Data Centric Systems
The Next Paradigm in Computing

Dr. Tilak Agerwala
VP, Data Centric Systems, IBM Research

Dr. Michael Perrone, *Research Staff Member*

SP Design Principles & Impact

STOCKPILE STEWARDSHIP

Principle 1: "Ride the technology curve"

Principle 2: Time-to-market

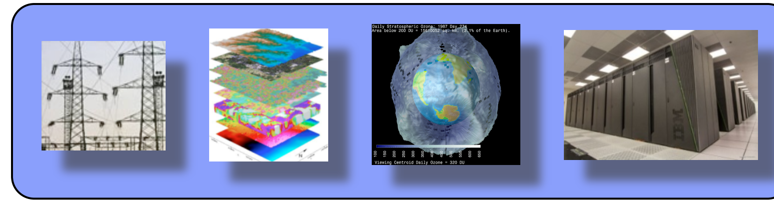
Principle 3: Communication is critical

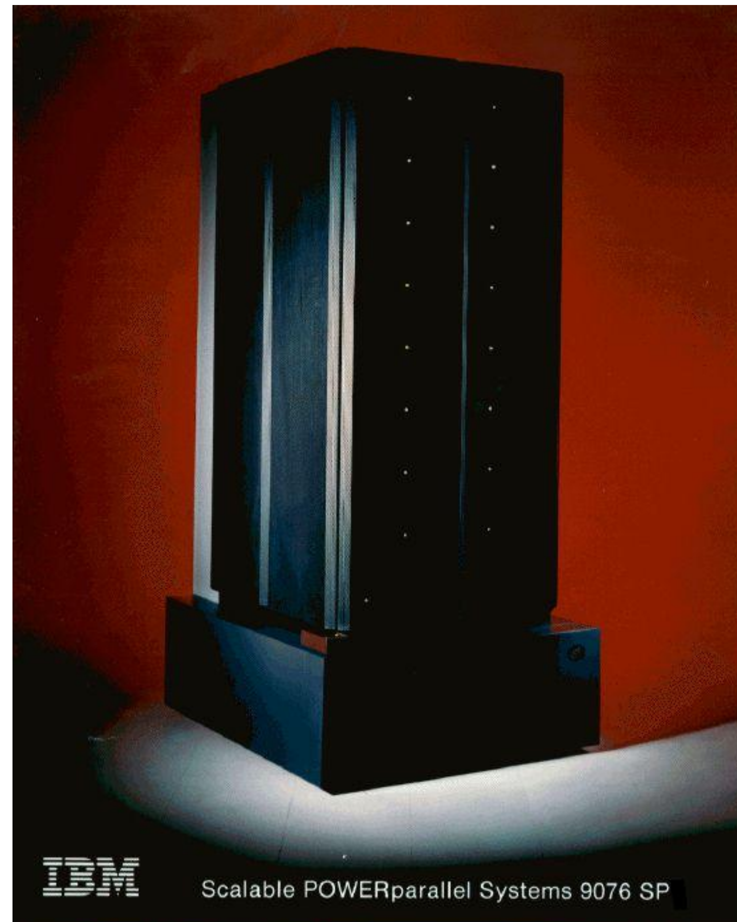
Principle 4: Standard UNIX

Principle 5: High-performance services

Principle 6: High Availability

Principle 7: Single-System Image Flexibility





- **Government**
  - **Science Based Stockpile Stewardship** (SBSS, 1994)
  - Dramatic new level of simulation accuracy
- **Industry**
  - Drove parallel database adaption: DB2, SAP, Oracle
  - Aerospace, Automotive, Chemistry, Database, Electronics, Finance, Geophysics, Information Processing, Manufacturing, Mechanics, Pharmaceuticals, Telecom, Transportation, etc.

