  - **Machine learning applied to code prefetching and code pre-positioning**
  - **Self-optimizing cooperation between the hardware and software directives**
    - Software support for streamlined or 'essential' coherency and synchronization
    - Gives hints and directives
  - **Policy manager: Merges software directives with hardware inputs using run-time mgmt. And adaptation**
- New computing paradigms with programming models designed to better tolerate memory latency

Integrated HW/SW approach to the Memory Wall

- Traditional SMP Systems
  - New h/w capabilities: Coherency, synchronization, partitioning, cache/memory replacement algorithms, pre-pushing, access to control information
  - Software: supports streamlined or 'essential' coherency, synchronization; gives hints and directives
  - Policy manager: Merges software directives with hardware inputs using run-time mgt. and adaptation
- Standard HW-based systems
  - Hardware: commodity processing nodes, interconnect
  - Global controller: data placement, task allocation guidelines, detection of communication patterns, recording data location, tracking health of node, moving/prefetching data, task allocation, affinity scheduling, reconfigure around faulty nodes. Works via distributed local controllers





III. Power

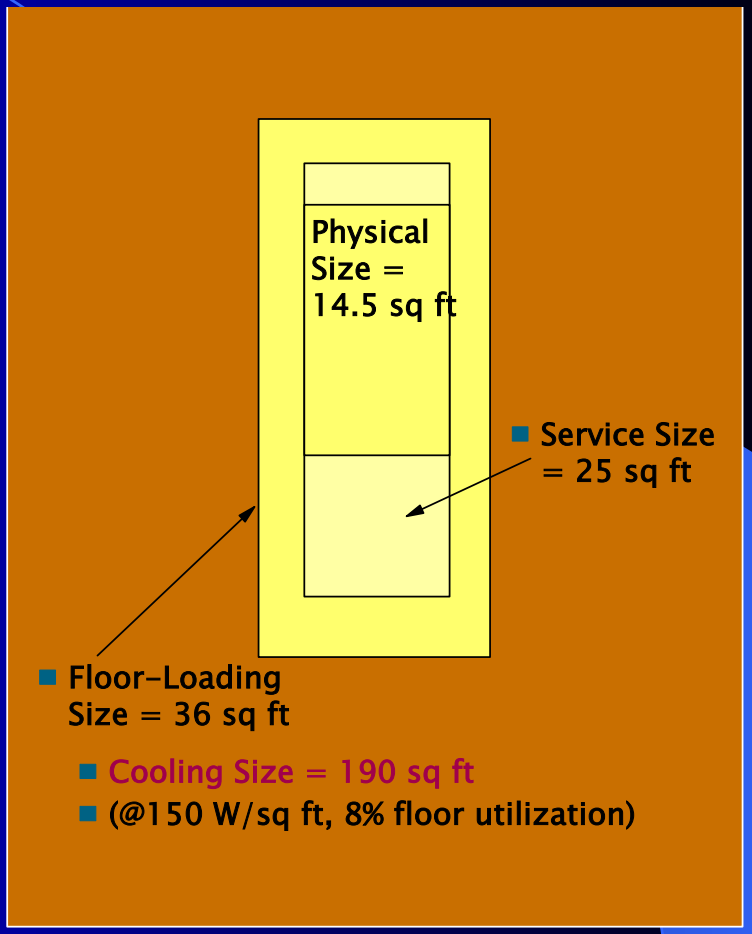
Electrical power consumption for IT is projected to reach crisis proportions

"What matters most to the computer designers at Google is not speed, but power -- low power, because data centers can consume as much electricity as a city."

- Eric Schmidt, CEO Google (Quoted in NY Times, 9/29/02)

