Present and Future Complex Systems:
Adaptation of Applications, System Software, and Hardware Technologies

Dr. B. B. Prahlada Rao,
Bindhu Madhav, Mangala

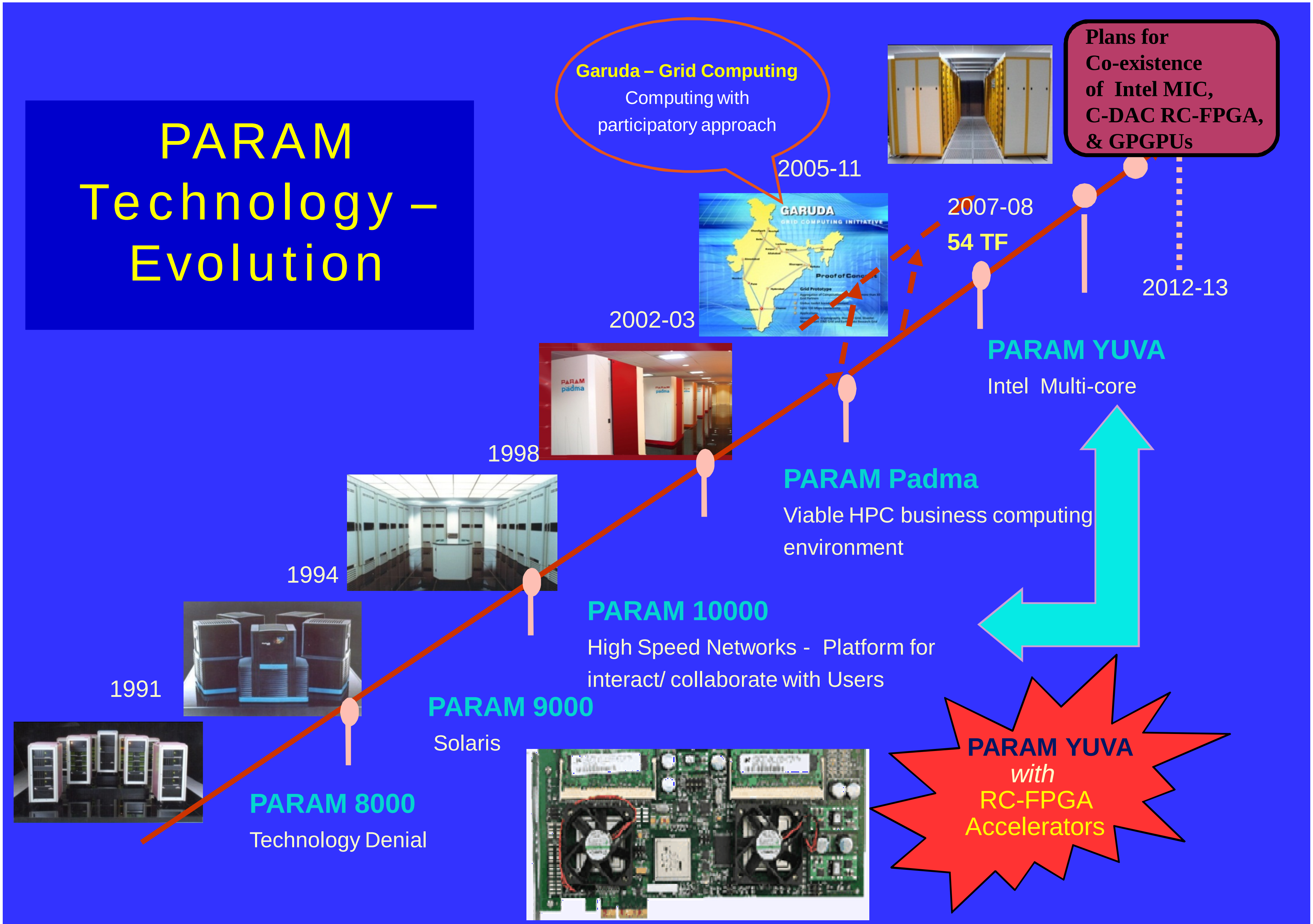
Joint Director– SSDG
prahladab@cdac.in

Centre for Development of Advanced Computing
C-DAC Knowledge Park, Bangalore

Presentation Outline

- C-DAC Current Research Areas
  - Grid Computing: GARUDA
  - Cloud Computing: C-DAC Scientific Cloud
  - Multicore-Accelerators
  - Mobile Computing: AR based mLearning
  - Ubiquitous Computing
- New Application Scenarios
  - Petascale Computing
  - Hybrid Computing
- C-DAC Future Complex Systems
  - Exascale Computing
  - Internet of Things: Mobile+WSN
  - Sensor Clouds
  - Large Complex Systems: Energy Internet

PARAM Technology – Evolution



1991

PARAM 8000

Technology Denial

PARAM 9000

Solaris

1994

PARAM 10000

High Speed Networks - Platform for interact/ collaborate with Users

1998

PARAM Padma

Viable HPC business computing environment

2002-03

2005-11

Garuda – Grid Computing

Computing with participatory approach

2007-08

54 TF

PARAM YUVA

Intel Multi-core

2012-13

Plans for Co-existence of Intel MIC, C-DAC RC-FPGA, & GPGPUs

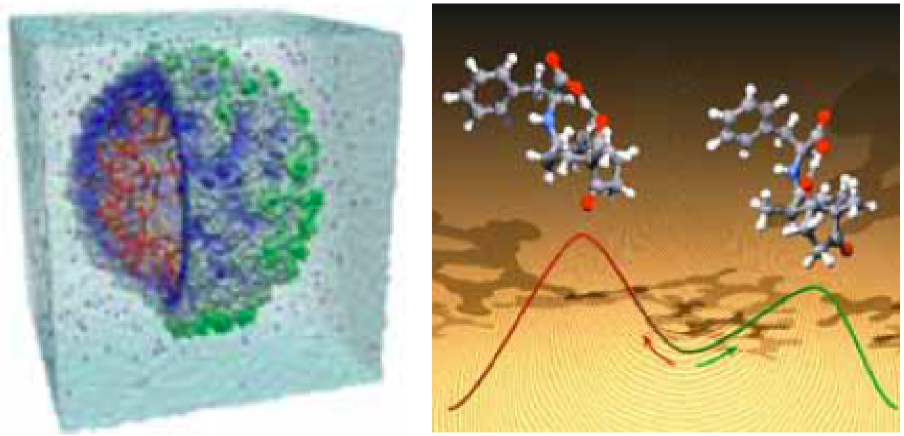
PARAM YUVA *with* RC-FPGA Accelerators

Need for moving ahead in HPC

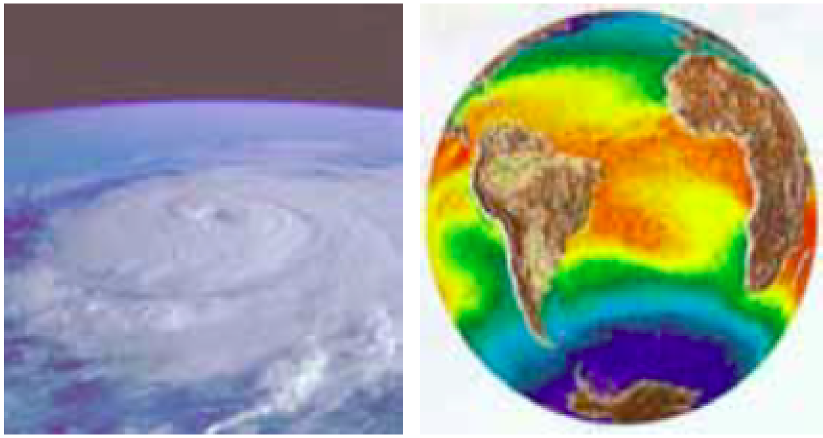
- Access to state of the art HPC is essential for international competiveness in science and engineering
- A world-class computing service will enable:
  - To solve problems today that could otherwise only be solved in 3-5 years
  - To cooperate and compete with the Nations leading in HPC
- Providing competitive HPC services and HPC enablement to the Scientific and Engineering community from National labs and academics is an important aspect and the engagement in this direction need to be continuous

Petascale computing will enable advances in a broad range of science and engineering disciplines:

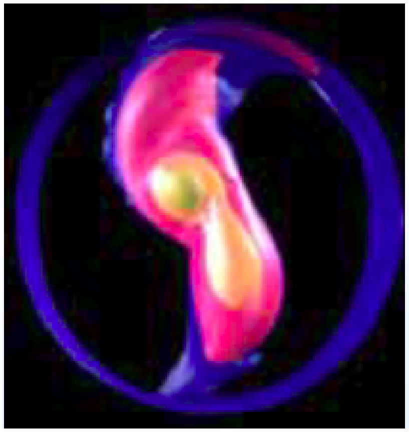
Molecular Science



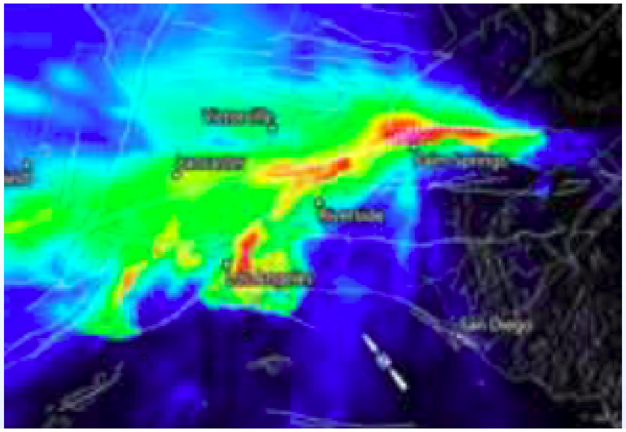
Weather & Climate Forecasting



Astronomy

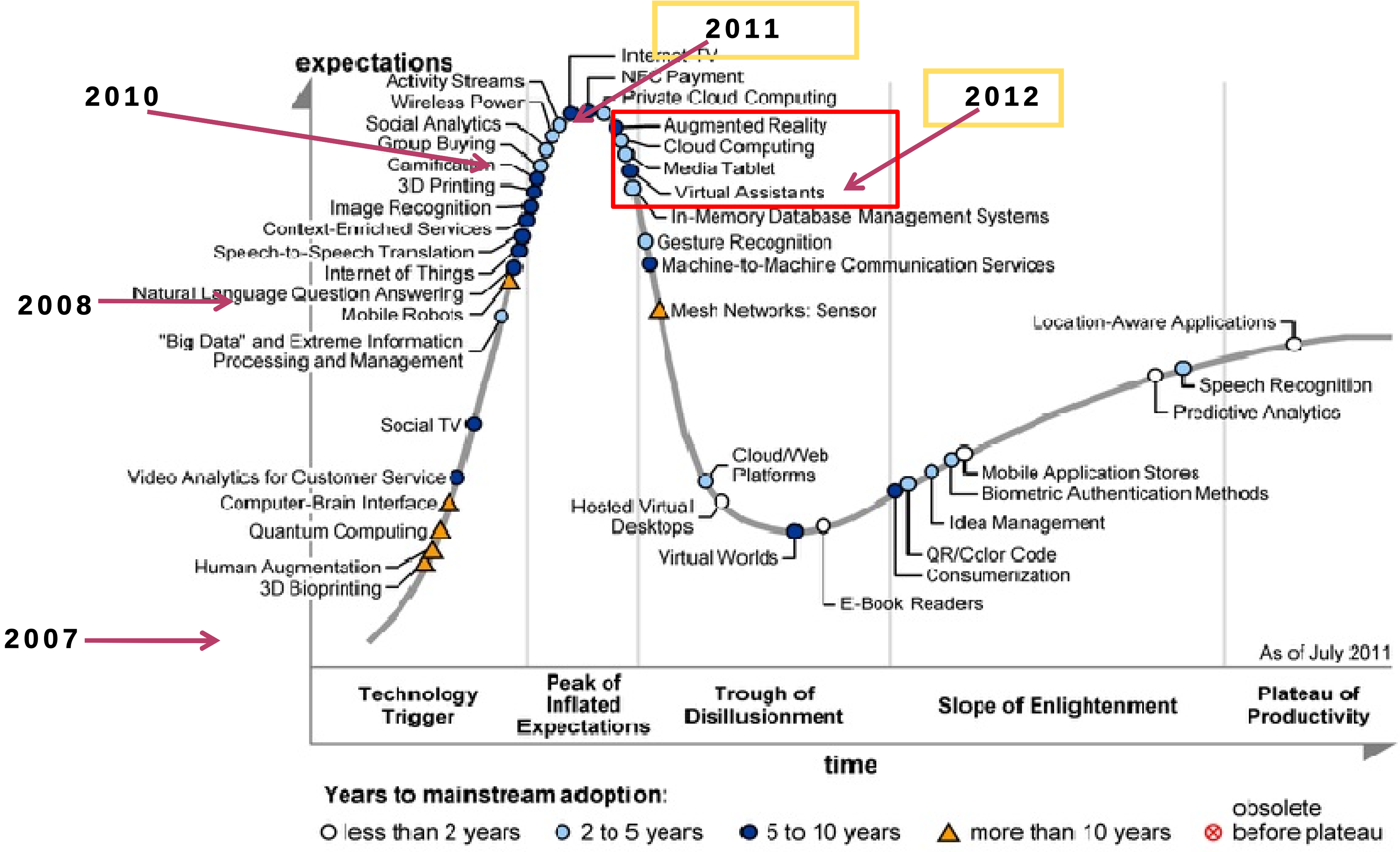


Earth Science

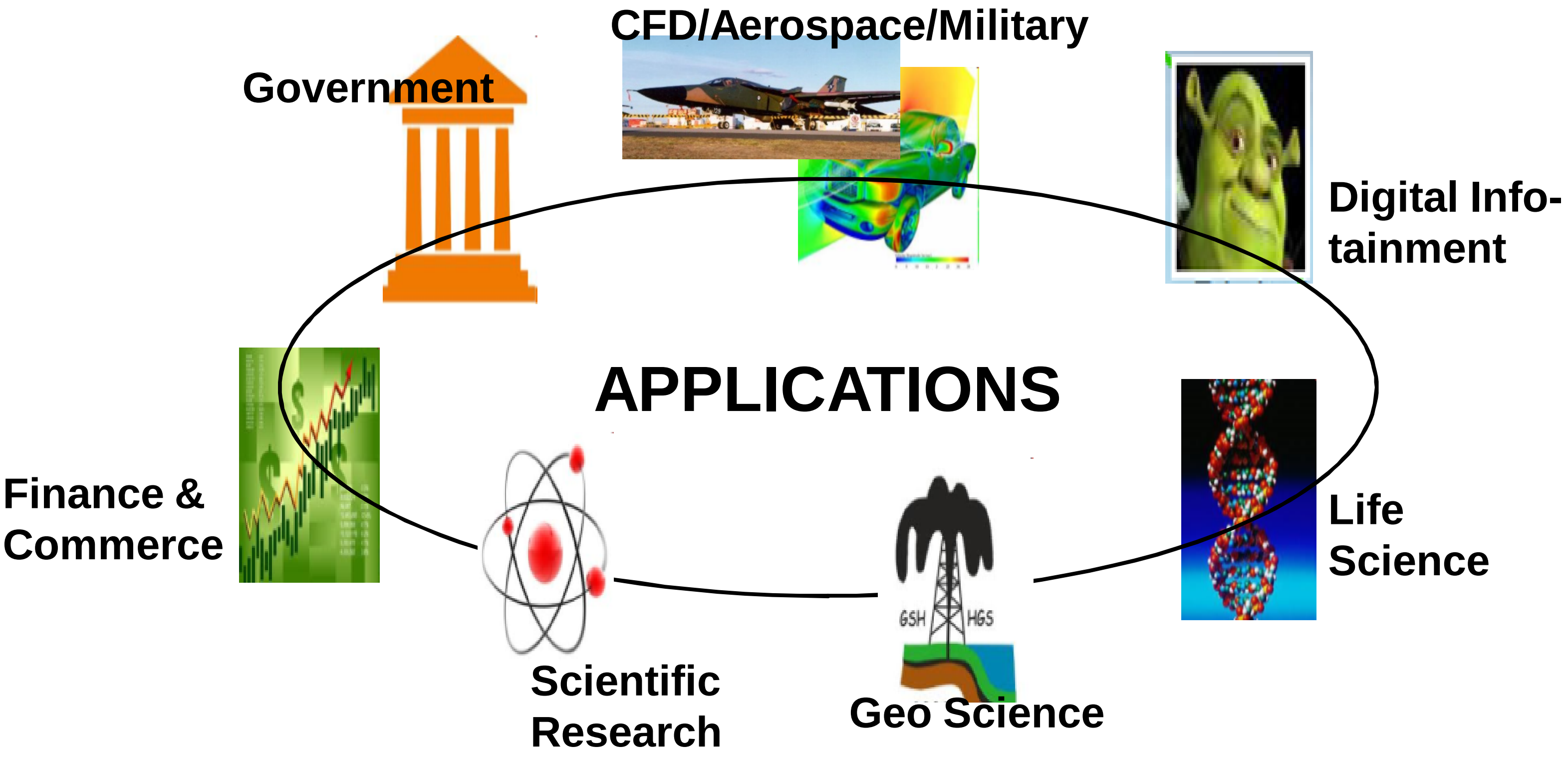


Health

2012 Gartner "IT Hype Cycle" for Emerging Technologies



New Age Applications



CFD/Aerospace/Military

Government

Digital Info-tainment

APPLICATIONS

Finance & Commerce

Life Science

Scientific Research

Geo Science

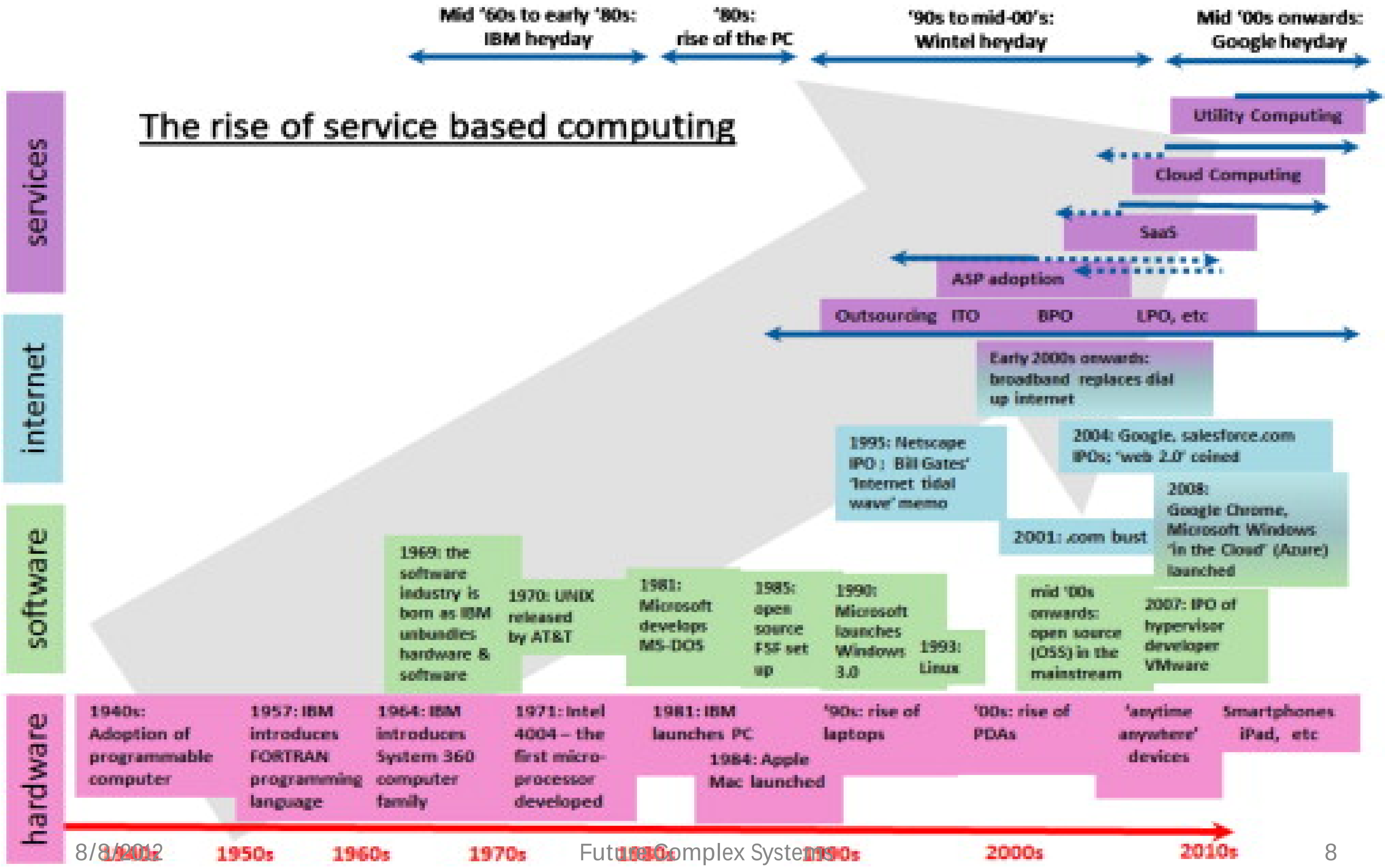
Complex Applications:

- Face recognition + Security
- Audio Processing + HMMSpeech Recognition
- Social Network + Demographic studies

Application Features:

- More Data
- Higher Performance
- Multidisciplinary
- InterConnected
- Green Computing

