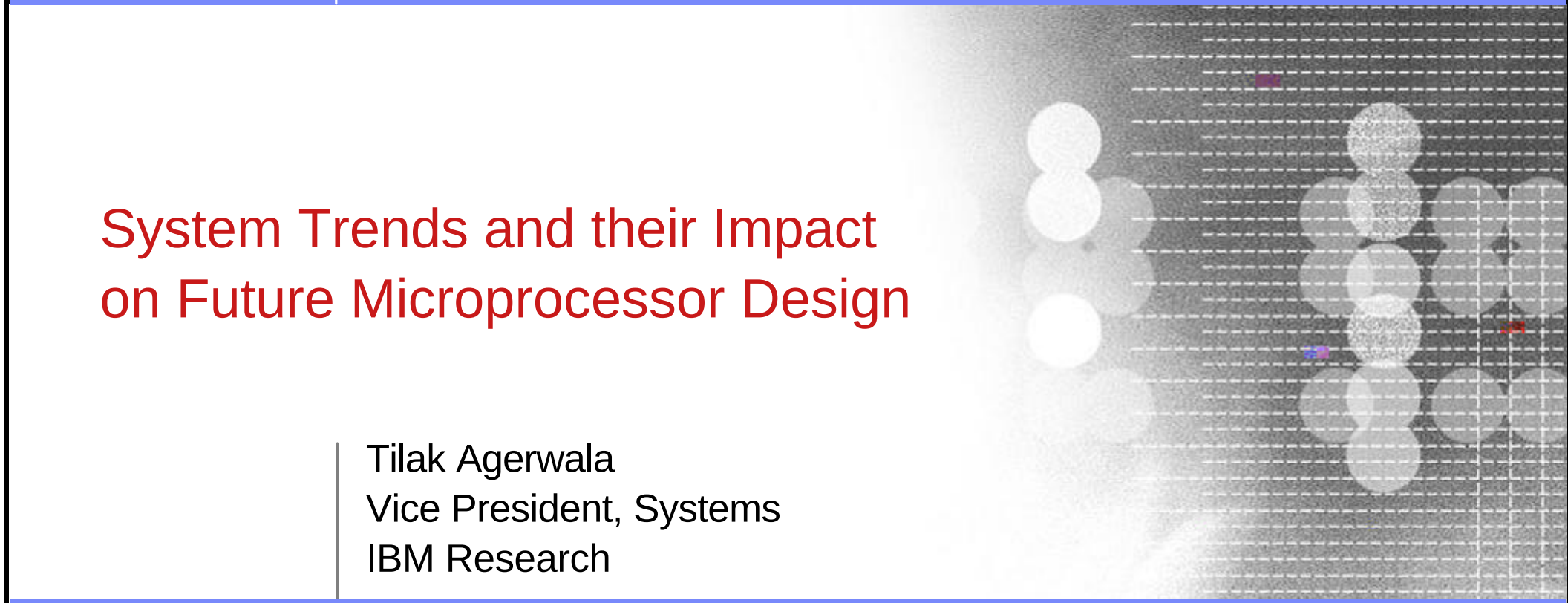
IBM Research



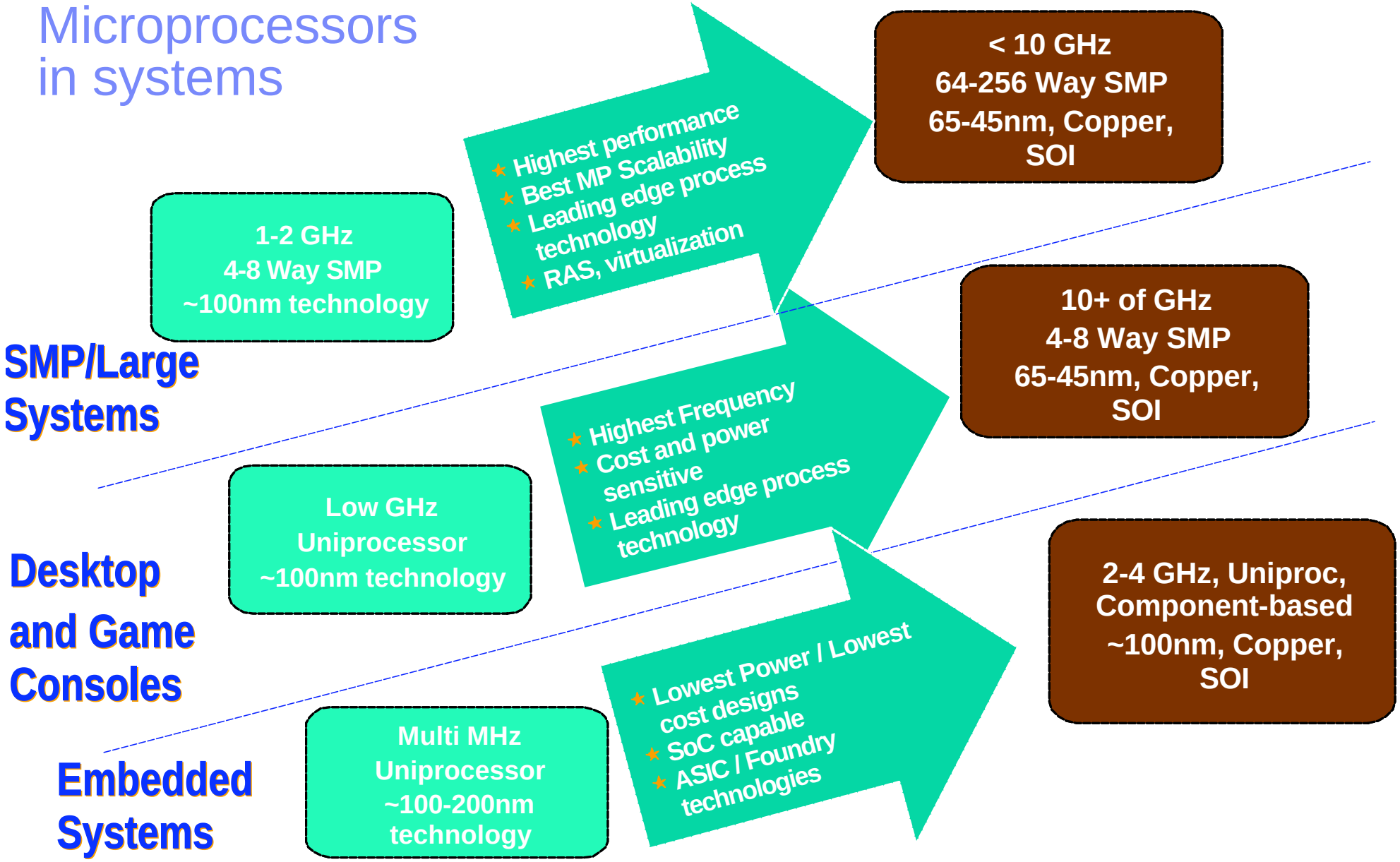
# System Trends and their Impact on Future Microprocessor Design

Tilak Agerwala
Vice President, Systems
IBM Research

# Agenda

- System and application trends
- Impact on architecture and microarchitecture
- The Memory Wall
- Cellular architectures and IBM's Blue Gene
- Summary