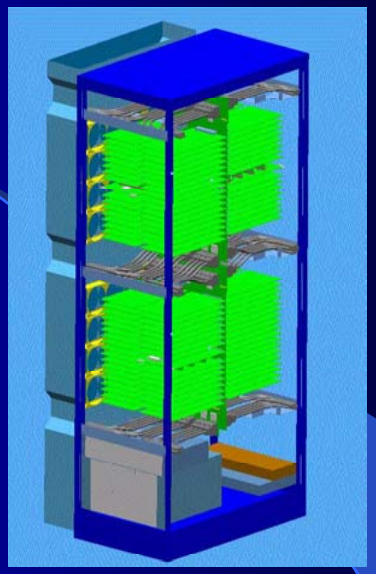
- Server farms power consumption increases exponentially, led by communication equipment
- CPU performance will be increasingly limited by power and cooling
- Communicating information consumes much more power than processing it - opportunity to optimize
- More ....
  - On a watts-per-sq-ft basis server farms use more energy than automobile plants
  - 27 farms proposed for South King County area near Seattle will require as much energy as the city of Seattle which includes the Boeing factory - termed as the "2,400 megawatt problem"
  - San Jose City Council approved 250 MW power plant for US DataPort server farm and allows installation of 80 back-up diesel generators at that site - political battles to follow
  - 60% of server farm cost is energy needs
  - 40% of power consumption is for air conditioning!

Power affects the system size

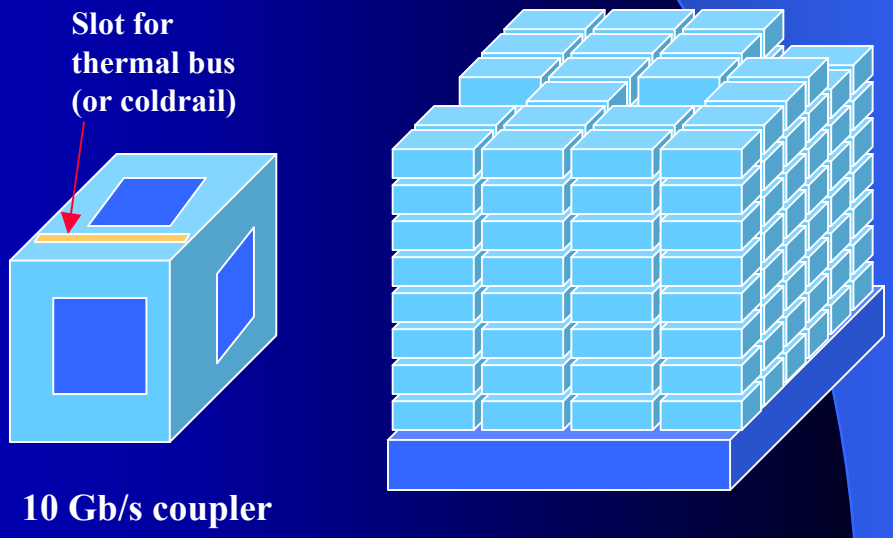


Solutions for High Compute Density

- BlueGene/L
  - Use lots of low power low performance processors
    - 128 Compute cards
    - 8 nodes (@2 processors/node)
    - 1024 compute nodes (2048 processors)
    - 2.8TF Peak per Rack
    - 64 Racks / System



- Ice Cube
  - Do not skimp on processor power
  - Water cool instead
    - SMP, memory, disk, in 9” cube
    - Nearest neighbor coupling



Traditional Approaches to Power Problem

- Design Techniques
  - Low power silicon processes e.g. SOI
  - Power efficient devices, circuits, latches, arrays
  - Multiple voltage islands
  - Multiple threshold devices
  - Dynamic frequency and voltage scaling
  - Dynamic bias control
- Power aware microarchitectures
  - Adaptive structures
  - Clock gating - coarse and fine grained
  - Fine grain control of frequency and voltage scaling
  - More intelligent approaches to redundancy and speculation

Approaches are based on power as a hardware design problem (like chip area)