Evolution of Computing Systems



Mid '60s to early '80s: IBM heyday

'80s: rise of the PC

'90s to mid-00's: Wintel heyday

Mid '00s onwards: Google heyday

The rise of service based computing

services

Utility Computing

Cloud Computing

SaaS

ASP adoption

Outsourcing ITO BPO LPO, etc

internet

Early 2000s onwards: broadband replaces dial up internet

1995: Netscape IPO : Bill Gates' 'Internet tidal wave' memo

2004: Google, salesforce.com IPOs; 'web 2.0' coined

2001: .com bust

2008: Google Chrome, Microsoft Windows 'in the Cloud' (Azure) launched

software

1969: the software industry is born as IBM unbundles hardware & software

1970: UNIX released by AT&T

1981: Microsoft develops MS-DOS

1985: open source FSF set up

1990: Microsoft launches Windows 3.0

1993: Linux

mid '00s onwards: open source (OSS) in the mainstream

2007: IPO of hypervisor developer VMware

hardware

1940s: Adoption of programmable computer

1957: IBM introduces FORTRAN programming language

1964: IBM introduces System 360 computer family

1971: Intel 4004 – the first micro-processor developed

1981: IBM launches PC

1984: Apple Mac launched

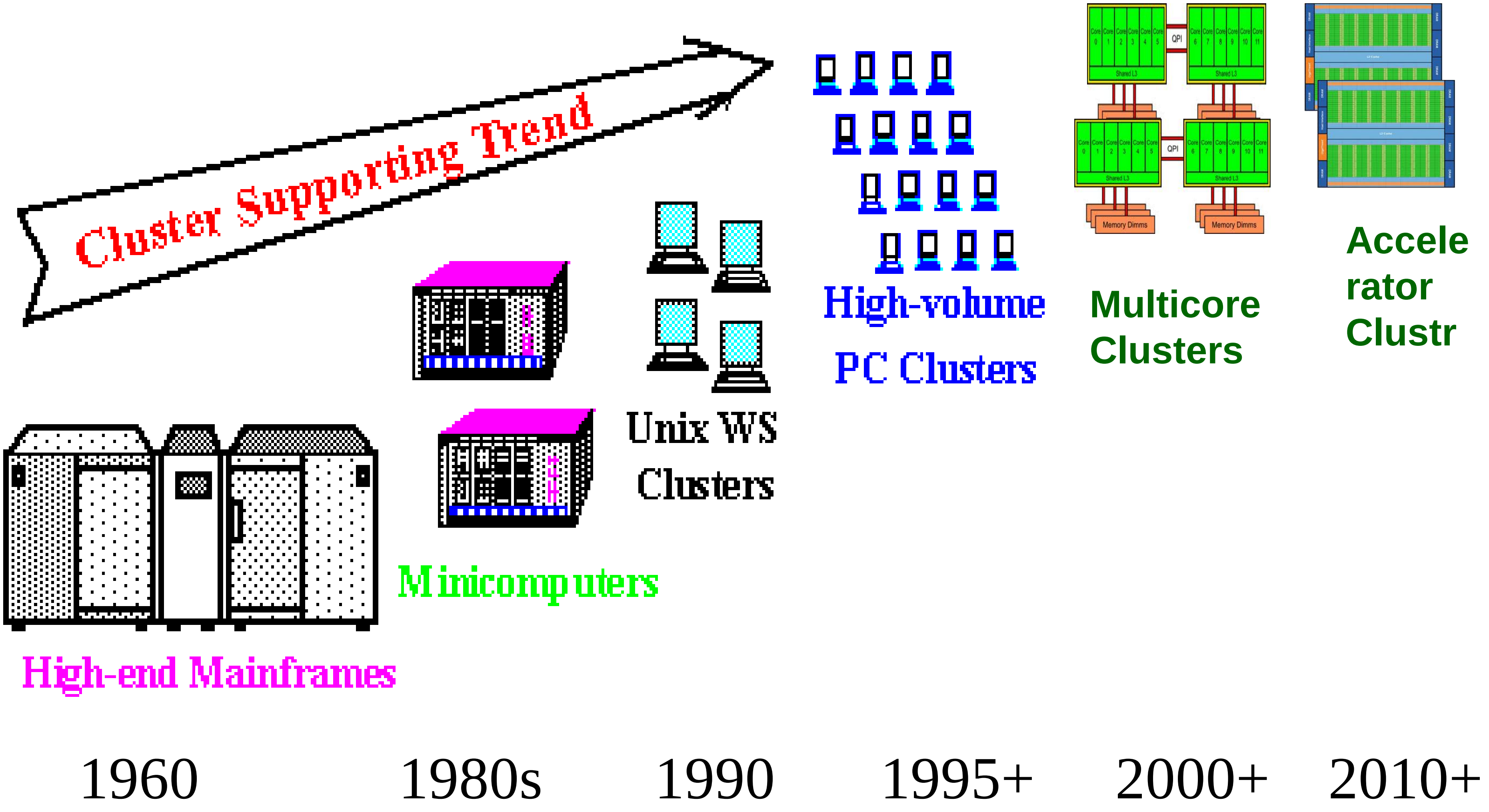
'90s: rise of laptops

'00s: rise of PDAs

'anytime anywhere' devices

Smartphones iPad, etc

1940s 1950s 1960s 1970s 1980s 1990s 2000s 2010s



Cluster Supporting Trend

High-end Mainframes

Minicomputers

Unix WS Clusters

High-volume PC Clusters

Multicore Clusters

Accelerator Clustr

1960

1980s

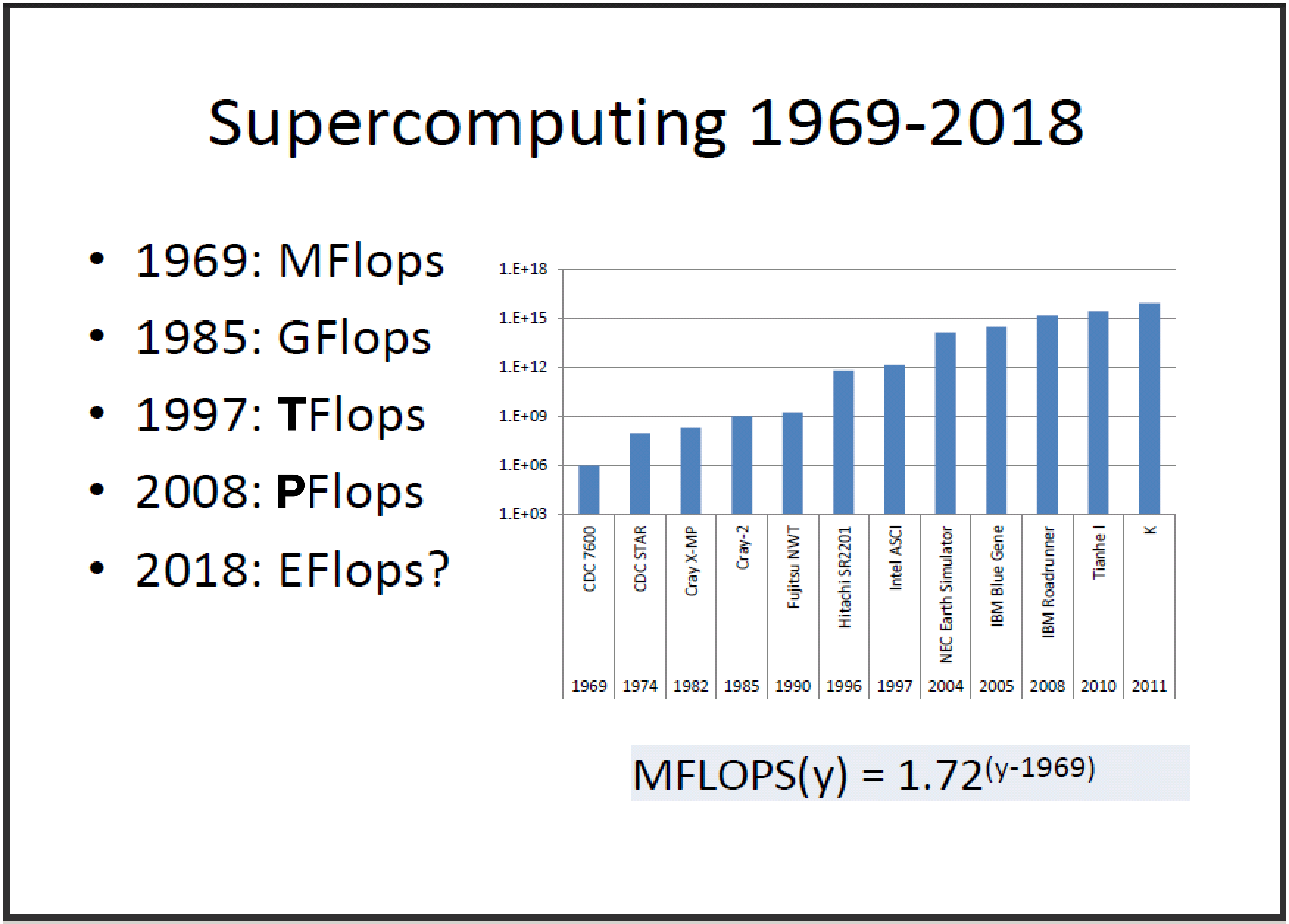
1990

1995+

2000+

2010+

Advances in Parallel & Distributed Computing



Supercomputing 1969-2018

- 1969: MFlops
- 1985: GFlops
- 1997: **T**Flops
- 2008: **P**Flops
- 2018: EFlops?