# Microprocessors in systems



# Large system application trends

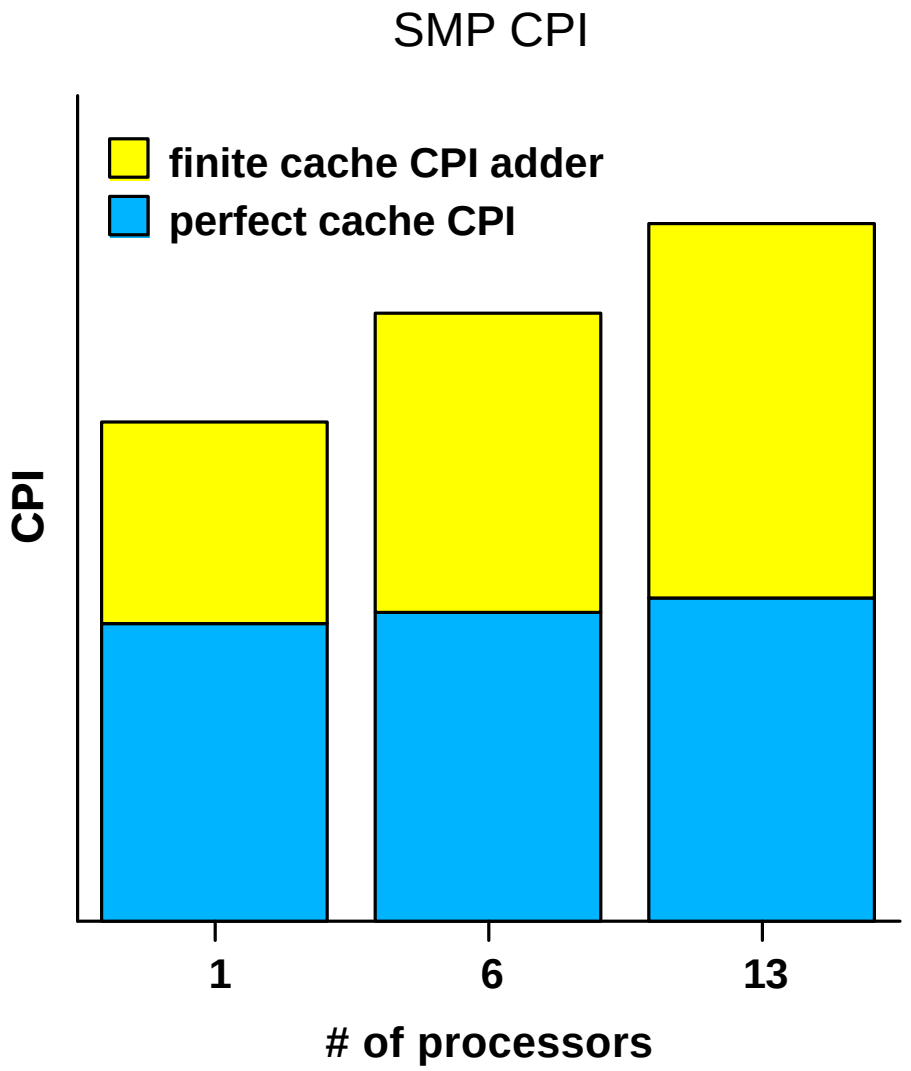
- Traditional commercial applications
  - Databases, transaction processing, business apps like payroll etc.
- The internet has driven the growth of new commercial applications
- New life sciences applications are commercial and high-growth
  - Drug discovery and genetic engineering research needs huge amounts of compute power (*e.g.* protein folding simulations)
- Important applications will *scale out*
  - Large-scale parallelism
  - Little or no interaction between computations
  - *e.g.,* web application serving, life sciences, softswitch, video streaming, financial front-ends, ERP, CRM, eProcurement

# Large SMP systems

- SMPs support *upward scalable* workloads
  - Workloads that scale well with more processors under a single system image
  - Close interaction between threads
  - *e.g.,* databases, decision-support systems
- SMPs are also efficient with workloads that scale out
  - Other cost-effective solutions are available
- Today: larger than 16-32 way SMPs
  - 64-256 way systems foreseen for future workloads
- Robust RAS characteristics
- Robust virtualization/logical partitioning capabilities
  - Staple on the mainframes for decades
  - Cross-pollination with other high-end platforms is happening now
  - Likely to trickle down to every platform except, maybe, embedded

# Synchronization and coherence structures create new bottlenecks in SMPs

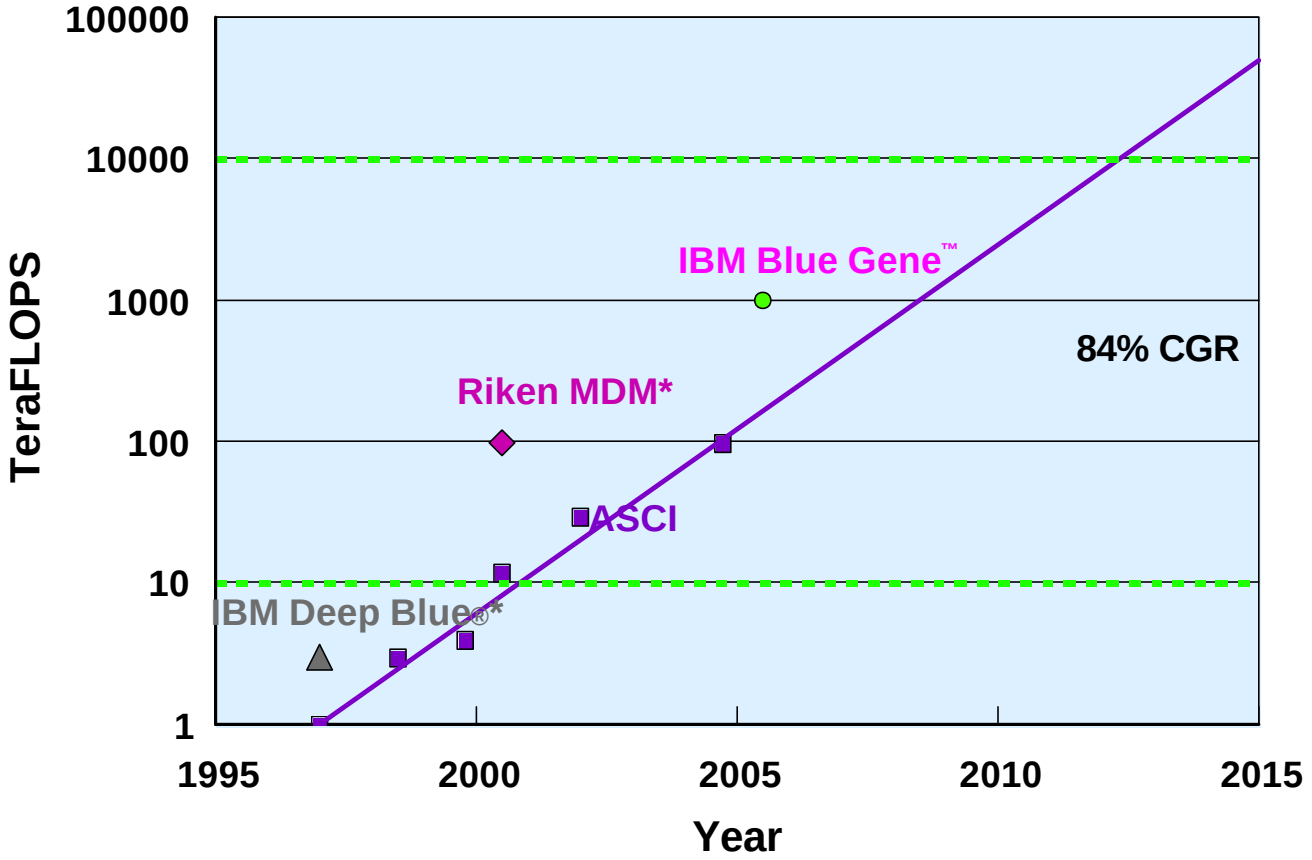
- Even the single thread performance worsens as the number of processors increases
  - Perfect cache performance worsens because of locking and synchronization
  - Greater OS pathlength than on a uniprocessor
  - Synchronization operations are slow
- Finite cache performance deterioration
  - Interaction among processors



# Large "blade" systems

- Usually intended for workloads that scale out
- Logically same as clusters
  - Classic distributed memory multicomputer systems
- "Data center" or "grid" in a box
- Many inexpensive, possibly heterogeneous, nodes called *blades*
- Shared power/storage infrastructure
  - Must minimize total system power and cost
- Each blade can be a small SMP
- One OS instance on each blade (typical)
- Simplified system management

# Large high-performance systems




New trends are towards

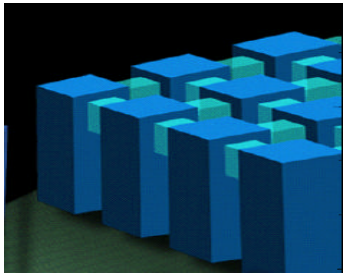
- High compute density
- Low cost
- Power efficiency
- High system reliability

* Special purpose machines, i.e. chess, molecular dynamics, protein folding.