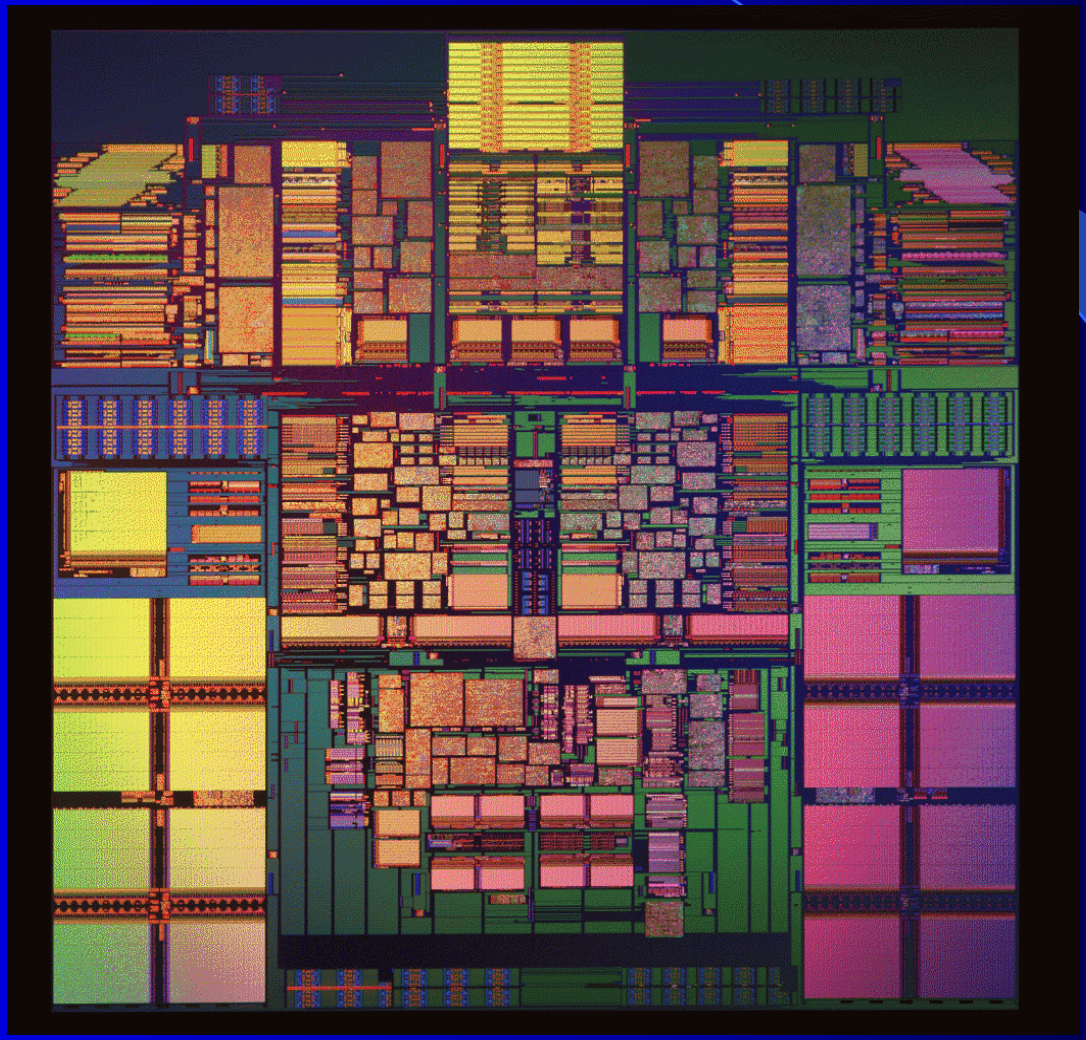
Opportunities lie beyond the traditional HW approaches

- **Self-optimizing cooperation between the hardware and software**
  - **Notion of "thermal power" is being redefined**
    - Frequency/voltage adjusted if chip gets too hot
  - **Feedback to allow workload management software to take power into account**