The Motivation for Parallelism: Power Savings

Amdahl's Law

Total time:

$$T = T_{Serial} + T_{Parallel}$$

Speed up factor:

$$\left(T_{Serial} + \frac{T_{Parallel}}{N} \right)^{-1} T$$

Acceleration by frequency scaling

$$P = CV^2 f \longrightarrow P = cf^{\alpha} \qquad \alpha > 2$$

Acceleration by parallelism

$$P = NP_0$$

If the parallel section is large enough,
it is more power efficient to use parallelism.

Blue Gene Design Principles – Optimized for power efficiency

Principle 1: Trade clock speed for lower power consumption

Principle 2: Use integration to lower power

Principle 3: Focus on network performance

Principle 4: Reduce OS jitter

Principle 5: Application and hardware Co-Design

AWARDS
- Top500
- Green500
- Graph500

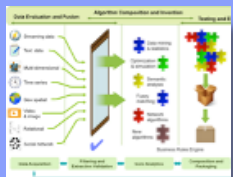
SEQUOIA

Data-Centric Systems: Application Domains

IBM

Complex analytics — HPA

Business Analytics



System G
Big Insights

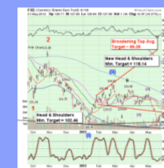
Business Intelligence

Watson

Social Analytics

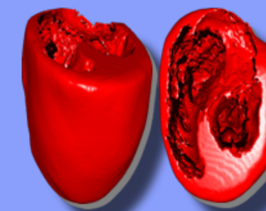
DOD All-Source
Intelligence

Financial Analytics



Integrated Trading
and VaR

Life Sciences



Health Care
Analytics

Complex analytics

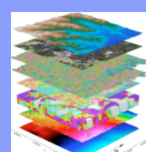
HPA

Modeling, Simulation — HPC

Technical Computing

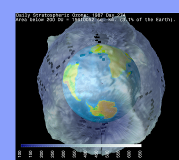
Engineering Design,
Prototyping, Analysis,
Optimization

Oil and Gas



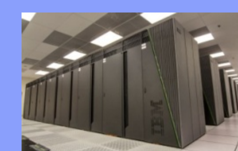
Integrated Wide
Azimuth Imaging
and Interpretation

Climate & Environment



Production
Weather

Science



DOE NNSA and
Office of Science

Modeling, Simulation

HPC

DCS = HPC + HPA = HPE (High Performance Environments)

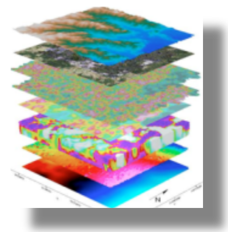
Key Domain Characteristics: Big Data, Complex Analytics, Scale and Time to Solution Requirements
Overlapping Requirements in HPC and HPA enable an converged solution

DCS Workflows: Mixed compute capabilities required

Analytics Capability

- Complex code
- Data Dependent Code Paths / Computation
- Lots of indirection / pointer chasing
- Often Memory System Latency Dependent
- C++ templated codes
- Limited opportunity for vectorization
- Limited scalability
- Limited threading opportunity