Source: ASCI Roadmap www.llnl.gov/asci, IBM
Moravec 1998, www.transhumanist.com/volume1/moravec.htm

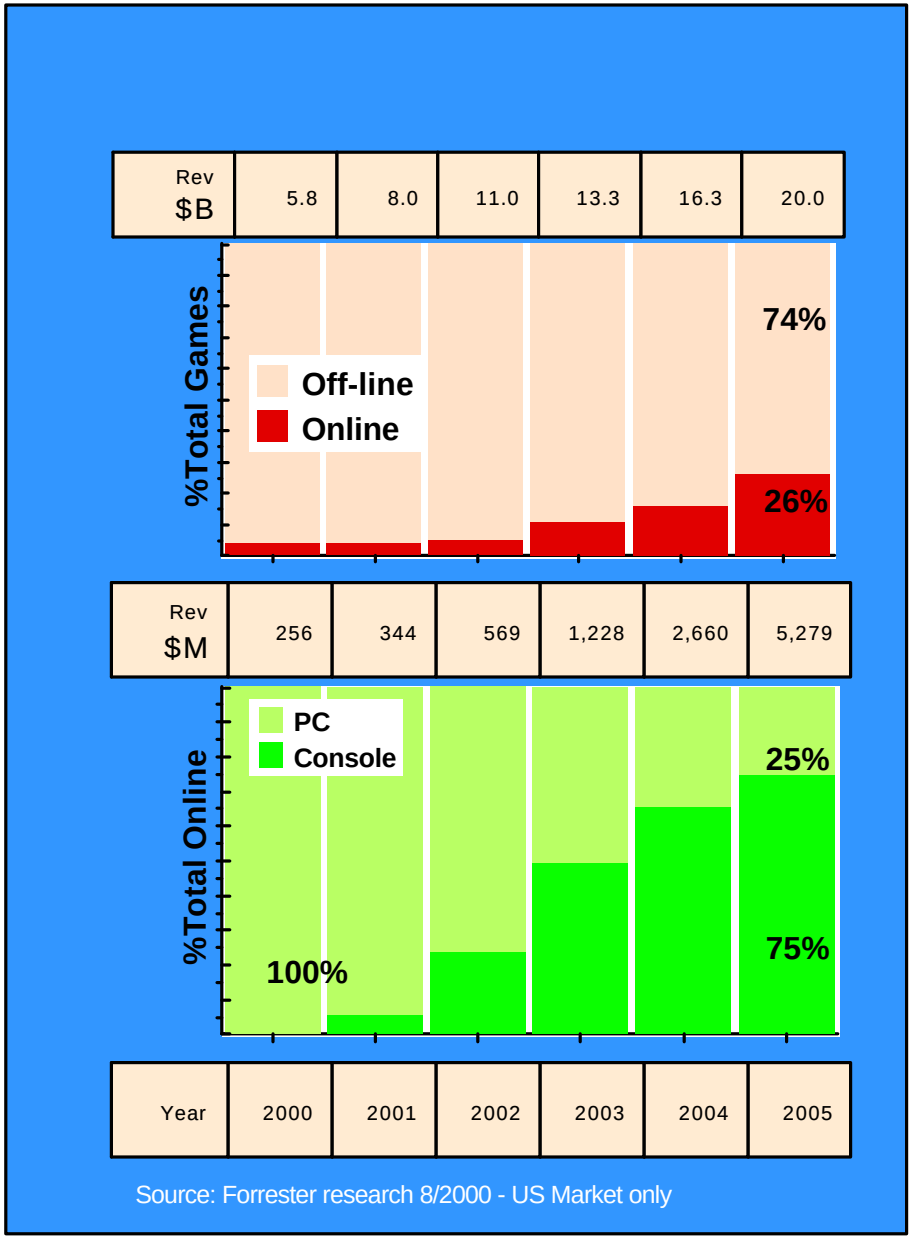
# Evolution of game applications and infrastructure

***Time***

| | 2001 | 2003 | 2005+ |
|---|---|---|---|
| Nature of Applications | • Pre-rendered<br>• Fixed content | • "Transactional"<br>• Enhanced user interaction<br>• Automatic generation<br>• On-demand<br>• Timely updates of versions | • **Full interaction, immersive**<br>• **Highly collaborative**<br>• **Dynamic synthesis, generation**<br>• **Adaptive**<br>• **Global** |
| Distribution Media |  **CD/DVD** | **The Internet** | **"On-demand"** |
| Infrastructure |  |  |  |

# Online gaming potential

- Online gaming set to become a significant part of the total games market
- Console-based online gaming is expected to pass PC-based online gaming in 2003-2004 timeframe



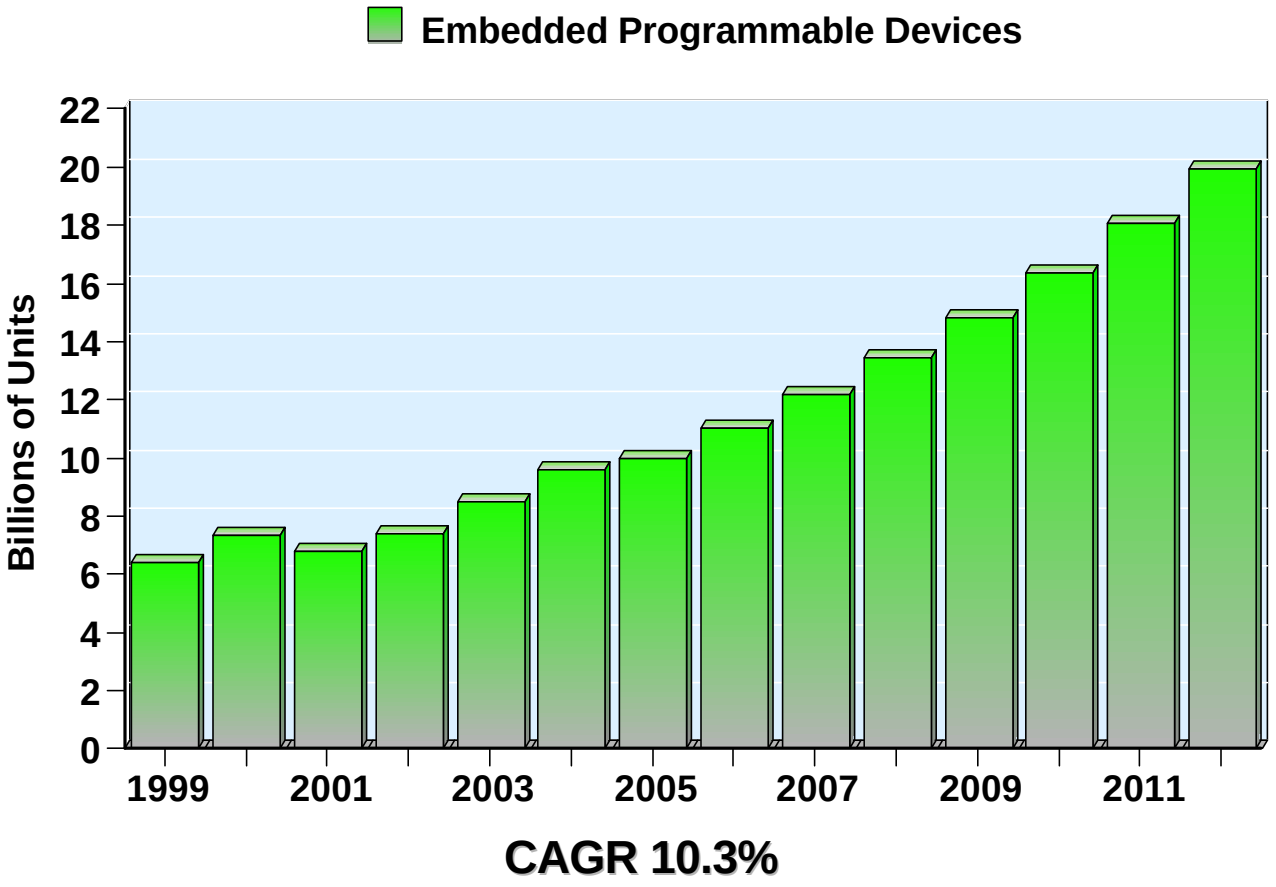
# Game processors and systems

- Huge growth in console game processors
- Desktop processors will push the frequency curve for the foreseeable future
- Desktop processors can be retooled, adapted, or used without modifications in game consoles
  - Special hardware (*e.g.* nVidia add-on cards)
  - Attached processors (*e.g.* Sony PlayStation2)
  - Dedicated coprocessors
  - Special purpose functional units (*e.g.,* AltiVec, SSE2)
- Game consoles could grow to become set-top boxes, home media and entertainment servers

# Embedded Systems

- Embedded Systems are specialized or dedicated computers used to control appliances, devices and machines
- Platforms that use embedded systems include consumer appliances, IT devices and industrial/commercial machines
- Consumer:  PDAs, game consoles, set top boxes, automotive control systems, home appliances
- IT: Printers, copiers, faxes, teller machines, telecom switches and routers, modems, videoconferencing, disk controllers
- Industrial/commercial: robotics, data acquisition, manufacturing control, process control, medical imaging and monitoring, aerospace, satellite systems, radar systems

# Growth in Embedded Devices




- Embedded devices will be pervasive
- On average, 3 embedded devices/person on the planet by 2011

*Source: Gartner 2002: Microprocessor, Microcontroller and Digital Signal Processor Forecast Through 2005*

