中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

Building Billion-Threads Computer and Elastic Processor

Guo-jie Li
Institute of Computing Technology (ICT)
Chinese Academy of Sciences (CAS)

www.ict.ac.cn
lig@ict.ac.cn

Great Computer Challenges in the Next Decade

- The characteristic of computer applications in the next decade is computing for the masses.
- High volume throughput computing to satisfy the requirements:
  - A high number of active users,
  - a high number of parallel requests,
  - massive amount of data, etc.
- Elastic computer architecture to satisfy the a great diversity of Internet-of-Thing applications
  - The ossified computer architecture cannot be suitable for the various niche applications.

Zhiwei Xu and Guo-jie Li, "Computing for the masses" Communication of the ACM, Vol. 55 No.10 ,Oct. 2011.

Open Problems

- What are the new workloads?
  - “real” workloads open to academic community
- What should be the new metrics?
  - Beyond Linpack and flop/s
  - Can we calculate energy complexity for each application?
- What is the suitable architecture for the high volume throughput computing and elastic computing?
  - How to deal with the mass amount of threads?
  - How to make tradeoff between energy-efficient and generality
- What is a good software stack?
  - What new properties? How to evaluate a stack?

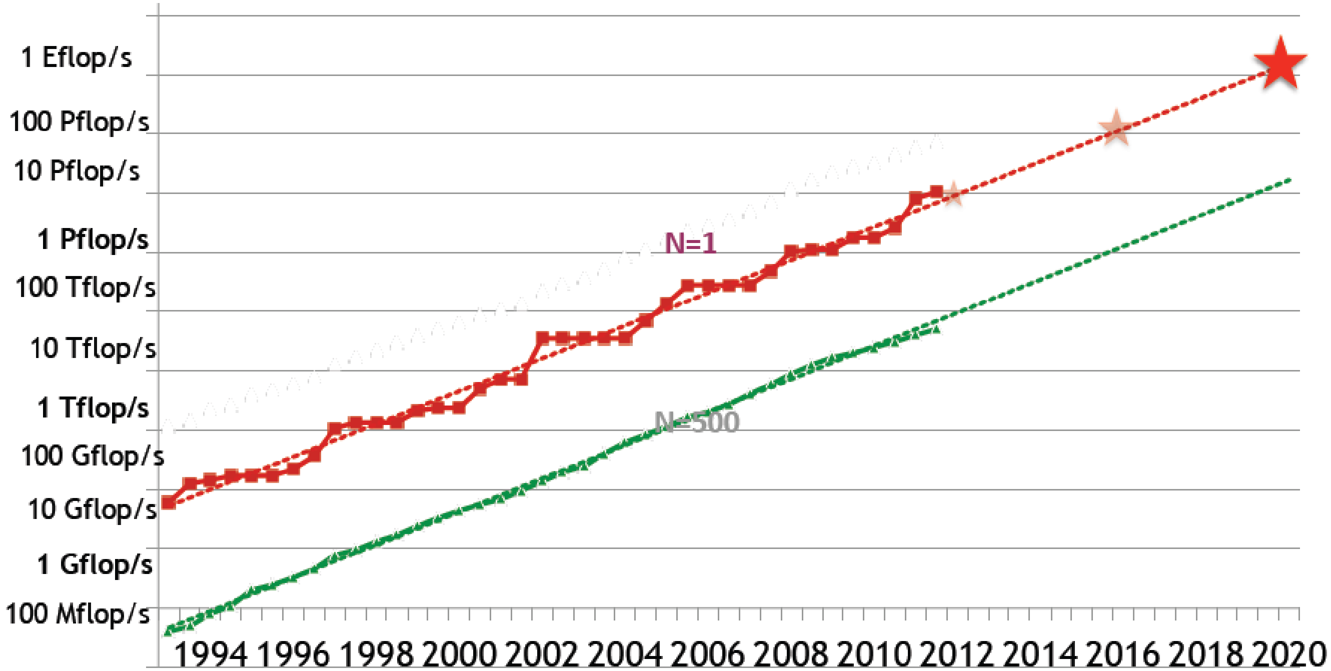
FIT Initiative of CAS and SKL on Computer Architecture in ICT

- To address the above issues, Chinese Academy of Sciences (CAS) has started-up the Future Information Technology (FIT) Initiative, a 10-years frontier research project for targeting applications and markets of 2020-2030.
- The unique State Key Lab on Computer Architecture (CARCH for short) in China, which located at Institute of Computing Technology (ICT), is one of major undertakers of the FIT project.
- The research directions of CARCH and ICT include thousands threads on a chip, billion-threads computer, elastic processor, new router of future Internet, etc.