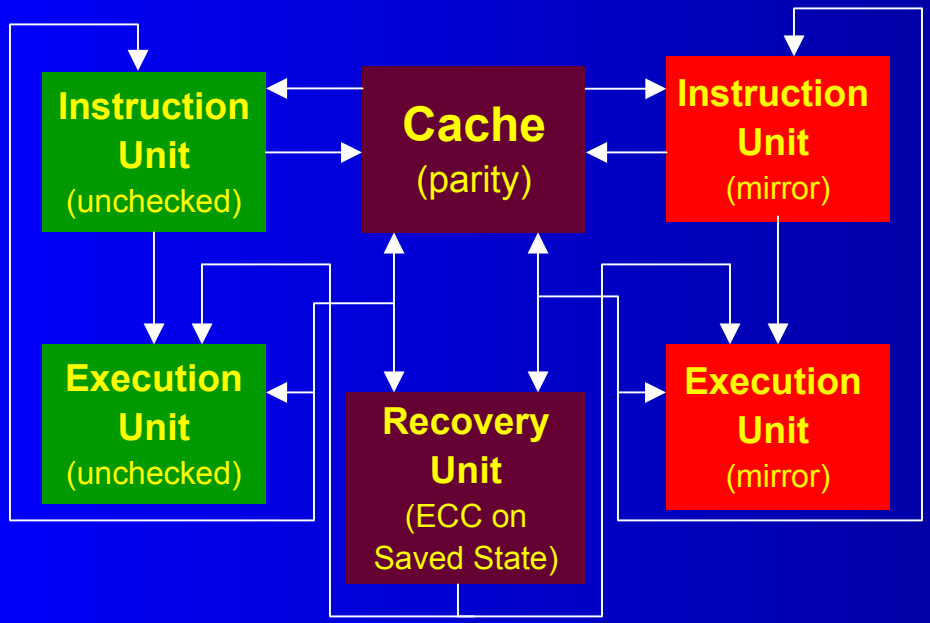
IV. Availability/Reliability

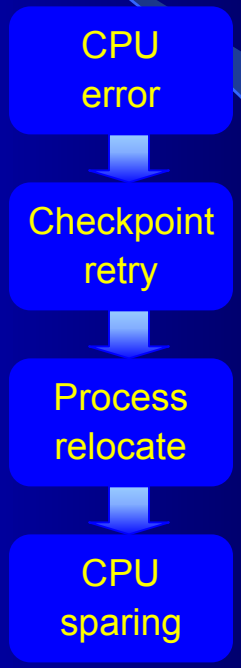
S/390 (zSeries) G6 Processor

Achieving Five 9's



zSeries CPU Fault Tolerance





- Detect error
- Instruction retry
- Hardware Control
- Transient recovery
- Permanent fault
  - Retry ineffective
- Restore full Capacity
- Move work to another CPU

How do BlueGene/L and Ice Cube deal with this?

- **BlueGene/L –** "lower quality" parts – little ECC or recovery infrastructure
- **Ice Cube –** components are physically inaccessible

Solution: Learn to live with it – **"Fail in Place"**

- Requires SW and system structures to "route around" faulty components

Ultimately….

- Reliability must be handled as a full system issue
  - **Operating System**
  - **Middleware**
  - **“Front” and “Back”-end Applications**

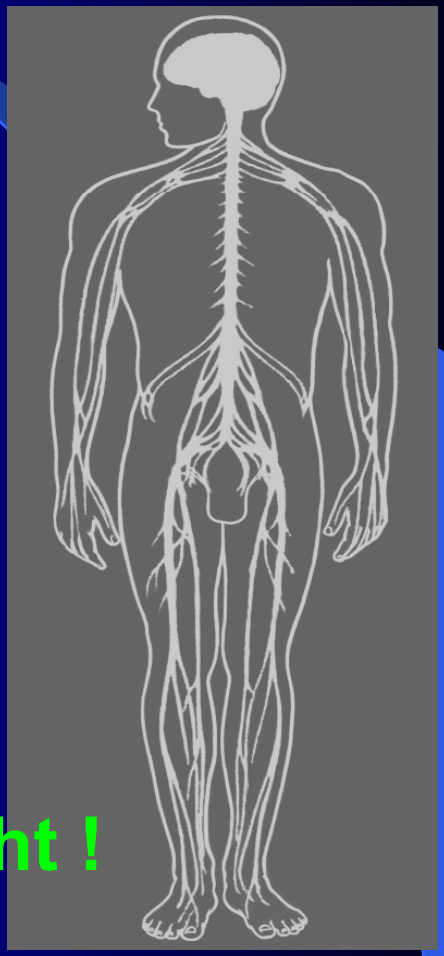
Autonomic Computing

Self-configuring
Adapt automatically to dynamically changing environments

Self-healing
Discover, diagnose, and react to disruptions

Self-optimizing
Monitor and tune resources automatically

Self-protecting
Anticipate, detect, Identify, and protect Against attacks From anywhere



The evolutionary sequence was right !
We got it backwards !

Why Autonomic Computing?

- Complexity becomes harder to control and continues to grow
  - **Number of environments and systems touched by an application**
    - Pervasive devices, clients, browsers, web servers, fire walls, application servers, back end servers, etc.
    - Unknown dependencies
- Pace of change makes continuous/dynamic optimization essential, but more difficult
  - **Number of changes of application parameters**
  - **Hardware and software version control**
- People Cost becomes a barrier
  - **Number of support people becomes prohibitive**
    - Over 50% of total IT costs
  - **Skills shortage impacts deployment of new projects**
    - 41% of ebusiness projects delayed because of skills shortfalls
    - Skills available to fill roughly half the demand

Autonomic Computing will become a compelling necessity for businesses – A key to the future of computing

What's interesting about autonomic computing?