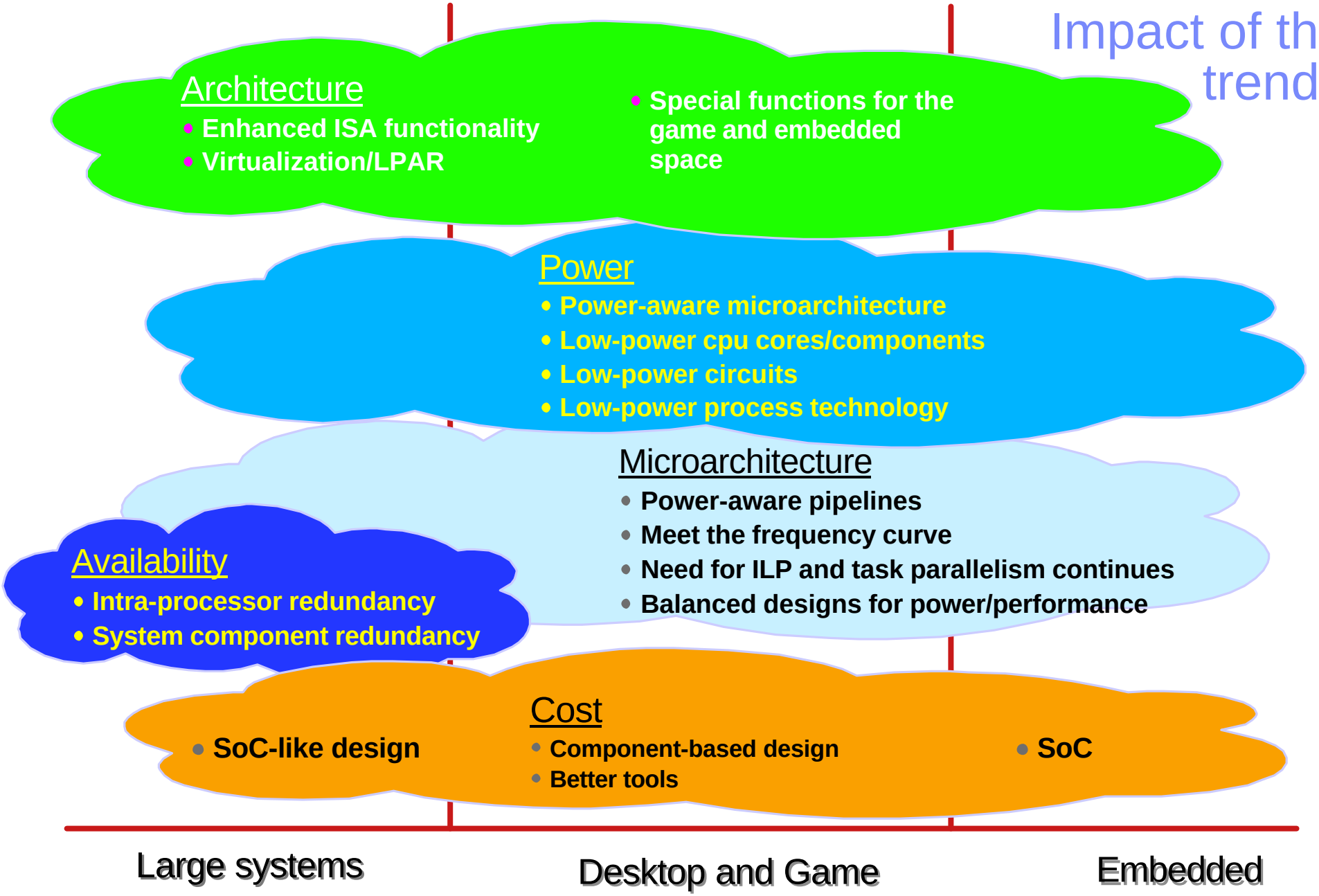
# Embedded systems characteristics

- New embedded applications are media-rich
  - *e.g.,*

# Agenda

- System and application trends
- Impact on architecture and microarchitecture
  - Architecture impact
  - Microarchitecture impact and system bottlenecks
  - Energy efficiency
  - Component-based design
  - System reliability
  - Virtualization and logical partitioning
- The Memory Wall
- Cellular architectures and IBM's Blue Gene
- Summary

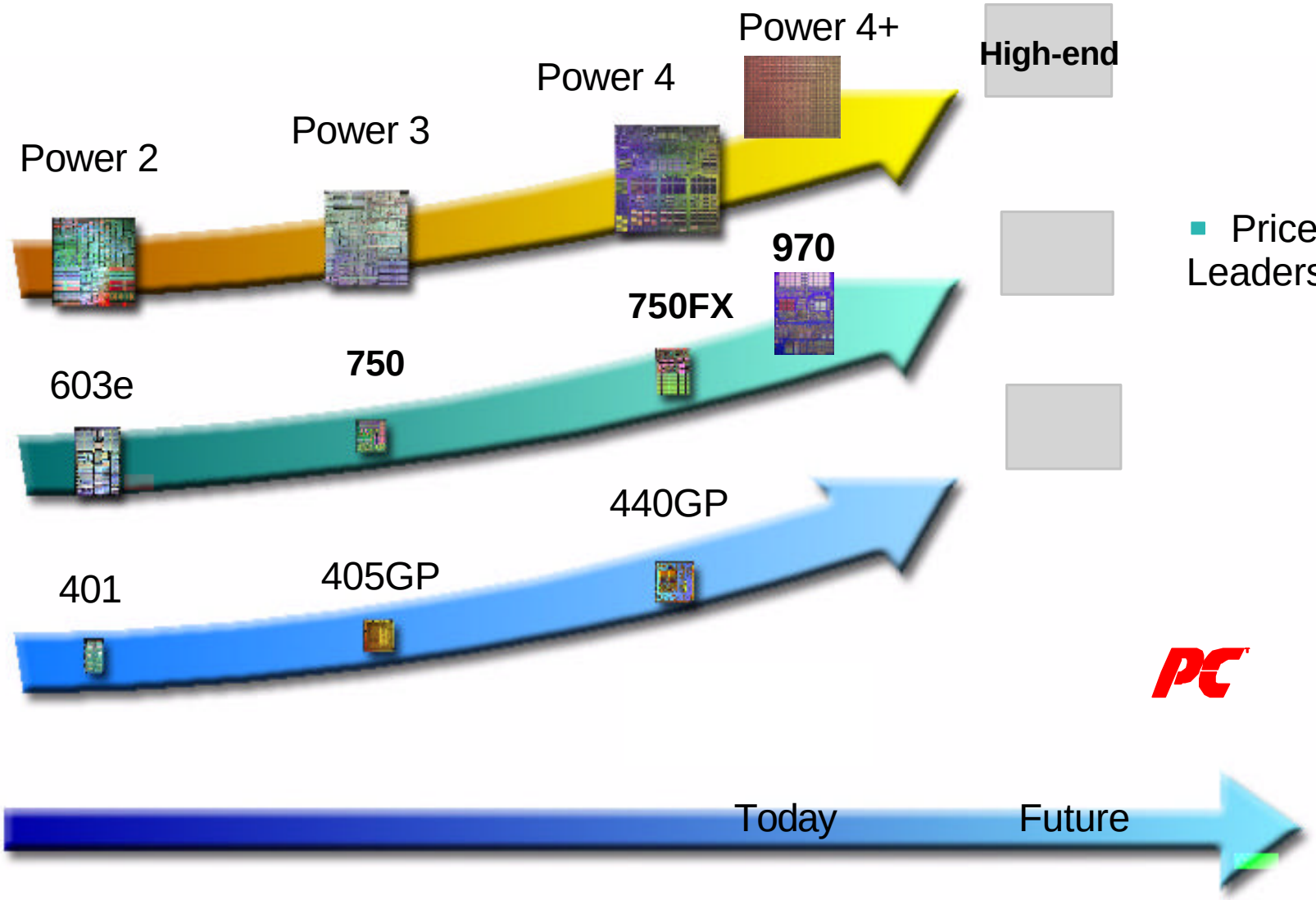
# Impact of the trends



# Architecture: ISA alone not a performance differentiator

- New applications are fairly ISA-agnostic
- But legacy applications are tightly connected to legacy ISAs
- RISC vs. CISC performance battle is over
  - Modern CISCs implemented as RISCs internally
  - Only the total system performance matters
- The new challenge is functionality
  - Virtualization/logical partitioning; SIMD/DSP extensions; power
- Clear advantage to having single ISA from high-end to low-end