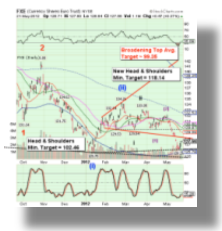
Oil and Gas



4-40 Racks

Reservoir ← → Seismic

Financial Analytics



2-20 Racks

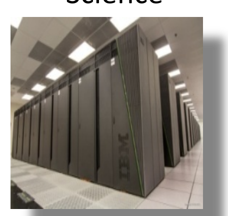
Analytics ← → Value At Risk

All Source Analytics

1-5 Racks

Graph Analytics ← → Image Analysis

Science



Labs: 100-400 Racks
Univ: 1-20 Racks

Throughput ← → Capability

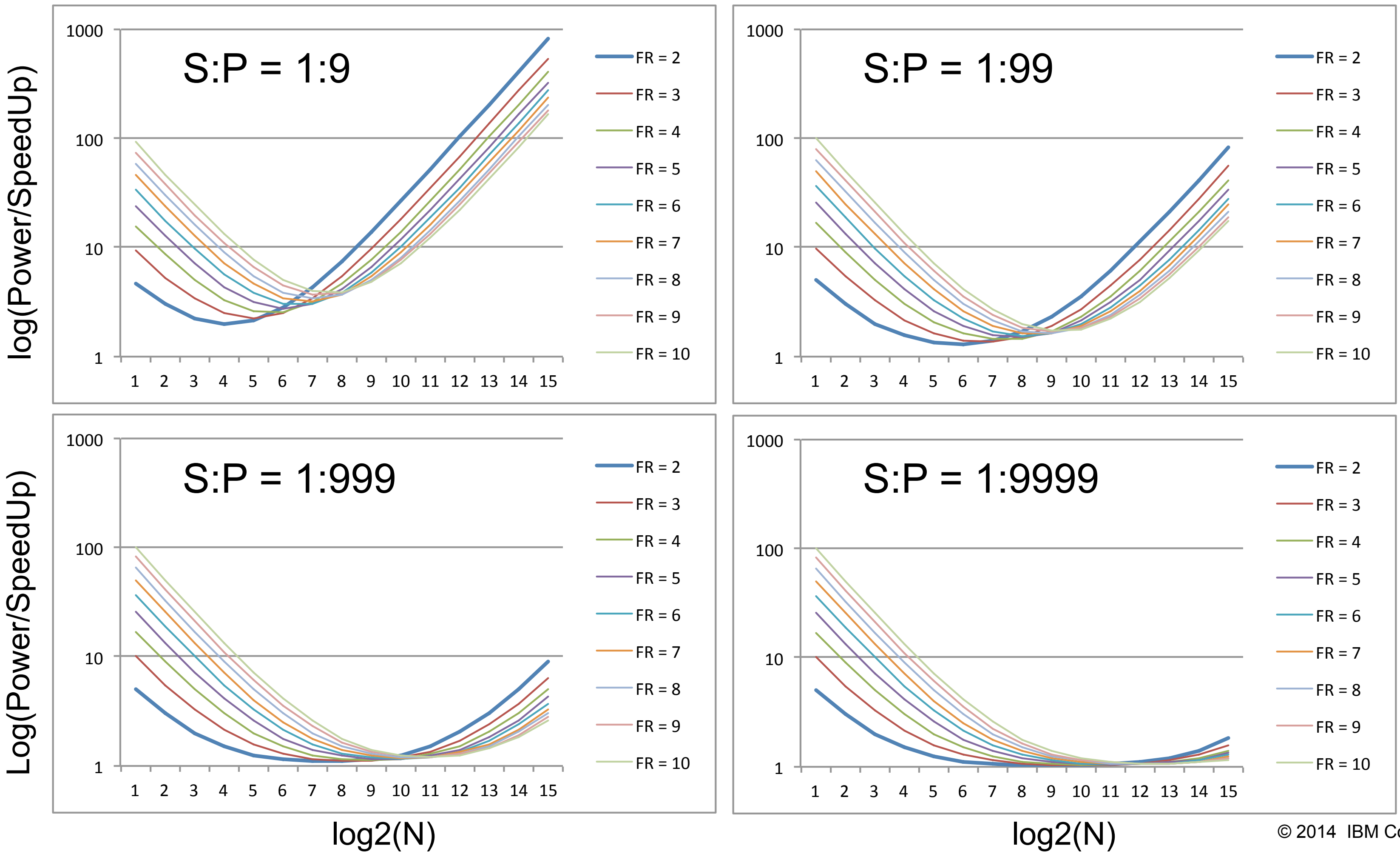
Massively Parallel Compute Capability

- Simple kernels
- Ops dominated (e.g. DGEMM, Linpack)
- Simple data access patterns
- Can be preplanned for high performance

Heterogeneity Is Important: Power Per Unit Speed Up Factor

- Optimal system design depends on frequencies and Serial/Parallel (S:P) split
- Today static – Tomorrow dynamic