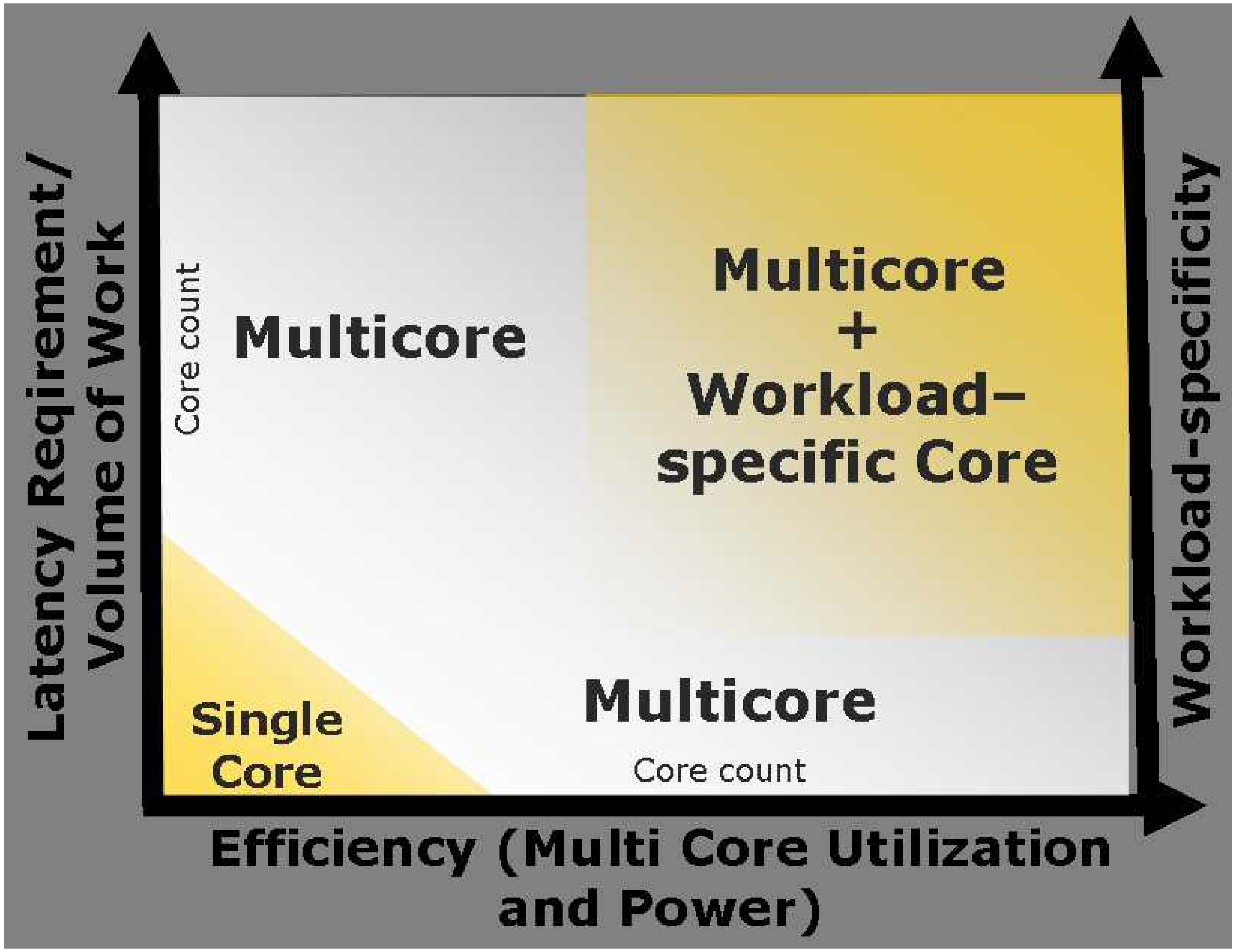
$$\mathrm{MFLOPS}(y) = 1.72^{(y-1969)}$$

Heterogeneous Computing

- Clear trend towards multi-core heterogeneous systems
- Problem: increased application-design complexity
  - Different resources require different algorithms to execute efficiently

- High performance computing is moving towards heterogeneous systems that combine
  - Multi-core CPUs with
  - Accelerators to extract more parallelism at a lower power footprint

Need of Multicore+Accelerator



Memory Wall , Power Wall

- This “memory wall”, the limited data speed between CPU and memory outside the CPU chip, can be highlighted by the limited communication bandwidth across chip boundaries. From the mid-1980s through 2000, CPU speed doubled every 2 years while memory speed took eight years to double

- CPU evolution has also reached a “power wall” that is limiting the design of new processing elements.

- As a result of this power wall and the stalling clock rates, chip manufacturers have been adding additional cores to chips rather than speeding them up

Parallelism Saves Power

- **Exploit explicit parallelism for reducing power**

Power = (C * V^2 * F)/4   Performance = (Cores * F)*1

Capacitance Voltage Frequency

- **Using additional cores**
  - Increase density (= more transistors = more capacitance)
  - Can increase cores (2x) and performance (2x)
  - Or increase cores (2x), but decrease frequency (1/2): same performance at ¼ the power
- **Additional benefits**
  - Small/simple cores → more predictable performance

Many Core Motivation

- **Limitations**
  - Memory Wall (limited data speed between CPU and memory
  - outside the CPU chip)
  - Power Wall

Power = (2C * V^2/4 * F/2)

C=Capacitance V=Voltage F=Frequency

Performance = (2Cores * F/2)

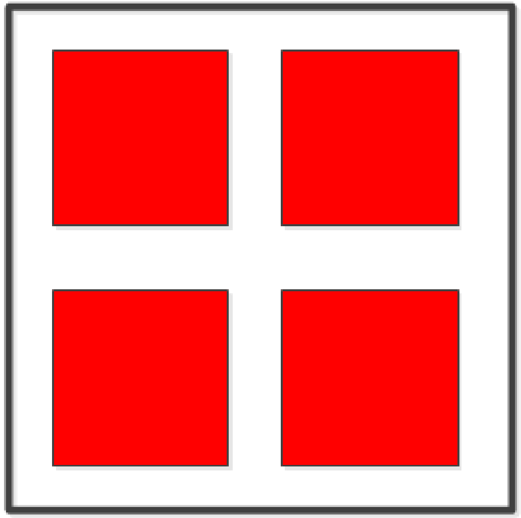
Same performance at lesser power

•Heterogeneous Computing - with 2 main categories of accelerators - GPUs and FPGAs

•Superior performance is achieved by leveraging the massive parallelism of these co-processor chips

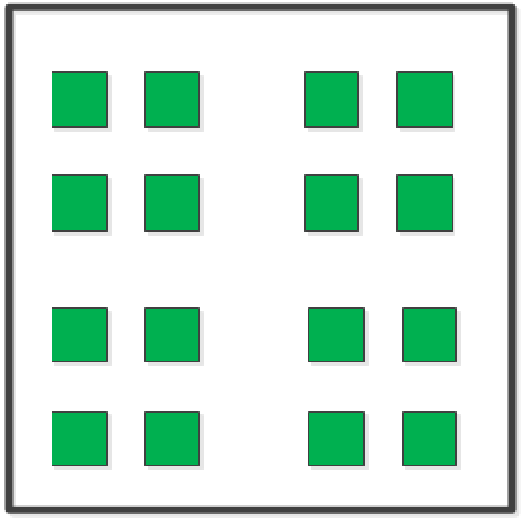
Multicore+Accelerators

CPUs



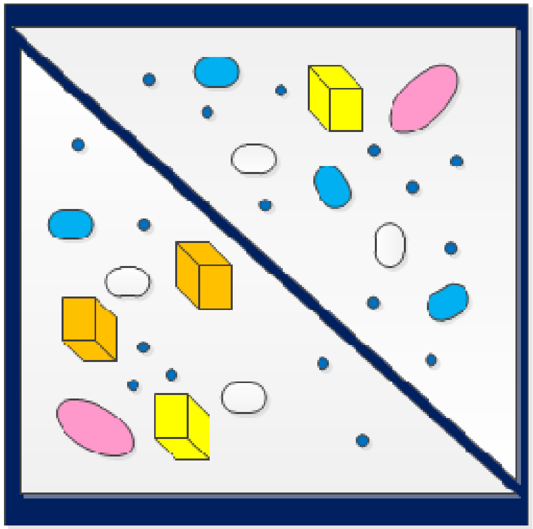
Multi-core
Chip Multi-processor (CMP)
Fixed High-Complexity
Instruction Sets

GPUs

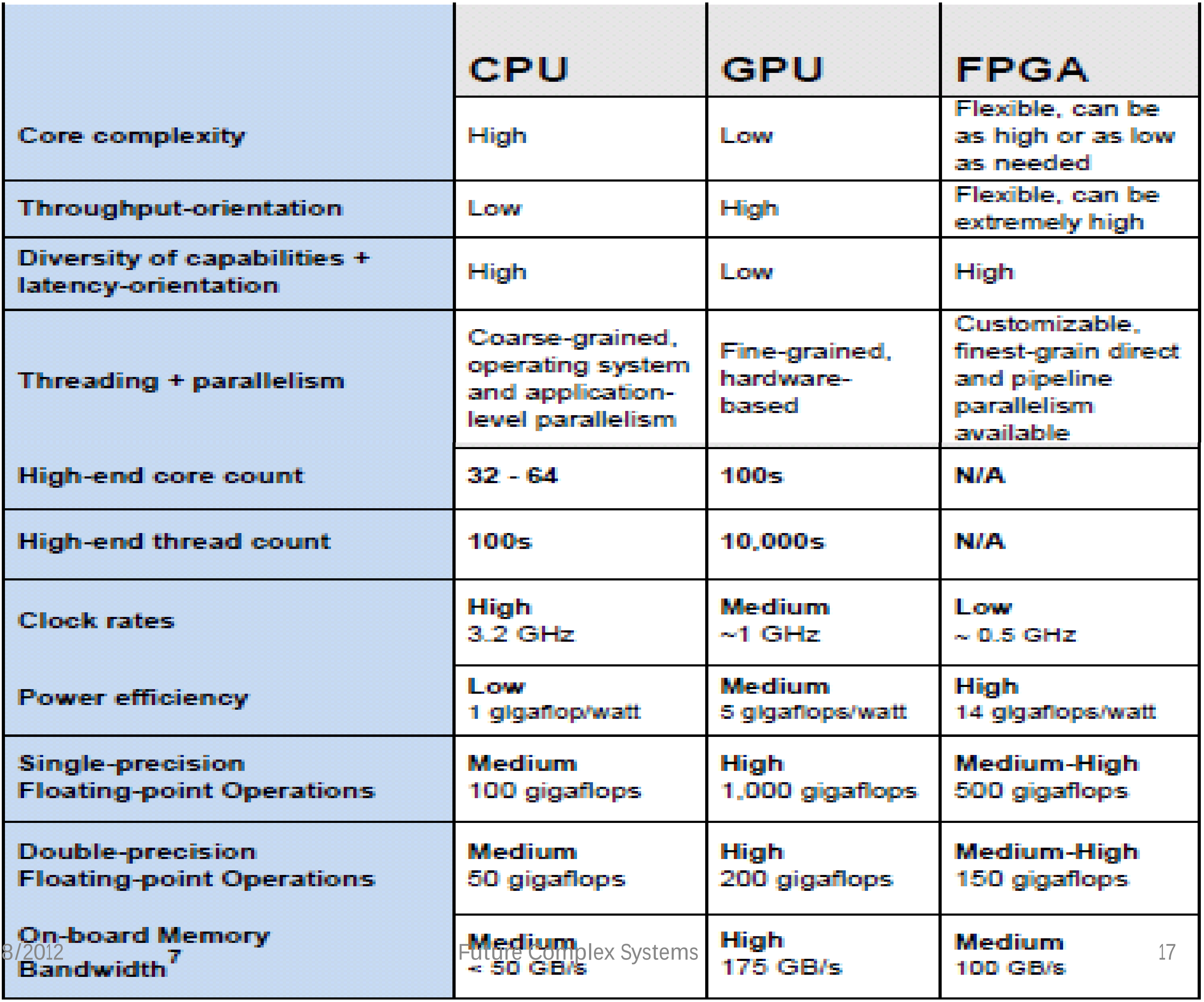


Many-core
Chip Multi-processor (CMP)
Fixed Simpler Instruction Sets
Hardware-based SIMD / SIMT