# PowerPC spans the entire spectrum for new applications



# Microarchitecture: exploiting parallelism

- Emphasis on Instruction-level and task parallelism
  - High-issue rate O-O-O execution
  - Multithreading and chip MPs
  - Large shared on-chip caches (L1, L2), on-chip L3 directories
- Pressure to stay on the frequency curve
  - Deeper pipelines, with many more latches
  - Need better branch prediction and active power management
- Balancing degree of pipelining with
  - Branch predictability and data forwarding latencies
  - Power consumption
  - Cache access latency and bandwidth
- Balancing issue width with
  - Clock frequency
  - Complexity of design and verification
  - Design cost

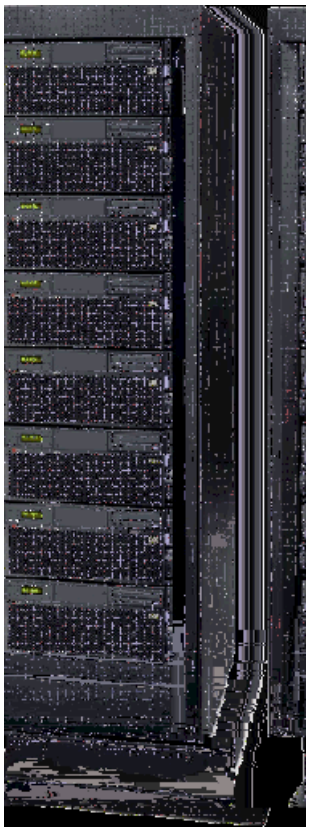
# Power-aware processors and systems

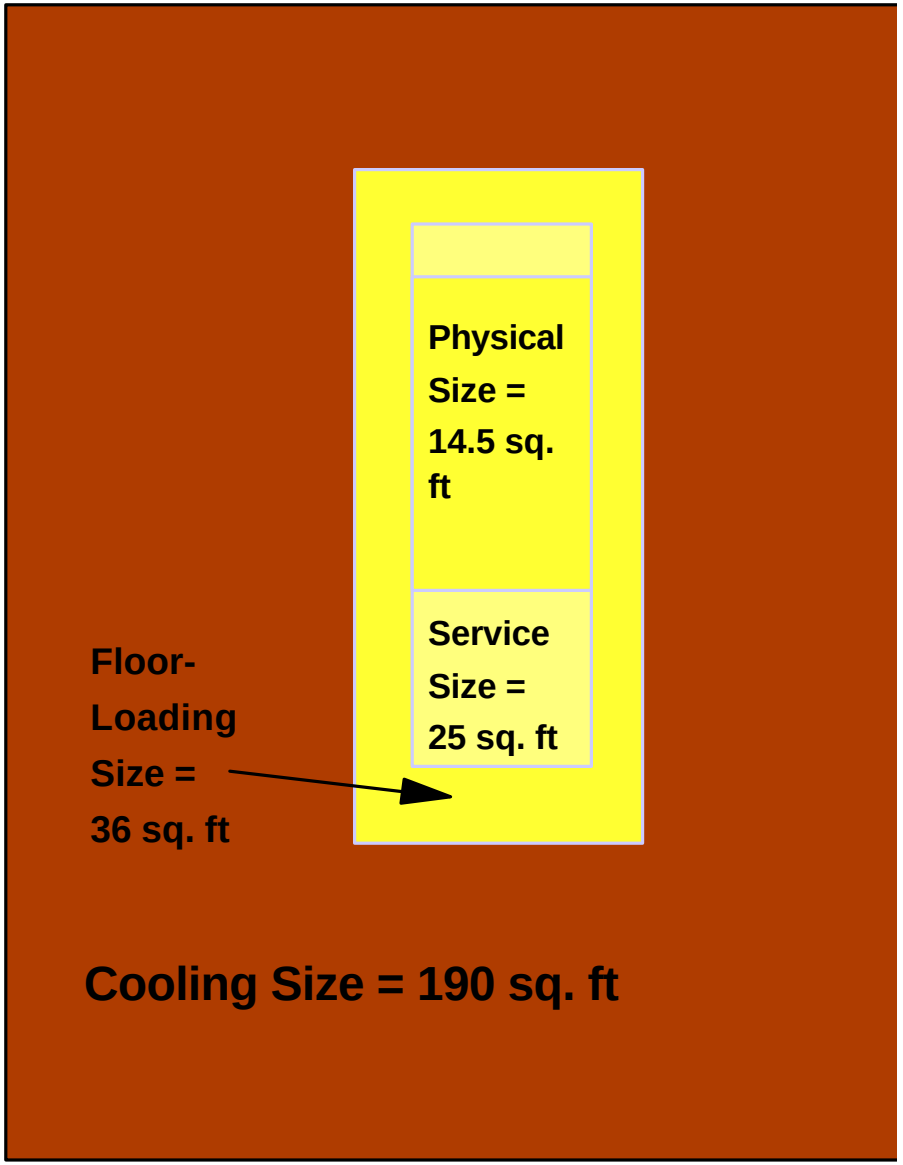
- Power-aware microarchitectures
  - Better understand pipeline depth and power consumption tradeoffs
  - Frequency, voltage scaling and clock gating
  - Eliminate redundancy and speculation to conserve energy while minimizing performance impact
- Power efficient circuits and semiconductor technologies
  - Power-efficient circuits, latches
  - Process technologies like Si-on-Insulator (SOI) offer better power/performance
  - IBM products show the advantages
    - Power4+, PowerPC405LP
- Processor power density is a hard problem
  - Processor is the hardest system component to cool
  - Uneven heat densities on the processor chip
- Software-controlled power consumption optimization
- 25% of papers, 2 tutorials in this symposium related to this issue

# Wattage affects **size**

**"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."**

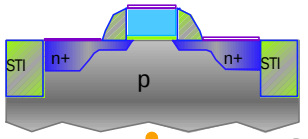
**- Eric Schmidt, CEO Google (Quoted in NY Times, 9/29/02)**



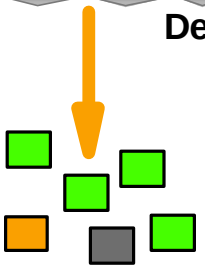


# Component-based design driven by cost and power

*System-on-chip (SoC) technology today*




Device

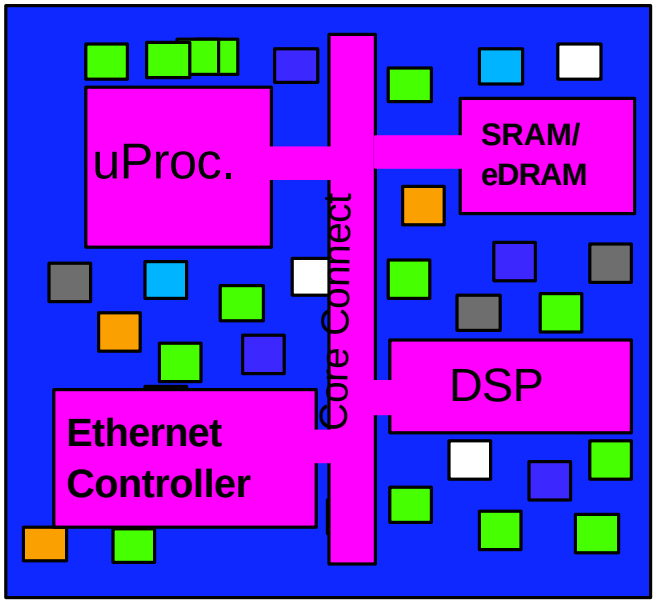


Cells

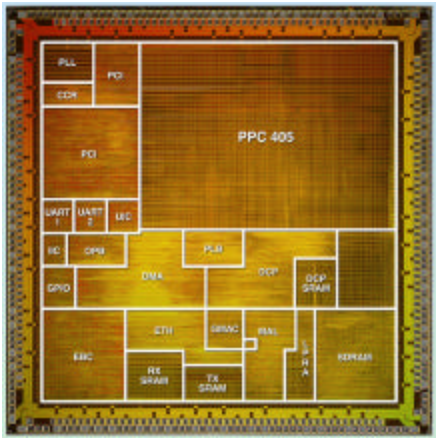
+

IP Blocks (Cores)