N = # of weak cores / # of strong cores
FR = Strong core frequency / Weak core frequency



IBM Data-Centric Design Principles

Principle 1: Minimize data motion

- Data motion is expensive
- Hardware and software to support & enable compute in data
- Allow workloads to run where they run best

Principle 2: Enable compute in all levels of the systems hierarchy

- Introduce "active" system elements, including network, memory, storage, etc.
- HW & SW innovations to support / enable compute in data

Principle 3: Modularity

- Balanced, composable architecture for Big Data analytics, modeling and simulation
- Modular and upgradeable design, scalable from sub rack to 100's of racks

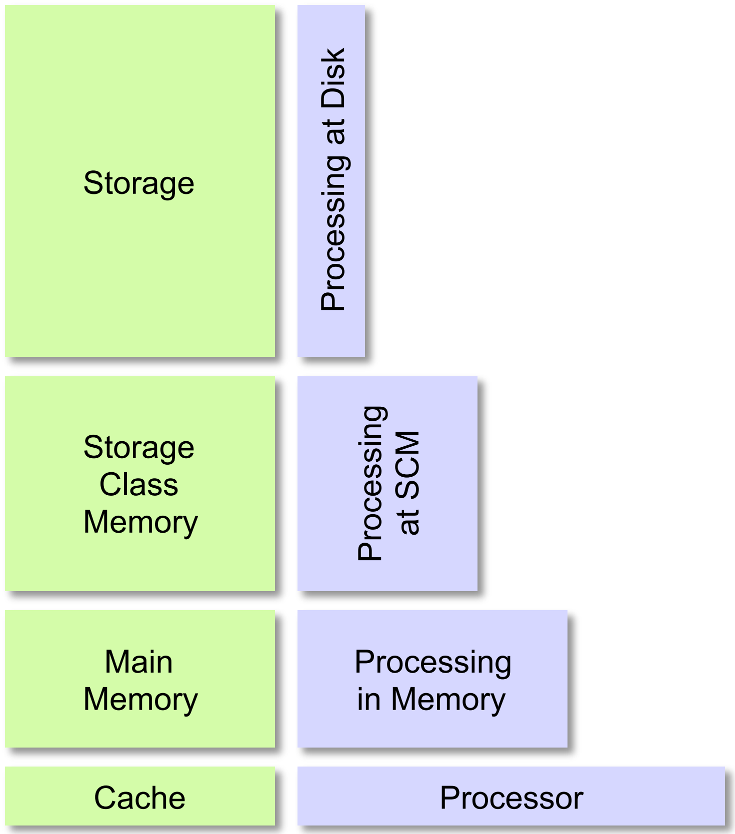
Principle 4: Application-driven design

- Use real workloads/workflows to drive design points
- Co-design for customer value

Principle 5: Leverage OpenPOWER to accelerate innovation and broaden diversity for clients

Data CentricSystems – Systems Built Around Data

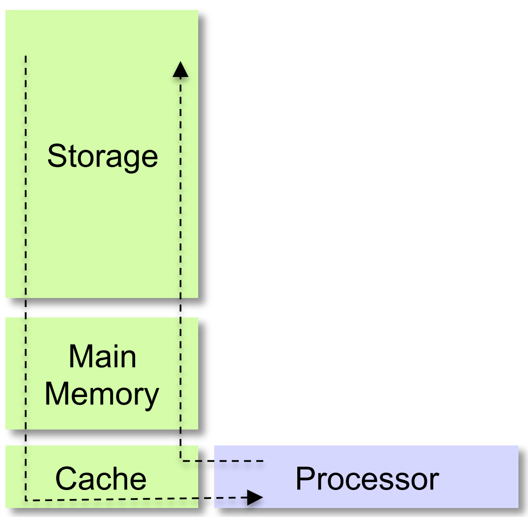
- Integration of massive data management and compute with complex analytics
- Optimized workflow components (compute and dataflow) across the system
- Data centric systems move computation to the data