FPGAs



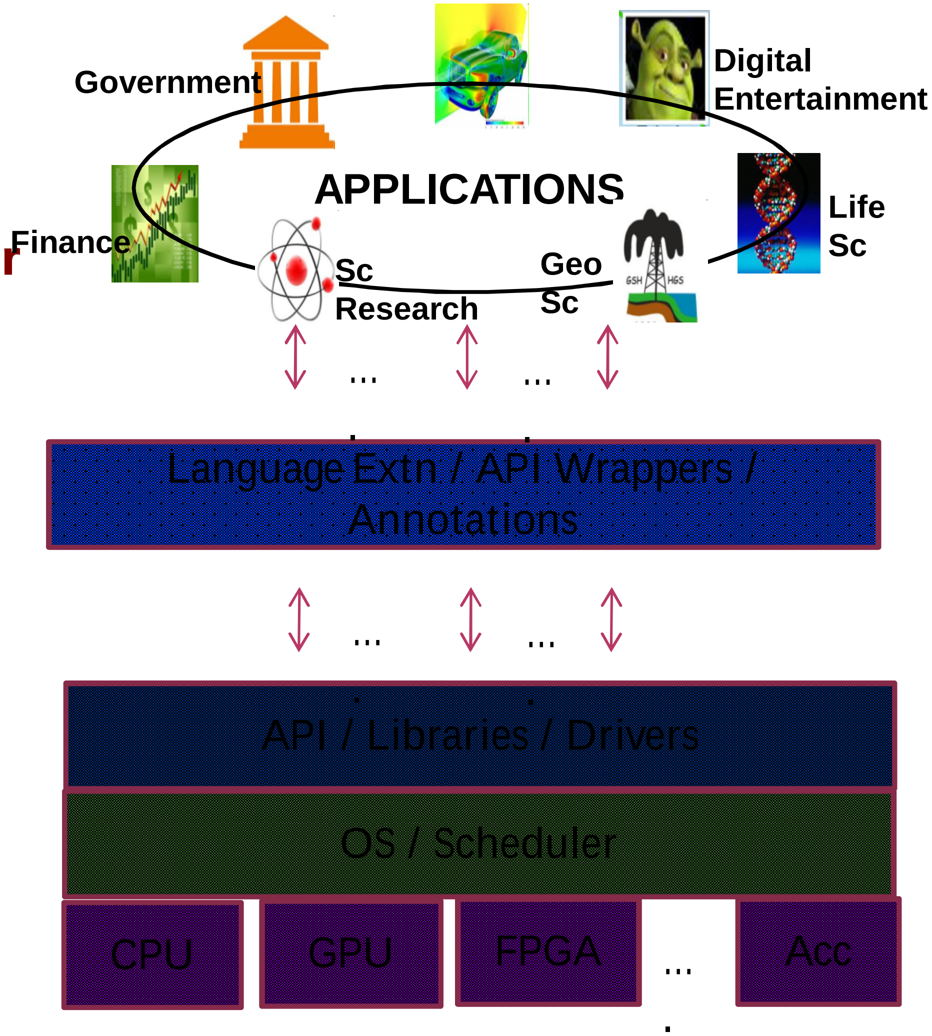
Customizable cores
Instruction-less
Direct Execution of Application



	CPU	GPU	FPGA
Core complexity	High	Low	Flexible, can be as high or as low as needed
Throughput-orientation	Low	High	Flexible, can be extremely high
Diversity of capabilities + latency-orientation	High	Low	High
Threading + parallelism	Coarse-grained, operating system and application-level parallelism	Fine-grained, hardware-based	Customizable, finest-grain direct and pipeline parallelism available
High-end core count	32 - 64	100s	N/A
High-end thread count	100s	10,000s	N/A
Clock rates	High 3.2 GHz	Medium ~1 GHz	Low ~ 0.5 GHz
Power efficiency	Low 1 gigaflop/watt	Medium 5 gigaflops/watt	High 14 gigaflops/watt
Single-precision Floating-point Operations	Medium 100 gigaflops	High 1,000 gigaflops	Medium-High 500 gigaflops
Double-precision Floating-point Operations	Medium 50 gigaflops	High 200 gigaflops	Medium-High 150 gigaflops
On-board Memory Bandwidth[7]	Medium < 50 GB/s	High 175 GB/s	Medium 100 GB/s

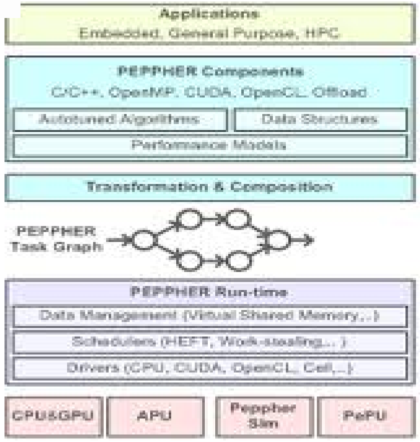
Hybrid System - Architecture & Components

- **Establish Hybrid System**
- **Parallel code development framework**
  - **Programming Model/ abstraction for hybrid**
  - **FPGA development tools**
  - **Optimized math & application specific libraries**
  - **Scheduler**
  - **Target code generating driver & Runtime**
  - **Debugger**
  - **Profiler/Performance Analyzer**
  - **Application Profiler**
  - **System management tools**



International Scenario: Hybrid Systems

- **PEPPHER – (PErformance Portability & Programmability for HETERogeneous many core architectures) European Commission – www.peppher.eu**



Applications
Embedded, General Purpose, HPC

PEPPHER Components
C/C++, OpenMP, CUDA, OpenCL, Offload
Autotuned Algorithms | Data Structures
Performance Models

Transformation & Composition

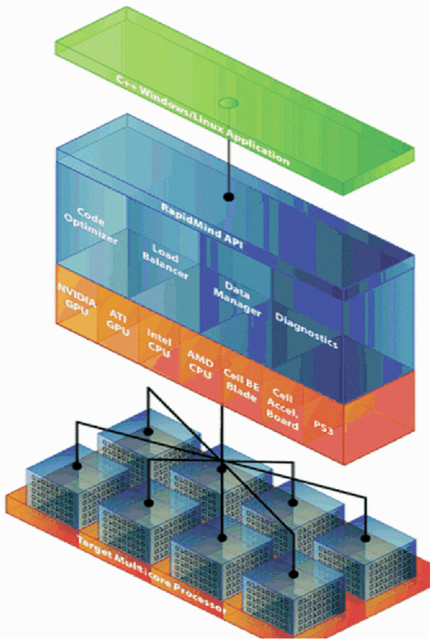
PEPPHER Task Graph

PEPPHER Run-time
Data Management (Virtual Shared Memory,..)
Schedulers (HEFT, Work-stealing,..)
Drivers (CPU, CUDA, OpenCL, Cell,..)

CPU&GPU | APU | Peppher Sim | PePU

- **VelociData** - **St Louis, Missouri, USA - High-performance products that leverage architecturally diverse compute platform, FPGAs, GPGPUs along with std compute resources**

- **RapidMind** Development Platform – **started in 2004, Univ Waterloo, Ontario, Canada, acquired by INTEL 2009**

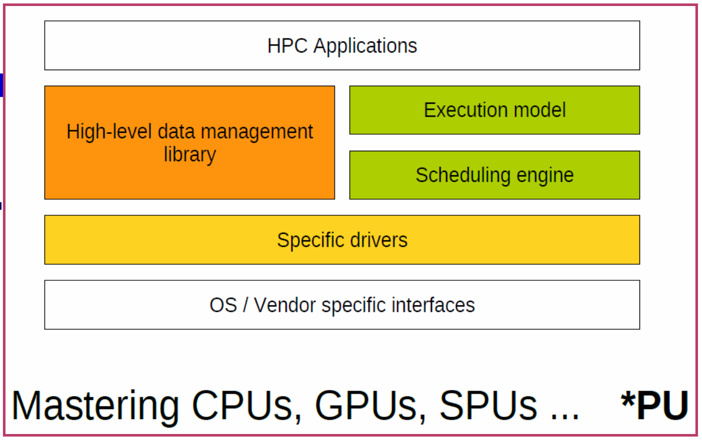


- **MAGMA** Project (M**atrix** A**lgebra on** G**PU and** M**ulticore** Ar**chitectures**) -**S. Tomov J. Dongarra V. Volkov J. Demn (Univ of Tennessee, California Univ, Colarado Univ.) ; innovative algos http://icl.cs.utk.edu/magma/**

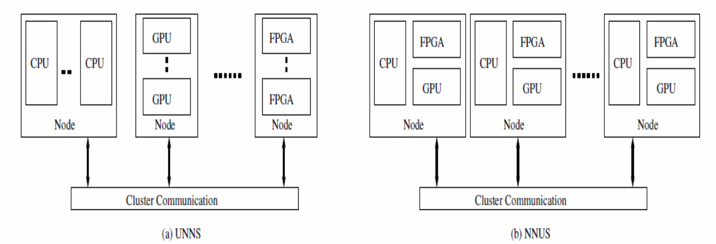
- **DarkHorse** – **Proposed Petascale Architecture (+) Los Alamos National Lab, Oak Ridge National Lab**

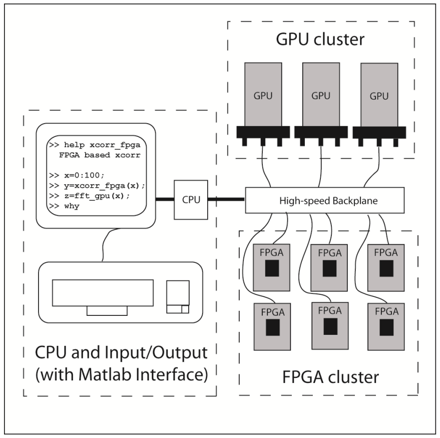
International Scenario

- **StarPU - INRIA Bordeaux – LaBRI, University of Bordeaux - A unified runtime system for heterogeneou multicore architectures**
- **IBM Liquid Metal Project – address difficulties if devp. Applns for heterogeneous machines http://researcher.ibm.com/liquidmetal**
- **Universities / R&D Labs**
  - **University of Virginia, Accelerating Compute-Intensive Applications with GPUs and FPGAs**
  - **Washington University St. Louis, Visions for Application Development on Hybrid Computing Systems.**
  - **University of Illinois at Urbana-Champaign , QP: A Heterogeneous Multi-Accelerator Cluster**
  - **The Australian National University, Canberra, Australia, An FPGA/GPU/CPU hybrid platform for solving hard problems**
  - **University of Tubingen, Germany & University of Pernambuco, Brazil, Exploiting Heterogeneous Computing Platforms by Cataloging Best Solutions for Resource Intensive Seismic Appln**
  - **Massachusetts Institute of Technology, StreamIT compilation infrastructure for streaming applns for variety of targets**
  - **Tianjin University, China, Object driven Workload allocation in heterogeneous computing systems**