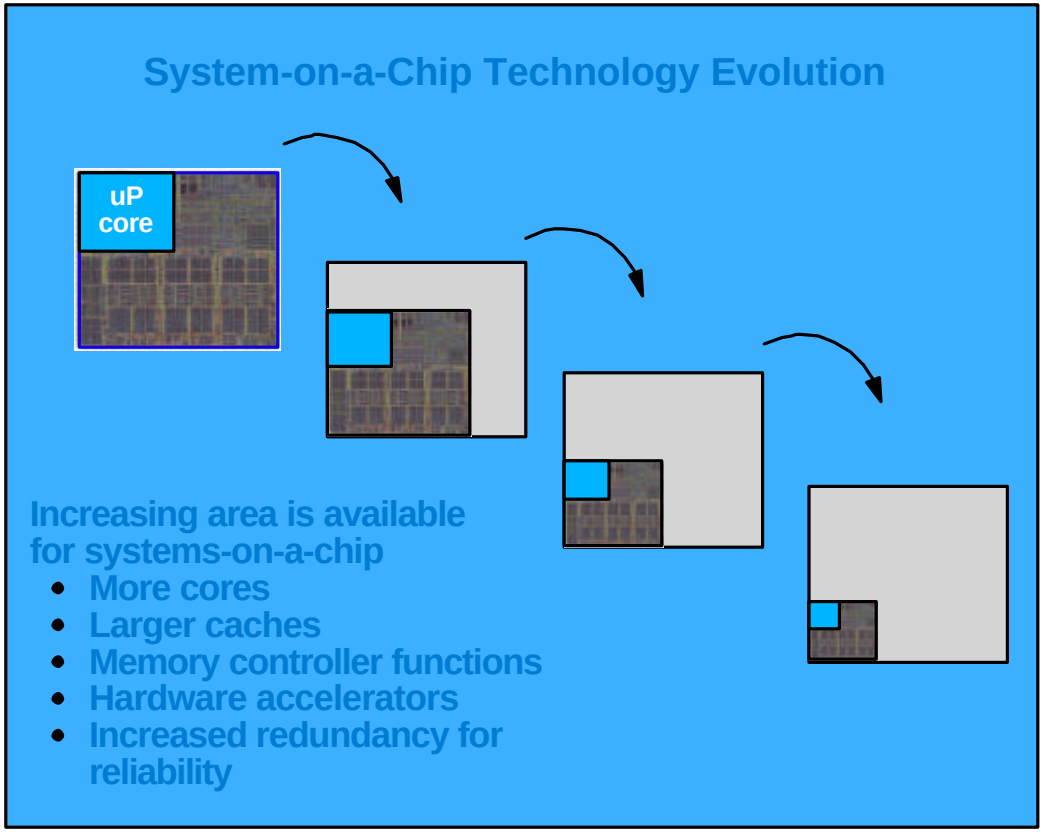

- SoC technology is still maturing
- A typical embedded SoC
  - Portable/reusable CPU cores
  - Embedded memory
  - Interfaces to the world (USB, PCI, Ethernet ***et al.***)
  - Mixed signal blocks (optional)
  - Programmable hardware (optional)
  - ROM (holds firmware/software)
  - Today: approx. 500K+ gates

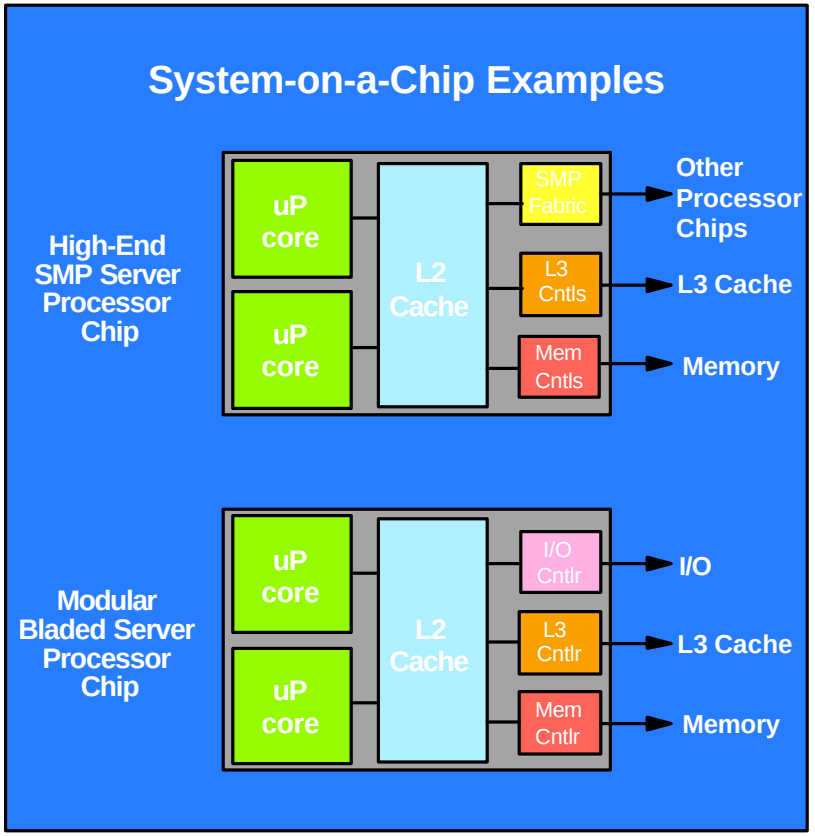



**PowerPC 405GP**

# High-end microprocessor design: an SOC-like approach

*Lower design cost, fast turnaround*



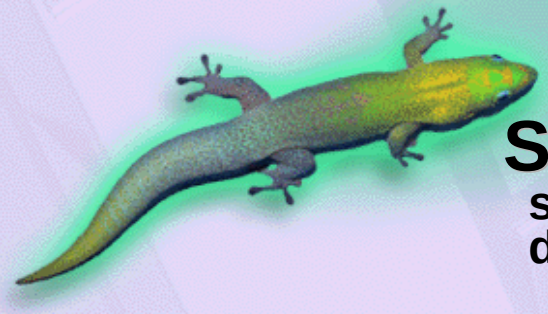



# Towards *autonomic infrastructures*

**Self-protecting**
System designed to protect itself from any unauthorized access anywhere

**Self-optimizing**
System designed to automatically manage resources to allow the servers to meet the enterprise needs in the most efficient fashion

**Self-healing**
Autonomic problem determination and resolution



**Self-configuring**
systems designed to define itself "on the fly"

# Availability

- Systems must meet higher availability standards
  - The 5-nines target: 99.999% uptime (~315 seconds/year)
- Soft error rates are going up as voltages and feature sizes are dropping
- Detection and correction of soft errors
  - Redundancy in the microarchitecture
  - ECC protection in cache and register structures
  - Microcode control for failure detection and restart
- Examples of hardware support
  - Pipeline mirroring in IBM mainframe processors
- High availability mechanisms are essential for an autonomic infrastructure

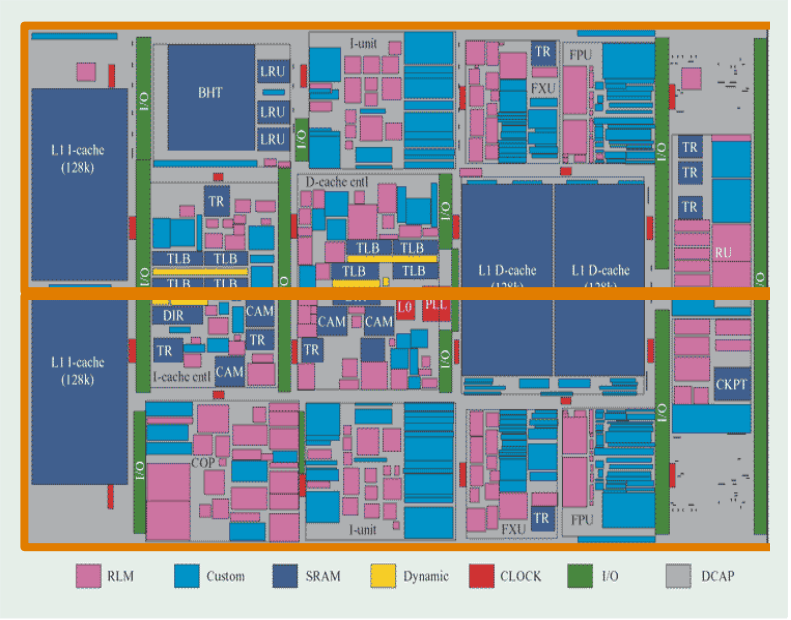



Figure 1

Floorplan of z900 microprocessor.