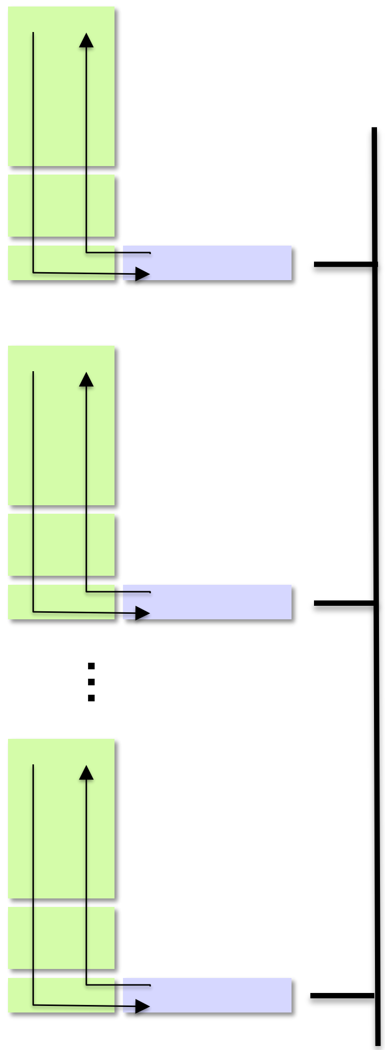


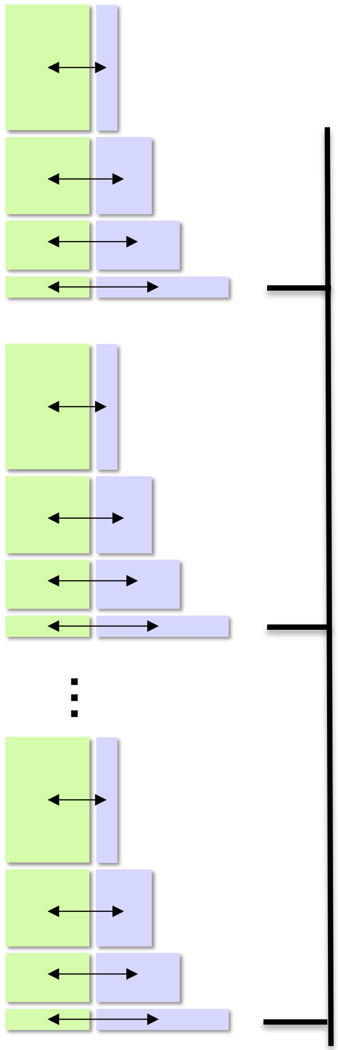
Data-Centric Computing

Data Centric System Design: Addresses Latency!

Traditional Computing	Si Tech + Parallelism	Data Centric Computing
Silicon Technology, Frequency Scaling	Amdahls Law, Density Scaling	Si Tech + Parallel + Systems







OpenPOWER Foundation

MISSION: The OpenPOWER Consortium's mission is to **create an open ecosystem**, using the POWER Architecture to share expertise, investment and validated and compliant server-class IP **to serve the evolving needs of customers**.

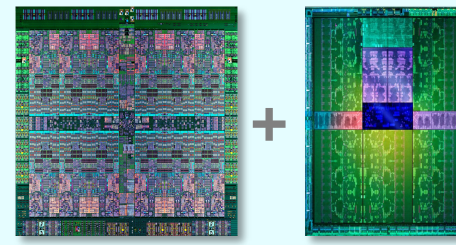
– Opening the architecture to give the industry the ability to innovate across the full Hardware and Software stack
  - Includes SOC design, Bus Specifications, Reference Designs, FW OS and Hypervisor Open Source

– Driving an expansion of enterprise class Hardware and Software stack for the data center

– Building a vibrant and mutually beneficial ecosystem for POWER

Example:

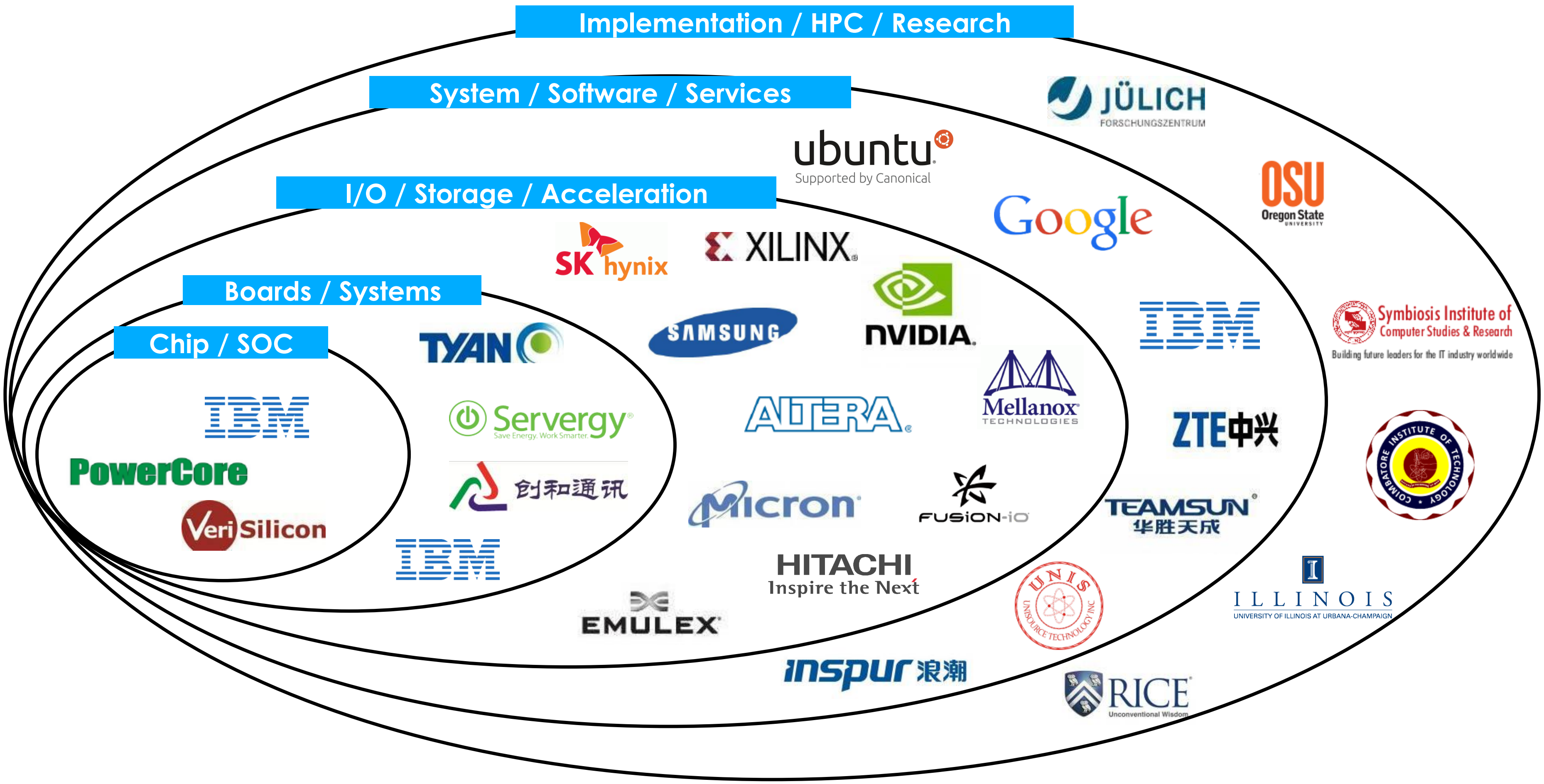
POWER CPU + Tesla GPU



Platinum Members

ALTERA	Google	IBM	Mellanox Technologies
Micron	NVIDIA	PowerCore	SAMSUNG
TYAN	ubuntu Supported by Canonical	9 Gold Members	16 Silver Members