HPC Applications
High-level data management library
Execution model
Scheduling engine
Specific drivers
OS / Vendor specific interfaces

Mastering CPUs, GPUs, SPUs ... *PU





```
>> help xcorr_fpga
  FPGA based xcorr

>> x=0:100;
>> y=xcorr_fpga(x);
>> z=fft_gpu(x);
>> why
```

Indian Scenario: Hybrid Systems

- VSSC: 220TF Hybrid (400CPU + 400GPU) --- now enhanced to 400TF
- IIT Mumbai (Project: High Performance Computing using GPU, FPGA and Multicore Processors; Approved under Naval Research Board, PI: Dr. Sachin Patkar, Associate Prof, IIT Bombay, EE Dept)
- IISc Bangalore (Programming Models, languages, and compilation for accelerator based architecture, Prof. Govindarajan, SERC; REDEFINE: Runtime Reconfigurable Polymorphic ASIC, Prof. S K Nandy, SERC)
- IIT Delhi (hybrid platform CPU, GPU, FPGA in CSE Dept)
- **Workshops:**

– First Workshop on Hybrid Multicore Computing in conjunction with HiPC 2010 @ Goa, India

– Casper Workshop 2011, Pune, India; how to build astronomical signal-processing instrument (polyphase filtering spectrometer) using FPGA & GPU

– Intl Conf. on Field Programmable Technology, New Delhi, India, Dec 12-14 2011

Exascale Challenges

Broadly classified under Concurrency, Resiliency, Energy, Heterogeneity

- Power–energy / Operation of computation, Data Transport, Memory
- Threading software to millions/billions of threads
- Memory/Storage capacity and bandwidth
- Scalable data analysis, mining SW
- New I/O models, SW, runtime systems and libraries
- Extremely scalable performance methods and tools
- Fault tolerance
  - Scalable fault tolerant MPI
  - Resilience API and Utilities
  - Managing high-node count systems in the existence of failures (MTBF)
- Exascale Programming Models
- Architecture aware algorithms
- Asynchronous methods
- Self adapting hybrid hierarchical based algorithms
- Algorithms that minimize communication
- Scalability for debugger issues
- Scalable control Algorithms to bridge gap between global & local power models
- Affordability

Ref: International Exascale Software Project Roadmap

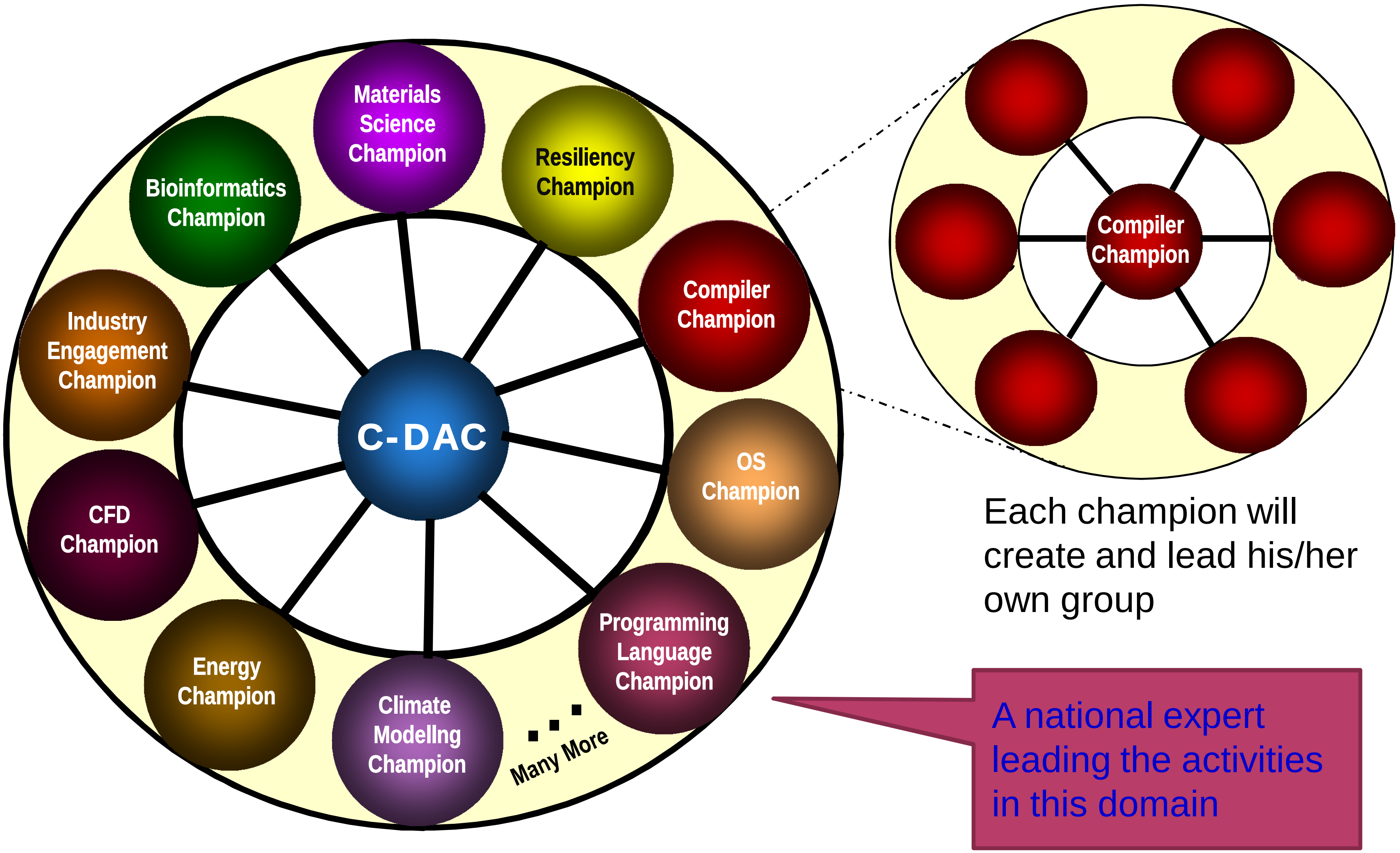
Revolutionary Approaches to Exascale

The following are some of the recommended Approaches by IESP_WG

- Execution Models
  - Current Parallel Model: Communicating Sequential Processes(CSP)
  - New Models: Provide Greater Support for Asynchrony than CSP
- Think Parallel
  - Future Systems Relinquish Controlling The machine,
  - Exploit runtime information that allow Variability of Exec Path
- Incorporating Intelligent Methods
  - Asynchronous methods Self adapting hybrid hierarchical based algorithms
  - Algorithms that minimize communication
- Operating Systems
  - Should address billion cores, associated Mem and Inter Connection Networks, Dynamic Scheduling, Fault Tol, and Energy Control.
- Programming Models
- Correctness & debugging
- Persistent Storage

Ref: International Exascale Software Project Roadmap-Working Group Recos

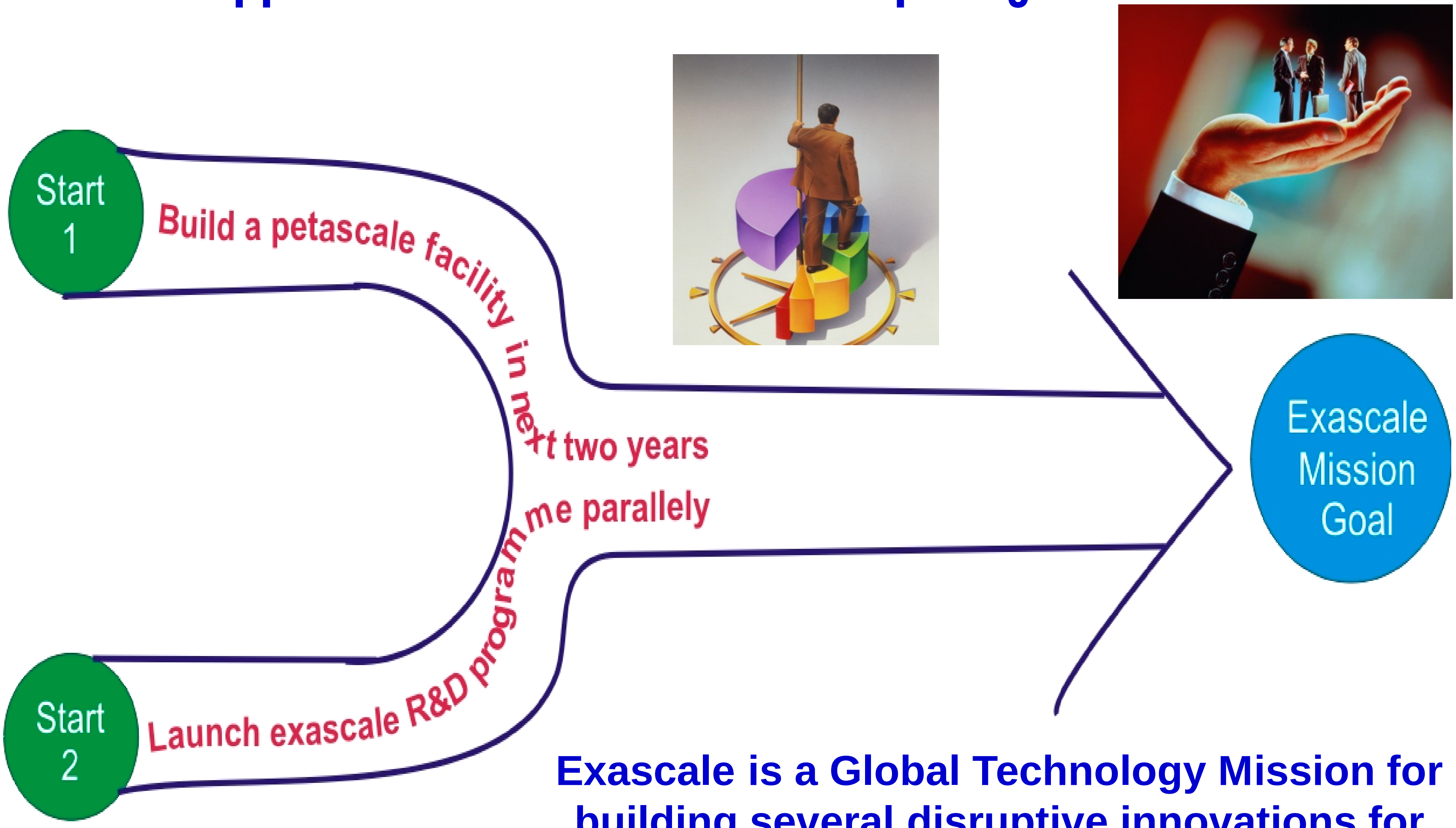
Creating the Wheel of Success for the Mega Mission