**Pipeline mirroring in IBM z900 mainframe processor**

# Research opportunity: Availability

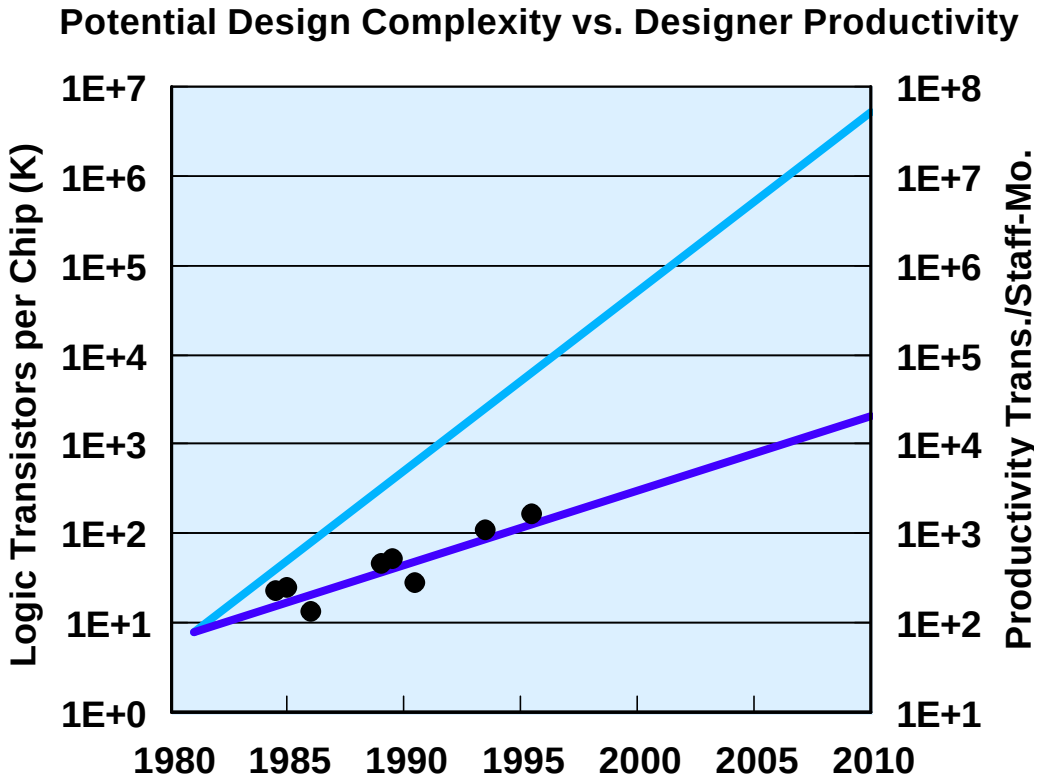
- Availability must be handled as a system issue
- Software
  - Front-end apps (edge-server applications)
  - Back-end apps (***e.g.*** databases)
  - OS, middleware
- Hardware
  - Network connectivity
  - Motherboard and peripherals
  - Package
  - Coherence structures
  - Memory hierarchy, buses, I/O
  - Processor

# Virtualization and logical partitioning

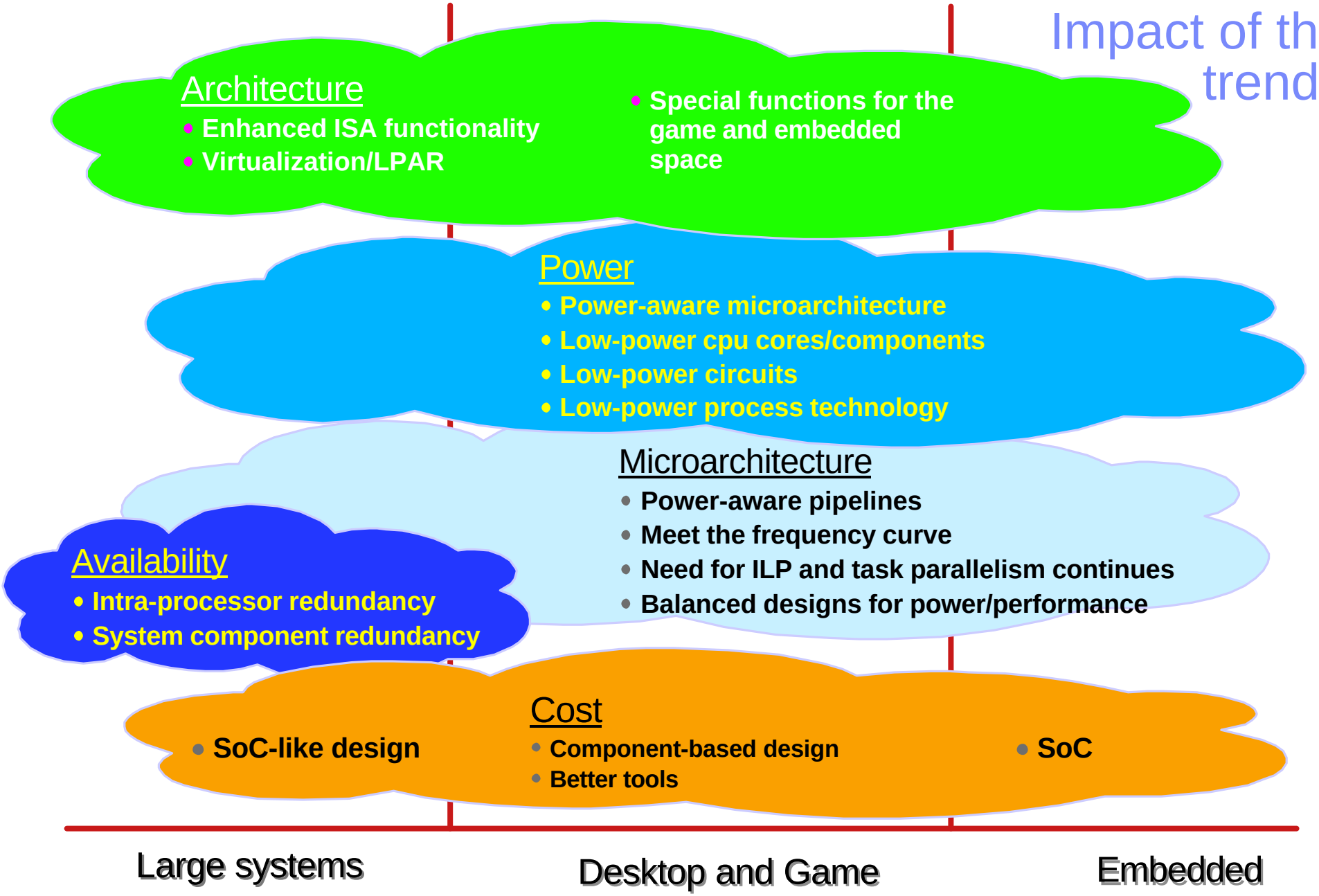
- Architectural support for virtualization
  - Distinguishing between user/supervisor/hypervisor state
  - Hypervisor mode instructions to create and protect logical partitions or full virtual machines
  - There could be ISA obstacles to full virtualization
- This is hard stuff
  - S/360 has been a pioneer since 1968 with the VM environment
  - Success of Linux on mainframes: 100s of Linux's in a box
  - LPAR on Power4: simple addition to the address translation hardware efficiently implements dynamic LPAR
- Significant cost and utilization advantage for customers
- Huge research opportunity
  - Improving performance of virtual machines via dynamic adaptation
  - Designing streamlined architectural support

# Design complexity

- Processor design complexity is increasing
- Designer productivity cannot keep up
- Time-to-market is critical
  - Design and deliver within fixed time budget
- Better, more intelligent design tools and automation methodologies are necessary
- Extensive work is ongoing at IBM Research to solve problems in this area
  - System-level design tools
  - Behavioral synthesis
  - Logic synthesis
  - Circuit tuning
  - Place/route, placement-driven synthesis
  - Circuit analysis/extraction
  - Manufacturing enhancement



# Impact of the trends



# Agenda

- System and workload trends
- Impact on architecture and microarchitecture
- The Memory Wall
- Cellular architectures and IBM's Blue Gene
- Summary

# The Memory Wall
## A prehistoric problem that won't go away anytime soon!

- 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
- Typical on-chip L2 cache performance degradation now more than 2*x* the ideal (*i.e.* perfect) cache
- For multiprocessors, inter-cache latencies increase this degradation to 3*x* or more for 4 processors and up