Building collaboration and innovation at all levels



Welcoming new members in all areas of the ecosystem

100+ inquiries and numerous active dialogues underway

35 members and groing

Data Centric Systems: Activities

- **Co-design**
  - Optimize system capability, trading off within constraints, e.g., power, cost, etc.
  - Arrive at system design points that are driven by real workflows
- **System Architecture**
  - Heterogeneous nodes and memory, e.g., near-memory processing, accelerators, etc.
  - Active Communications / Processing-in-Network to reduce software path length and data movement
  - Active Storage: Low latency storage model for working set and efficient check pointing
  - Continuous workload rebalancing and optimization
- **Resilience**
- **System-wide power management**
- **Software**
- **Performance**

Power Efficiency

- **Need significant improvement over what we can get from technology alone**

- **Workflow efficiency**
  - Remapping workflows to data centric elements
  - Data motion is expensive
  - Cost/Performance benefits

- **Architectural efficiency**
  - Increase workflow parallelism to leverage low-power cores

- **Engineering efficiency**
  - Improved dynamic power management
    - Power only what's being used
    - Vary voltage dynamically
  - Minimize power losses
    - New power device technology, power conversion techniques and dense packaging

      E.g., Reduce electrical current conversion loss from 30% (today) to 10% (future)

Resilience

- **Need 10-100x improvement in fault resilience**

- **Fault detection**
  – Expose all hardware faults
    - Spend more transistors on error detection
    - “Silent errors” – e.g., Cosmic ray in a multiplier is expensive to protect against

- **Fault handling options**
  – Hardware faults recover in hardware (e.g., Error Correction Code)
  – Recover in software
    - e.g., reset to a previous checkpoint
  – Identify “don't care” states
    - <1:10 of the time data was not used an fault was irrelevant
    - E.g., unused portions of cachelines & pages; stale variables, etc.

Systems Software Stack

- **Workflow driven data-centric execution model**
  - Computation occurring at different levels of the memory and storage hierarchy
  - Compute, data and communication equal partners
  - Late binding to heterogeneous hardware element
  - Dynamic optimization: Increasingly automated and self-optimizing
  - Hardware support for productivity

- **Programming model**
  - Encompass all aspects of the data and computation management
  - Enable new system functionality while minimizing the impact on programmers MPI, OpenMP and OpenACC extensions
  - Co-existence with lower level programming models

Some Research Areas

- **SYSTEMS**
  - Consistent formal data/system/execution objects & abstractions for efficient reasoning about the system
  - Systems API's for Power Management, Active networks, Active storage, Active memory, Continuous workload rebalancing and optimization

- **PROGRAMMING MODELS AND RUNTIMES**
  - Heterogeneous massively multithreaded model
    - Enable peer-to-peer heterogeneous distributed compute
    - Late binding of 100's of millions of threads on millions of elements
    - Dynamic management of time-varying ensembles of workloads

- **RESILIENCE**
  - Full transparency and instrumentation to handle software errors
  - Anomalous pattern detection
  - API's for Resilience

The Future

- **A time of significant disruption** – industries are digitizing aggressively - Data is emerging as the "critical" natural resource of this century.
- **Data is joining theory, practice and computation** to drive discovery in research and industrial / commercial impact.
  - Integrating compute with data from multiple sources will drive enormous innovation over the next decade!
  - We must address the data explosion and make efficient data management our number one design parameter
- **The Era of Cognitive Supercomputers**
  - Quantify the uncertainty associated with the behavior of complex systems-of-systems and predict outcomes
  - Learn and refine underlying models based on constant monitoring and past outcomes
  - Accommodate "what if" questions in real-time
  - Provide real-time interactive visualization