Each champion will create and lead his/her own group

A national expert leading the activities in this domain

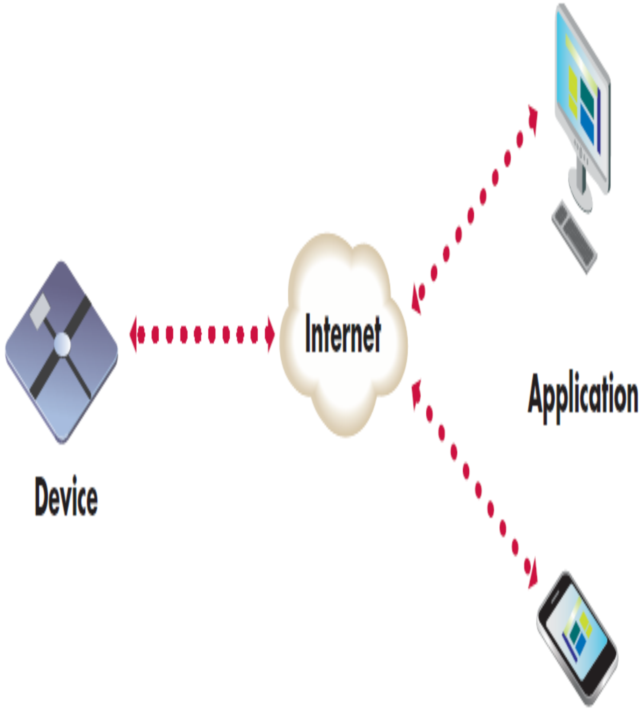
C-DAC approach for Peta2ExaScale Computing


Start 1
Build a petascale facility in next two years
Start 2
Launch exascale R&D programme parallely
Exascale Mission Goal

Exascale is a Global Technology Mission for building several disruptive innovations for HPC in 21st Century.

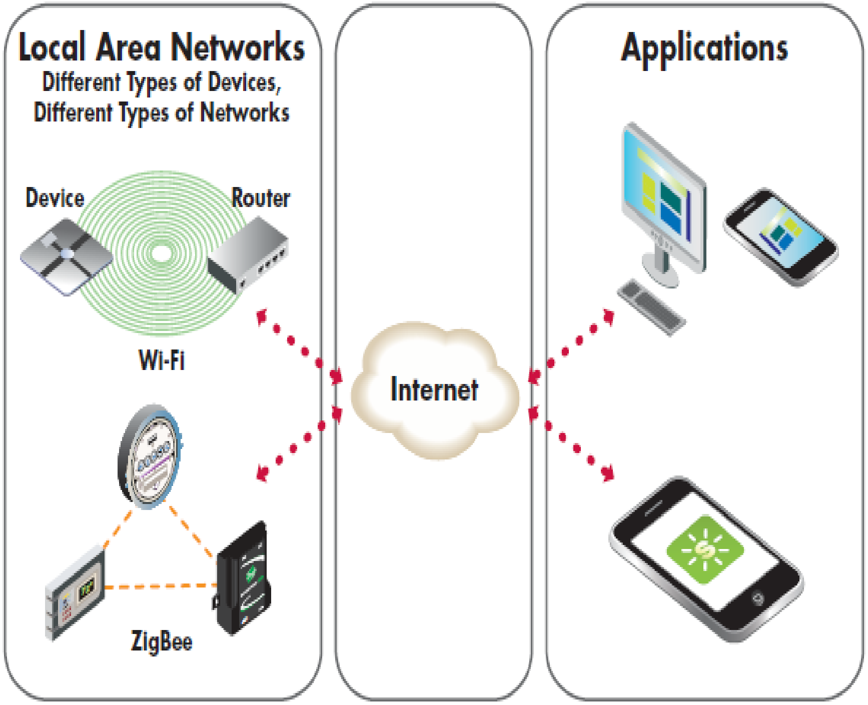
Internet Of things On Cloud

Internet Of things



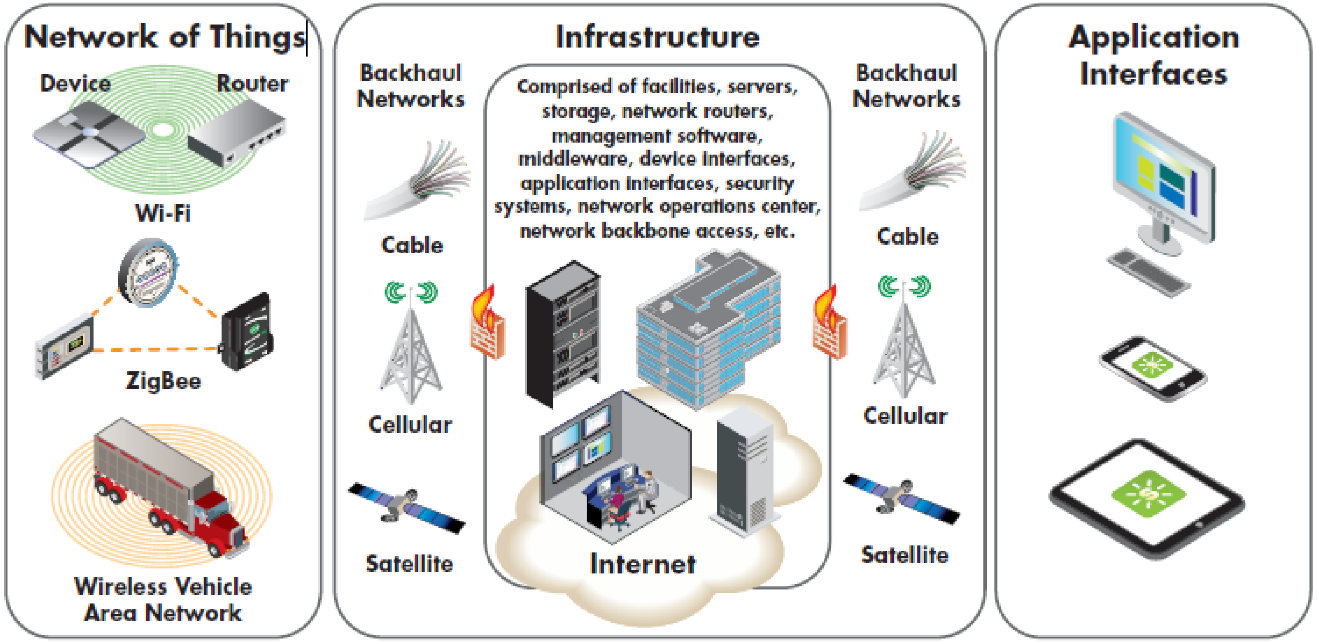
- In internet Computers are attached
- Internet of things, things around us like car, refrigerator etc will be attached.
- Consists any device with the ability to gather and process information, and communicate it across the network.

How Does This Work?



- Different types of devices comprise a network (LAN, ZigBee as shown in figure)
- These networks communicate with the applications via Internet
- The applications may have different interfaces: PC,cellphone,tablets etc

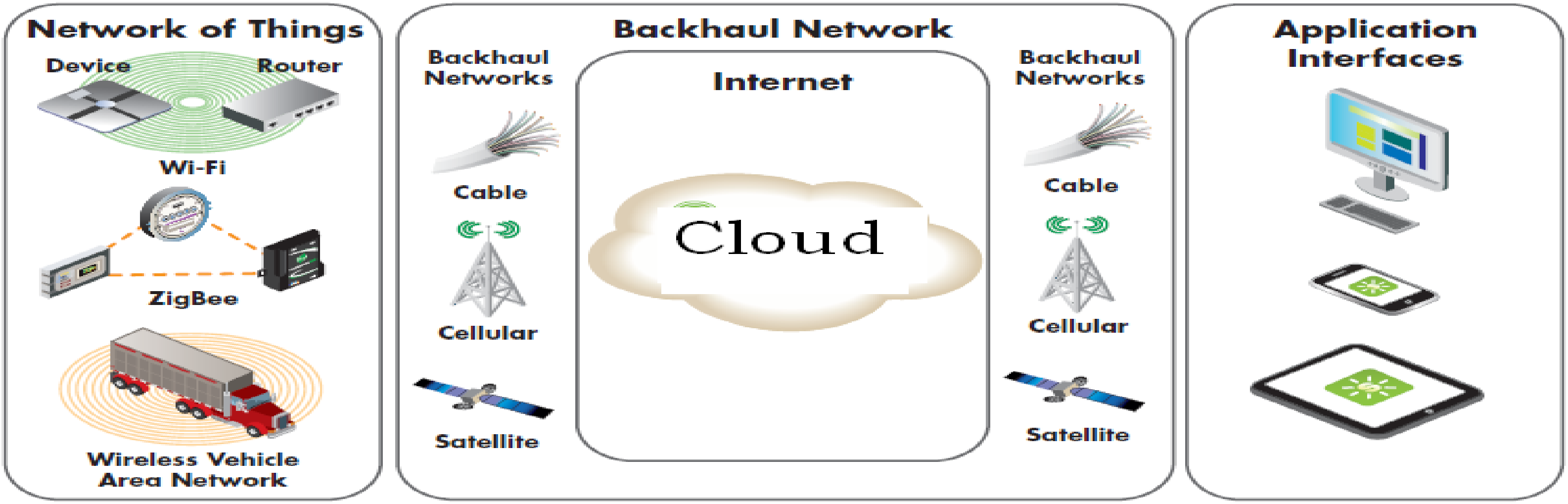
Problem With this Approach?



Network of Things
Device
Router
Wi-Fi
ZigBee
Wireless Vehicle Area Network
Infrastructure
Backhaul Networks
Cable
Cellular
Satellite
Comprised of facilities, servers, storage, network routers, management software, middleware, device interfaces, application interfaces, security systems, network operations center, network backbone access, etc.
Internet
Backhaul Networks
Cable
Cellular
Satellite
Application Interfaces

- To connect to Internet and communicate information to applications, we need back-haul network (as shown in figure).
- The many types and individual components of infrastructure can create massive complexities for solution providers.
- Challenges: Scalability, less usability of resources etc

Internet Of things On cloud:



- If infrastructure is provided as a service, most of the problems of solution provider can be solved like scalability, flexibility etc.
- Now with this kind of set-up, solution provider has to worry about his application and not on the complex infrastructure.