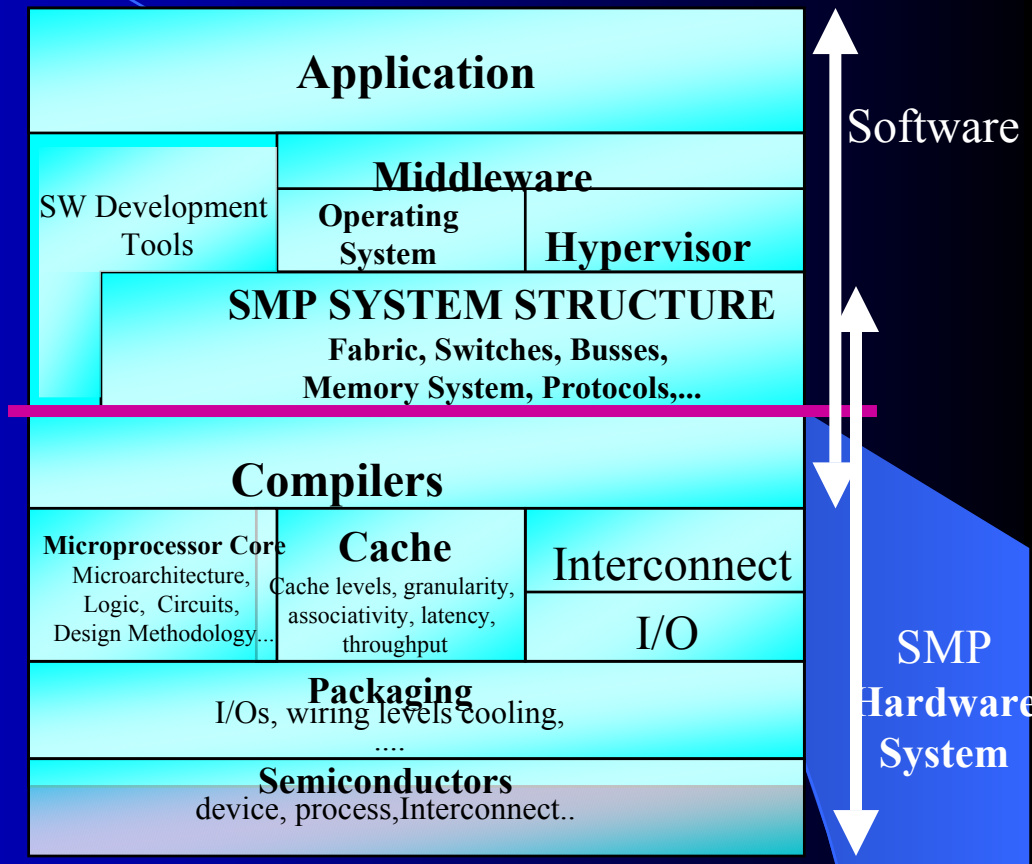
- **Not entirely new –**
  - Systems management
  - A form of process control
  - Application of adaptive control
- **Can address many of the above themes**
  - Reliability/Availability (of course)
  - Memory Wall
  - Power efficiency
  - Performance / frequency management
- **An attempt to deal with complexity**

Complex Systems

- **Historically human beings deal with complexity by abstraction and specialization**
  - Almost all fields of human endeavor eventually evolve specialties
    - **And generalists give way to specialists**
  - It's what we do in Mathematics, Science, Engineering, etc.
- **From a systems point of view specialization leads to componentization**
  - Formal and informal interfaces make this happen
  - We need specialists called integrators

Abstraction and specialization has significantly advanced the state of the art of computing

- **Computing systems**
  - Separation of hw & Sw
  - Concept of architecture
    - **Processor architecture**
      - Microarchitecture
    - **System architecture**
      - Processor, memory, storage, I/O, communications
  - Software and Communications Layers
  - Computing device types
    - **Handhelds, clients servers, supercomputers**
      - Tiered servers