# Some approaches to attack the Memory Wall

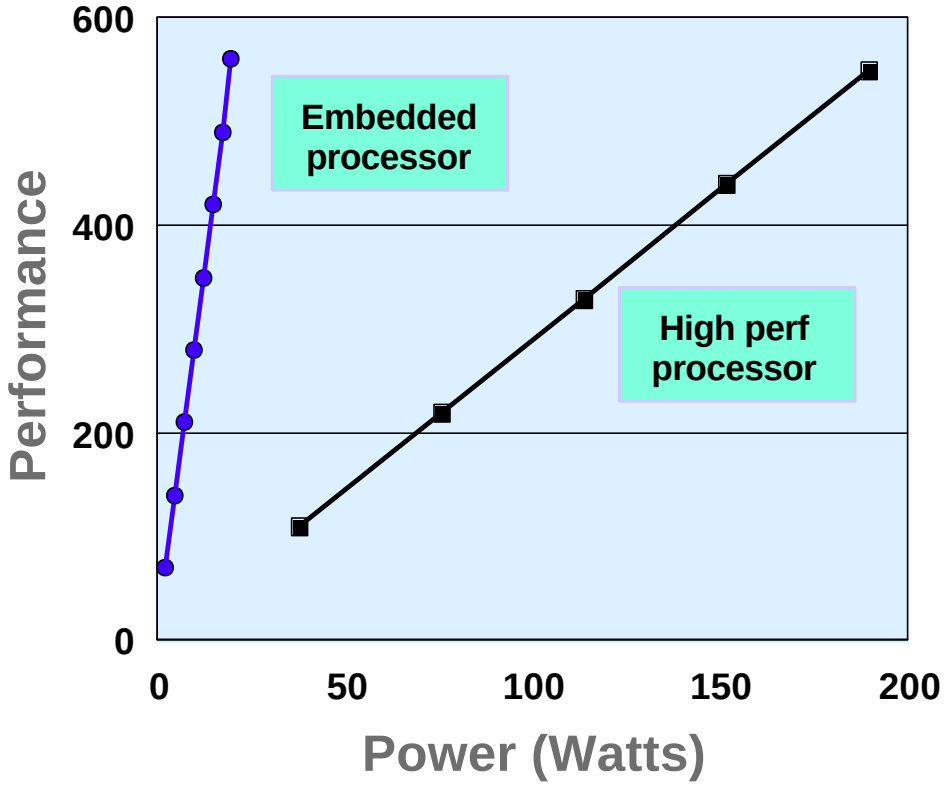
- Traditional techniques
  - Larger caches, deeper cache structures
  - Latency hiding via prefetching (h/w, s/w, both)
  - Compilers/application software cognizant of the memory hierarchy
  - Hardware multithreading
- Emerging research opportunities
  - Reduced intercache/scaling effects via affinity scheduling of tasks
  - Machine learning applied to code prefetching and code pre-positioning
  - Self-optimizing cooperation between the hardware and software directives
- New computing paradigms with programming models designed to better tolerate memory latency

# Agenda

- System and workload trends
- Impact on architecture and microarchitecture
- The Memory Wall
- Cellular architectures and IBM's Blue Gene
- Summary

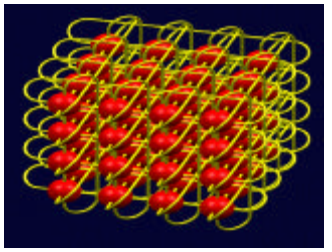
# Building cellular architectures with embedded processors

- Compares favorably to conventional systems
  - Power efficiency 10-50x better
  - Cost/performance 10x better
- Cost is drastically lower than conventional systems
- Exploitation of high redundancy helps availability




**Web Serving performance: pages served (MB/s)**

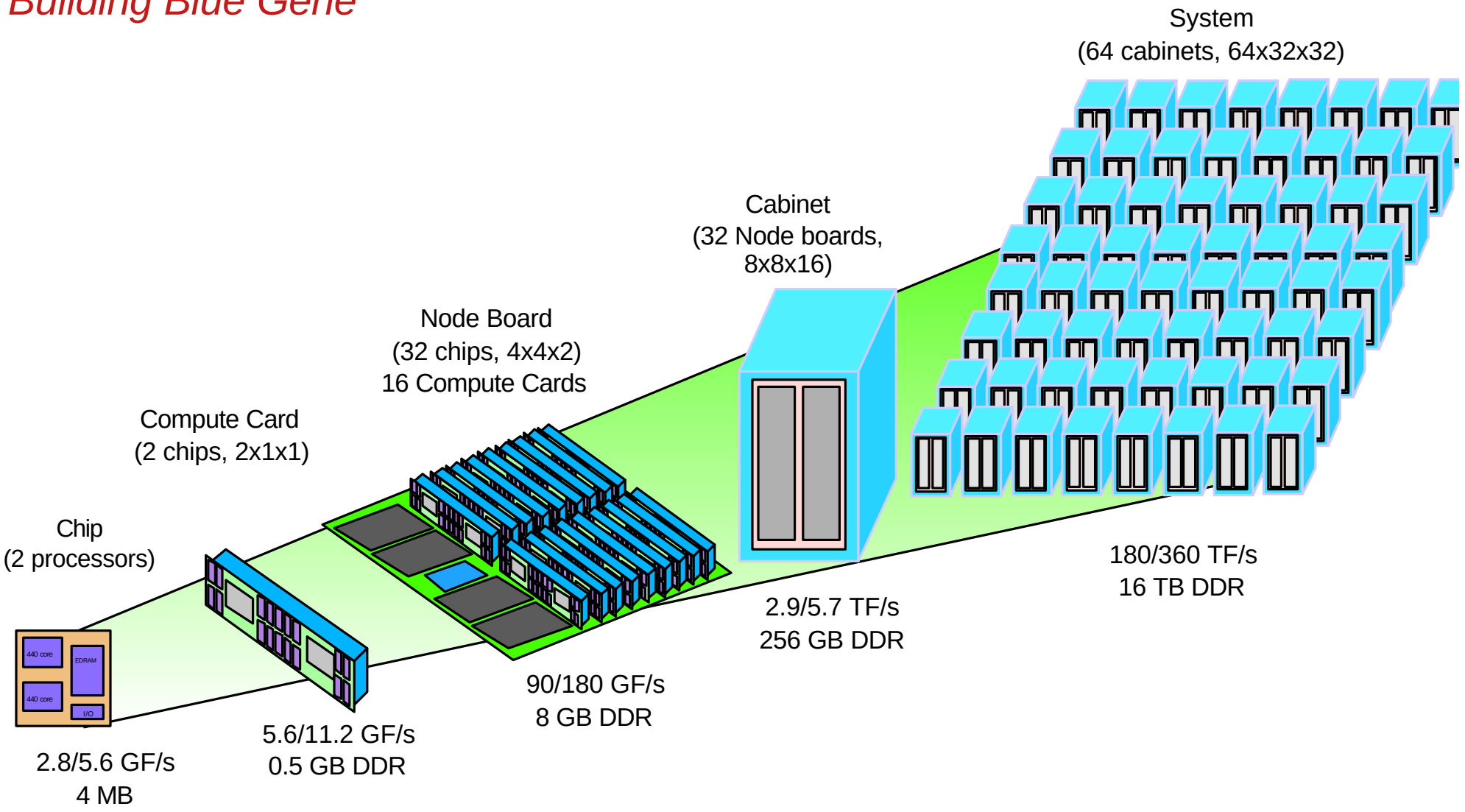
# IBM's Blue Gene



- Advance the state of the art in computer design and software for extremely large scale systems
- Blue Gene is a cellular architecture
  - A homogeneous collection of simple independent processing units called **cells**, each with its own system image
  - All cells have the same computational and communications capabilities (interchangeable from OS or application view)
  - Integrated connection hardware provides a straightforward path to scalable systems with thousands/millions of cells
- Future blade systems could be cellular architectures
  - Low-cost, high-performance, better power characteristics
  - For many new high-growth apps

# Cellular architectures user component-based design

*Building Blue Gene*



# Cellular architectures offer high levels of availability

- We expect transient errors to be 10 times more common than permanent errors (1 per day *vs.* 1 in 10 days)
- Blue Gene supports semiautomatic, coordinated checkpointing
- Application re-launched on same partition after loading data from previous checkpoint - handles transient errors
- Approach insufficient for dealing with undetected soft errors - may occur once/month for machine with 64K nodes

# Agenda

- System and workload trends
- Impact on architecture and microarchitecture
- The Memory Wall
- Cellular architectures and IBM's Blue Gene
- Summary

# Summary

- Exciting opportunities for microarchitecture, architecture, and design
- Integrated system design approach is key to performance, functionality, and development cost
  - Memory Wall
- Architects must focus on enhanced functionality
  - Virtualization
  - Power awareness
  - Autonomic computing support
- Low design cost and complexity demand component-based designs
- Computation model shift in new applications
  - Leverage high-volume embedded processors

This talk is dedicated to John Cocke, the father of Reduced Instruction-Set Computing (RISC).