Internet Of things On cloud:

- The individual devices will gather the information and send it to the cloud, the application will read the information from the cloud and show it in different interfaces.
- The features of Internet of things on cloud are:

  1. Scalability,
  2. Flexibility
  3. Better Resource Utilization,etc

Applications of Internet Of Things

Smart Parking:

- **Monitoring of parking spaces availability in the city**

Traffic Congestion:

- **Monitoring of vehicles and pedestrian levels to optimize driving and walking routes.**

Radiation Levels

- **Distributed measurement of radiation levels in nuclear power stations surroundings to generate leakage alerts.**

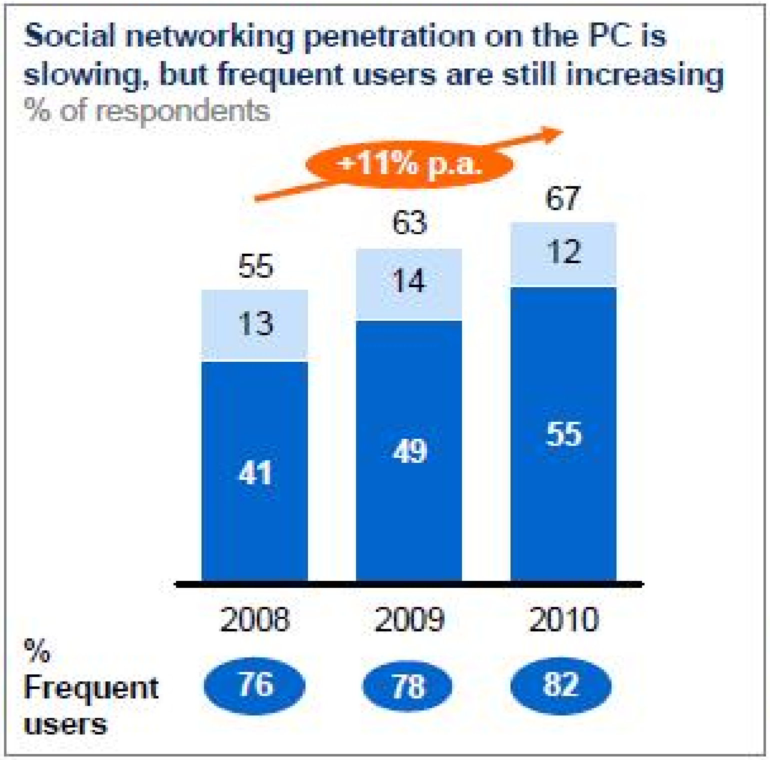
Fall Detection(e-Health)

- **Assistance for elderly or disabled people living independent.**

Sensing As A Service & BigData

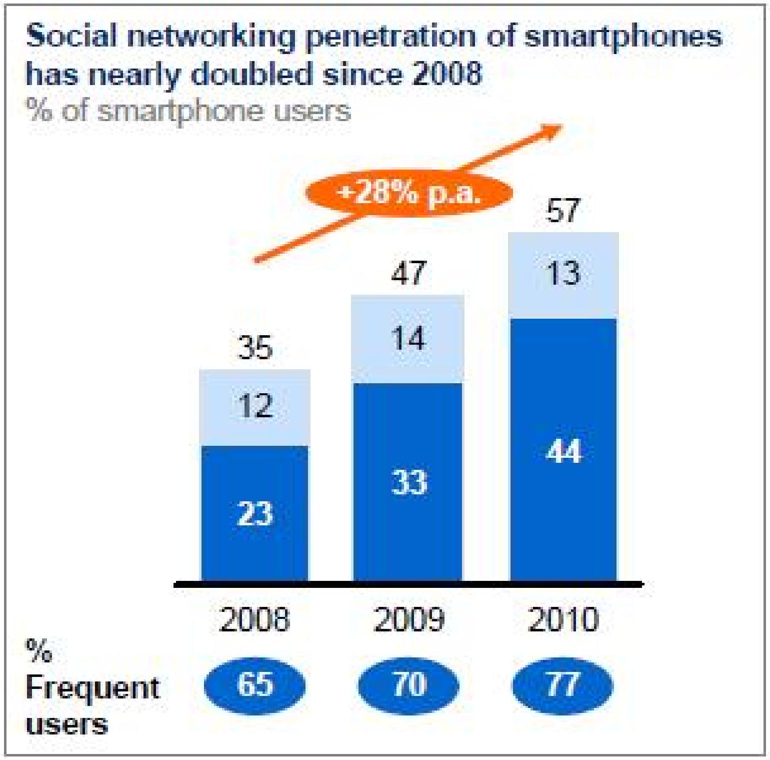
- **Modern world is full of devices comprising sensors, data processor.**
- **Such resources enable sensing, capturing, collection, and processing of real time data from billions of connected devices.**
- **Fact: In 2010 , the total amount of data on earth exceeded one ZB (zetabyte). By the end of 2011, the number grew up to 1.8 ZB.**
- **It is expected that this number will reach 35 ZB in 2020.**

Sensing Big Data



Social networking penetration on the PC is slowing, but frequent users are still increasing

% of respondents



Social networking penetration of smartphones has nearly doubled since 2008

% of smartphone users

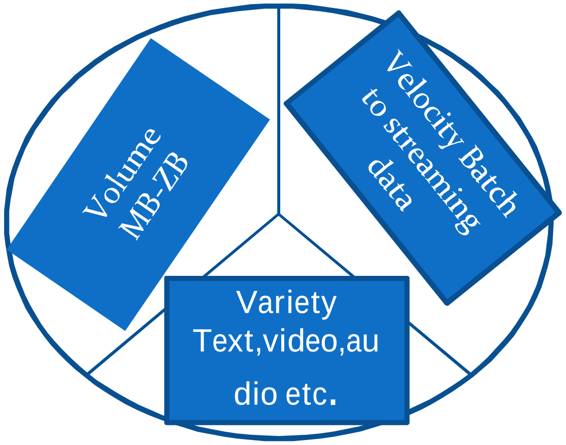
1 Based on penetration of users who browse social network sites. For consistency, we exclude Twitter-specific questions (added to survey in 2009) and location-based mobile social networks (e.g., Foursquare, added to survey in 2010).

2 Frequent users defined as those that use social networking at least once a week.

SOURCE: McKinsey iConsumer Survey

Characteristics Of Big Data

- **Volume : Size of the data terabyte, petabyte, zetabyte.**
- **Variety : Types of data. In addition different sources will produce big data such as social networks, web etc.**
- **Velocity : It means how frequently data is generated. For Example, every milliseconds, seconds , minute, hour etc.**



Uses of Sensors

- Example of sensing technology is Nike+iPod/iPhone that collects and tracks information details such as workout details, distance, calories burnt etc.
- State of California has developed a greenhouse gas sensor network , it collects information about greenhouse gases and their behavior.
- Statistics: 12TB of tweets in twitter and 25TB of data in facebook generated everyday. There are 2B people on the web.

Sensing As A Service Model

- IoT envision that sensors to be attached everywhere. In such environment owner will be able to generate data.
- Sensing as service can be done at personal , private and public organization level.
- Owner of sensors will be able to generate data and get a return investment.

Sensing as a Service: Applications

- Mike bought a refrigerator, fridge automatically identified the availability of wifi in house.
- The sensors attached to fridge generate data so interested parties can access the data by paying a fee.
- One ice-cream company access the fridge data and give a 5% discount to Mike.
- Now, that company can find out information such as how many times users consumes dairy per week, at which time user likely to eat them and so on.

Sensor Cloud ?

- Infrastructure that aims at managing physical sensors by connecting them to the cloud.
- It uses SensorML to describe metadata of physical sensors description and measurements.
- SensorML is an standard model and XML encoding mechanism for describing sensors.

Sensor Cloud

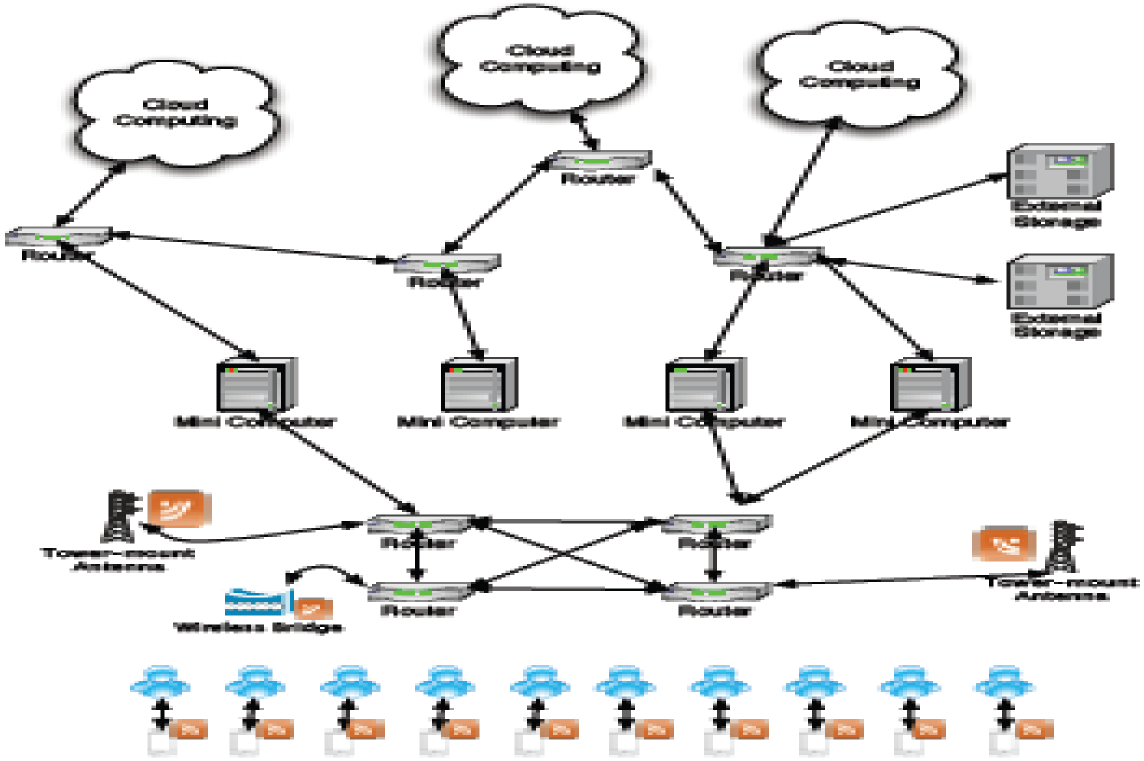


Figure 1. A depiction of the different components of the sensor and cloud-computing network. Android smartphones denote the sensors in the system, and are in the possession of individuals. The smartphones have some computational capacity, and transmit through WiFi or cellular services to a brokering network, running over traditional TCP/IP services. The brokering service can itself have computers performing filtering, processing and/or creating other mashups of sensor data.

Presentation Outline

- C-DAC Current Research Areas
  - Grid Computing: GARUDA
  - Cloud Computing: C-DAC Scientific Cloud
  - Multicore-Accelerators
  - Mobile Computing: AR based mLearning
  - Ubiquitous Computing
- New Application Scenarios
  - Petascale Computing
  - Hybrid Computing
- **C-DAC Future Complex Systems**
  - Exascale Computing
  - Internet of Things: Mobile+WSN, Sensor Clouds
  - **Large Complex Systems: Energy Internet**

THANK YOU