- How we think about Computer design
  - Semiconductors: Devices, interconnect, lithography
  - Chip design: Processors, support chips – logic, circuit, physical design
  - Card, Board, Racks – Mechanical, interconnect'
  - Power, Packaging Cooling

Advantages of componentized approach

- **Manages complexity**
  - Divide and conquer
  - We can't do this without it
- **Allows focus of intellectual attention**
  - Deep expertise
- **Enables institutional and shared learning**
- **Enables Manageable investments**
  - Encourages competition
  - Generates
    - **Lower cost**
    - **Greater variety**
    - **Overall better quality**

Shortcomings of componentized approach

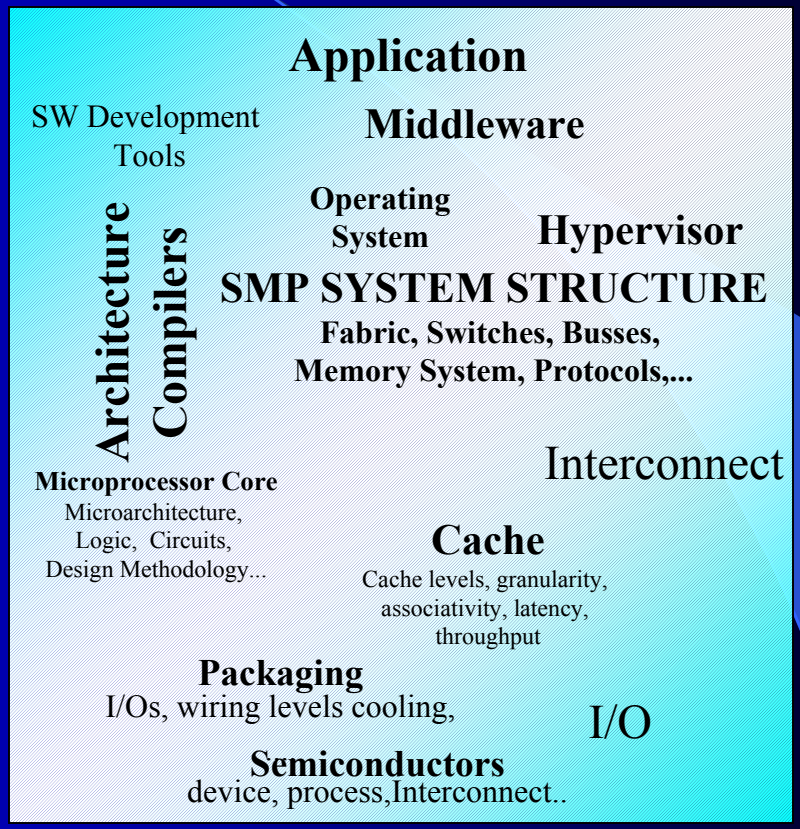
- **Tendency to local optimization**
  - Centrifugal organizational forces
- **Problems / issues that cross boundaries are poorly attended to or not attended to at all**
  - Communications even language becomes differentiated
- **Bursts of progress occur where distinct infrastructures, organizations, cultures, etc., are brought into contact with one another**
  - New opportunities
  - New ideas
  - Progress

The term Holistic is frequently used to refer to the opposite of specialistic

Towards Holistic Design

- **"Holistic":** (Webster's) **Emphasizing the importance of the whole and the interdependence of its parts.**

- **"Holistic Design":** (Warren's) **Emphasizing the importance of the whole technology stack and the interdependence of its components in the design process.**



The most competitive solutions going forward will optimize their solution across the entire technology stack: application software to process technology.

Towards Holistic Design

- **Consideration of more of the entire design flow at each stage**
  - Spanning much earlier and much later in the design flow at any particular stage
- **More "Co-Design"**
- **Longer feedback loops and adaptive control paths**
  - Longer in the sense of distance in the traditional design hierarchy
  - Interaction of
    - **Software and Circuits**
    - **Microarchitecture and device design**
- **It's not all that easy**
  - Teams are comfortable with the old process
  - May require more people/skills
    - **Certainly requires different definition of skills**

Holistic Design

- **But is it new?**
  - "System Design"
  - "Tall Thin Designers"
  - Mead Conway approach to VLSI Design
  - ….
- **On the other hand we may end up with…**
  - A different picture of how pieces fit together
    - **A different component taxonomy**
  - A different way of thinking about the way we do design