Why Do We Suggest Building Billion-Threads Computer?

International Parallel & Distributed Processing Symposium (IPDPS)

- What can “Distributed processing” learn from “Parallel”?
- **A focused, evolving, multi-decade goal** for supercomputing systems
  - Manifested as a history-proof (maybe future proof) performance metric: **flops**
  - which can compare and rank parallel systems (Top500.org)



Jack Dongarra, On the Future of High Performance Computing: How to Think for Peta and Exascale Computing, SCI Institute, University of Utah, February 10, 2012,

Five Benefits of Having Such a Focused Goal

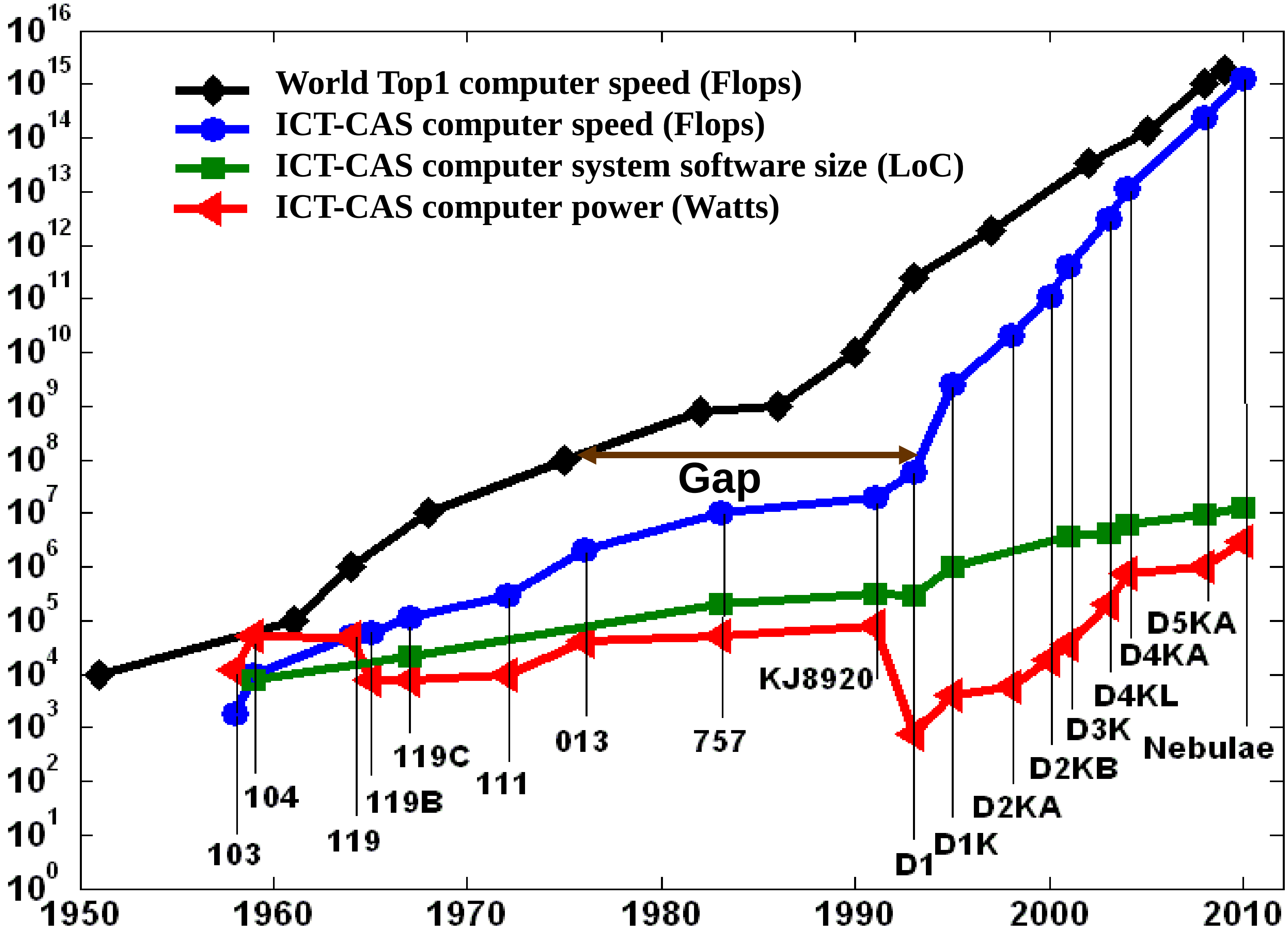
- **Provide objectives**: concrete and precise objectives for supercomputer systems R&D
  - – Linpack flops is a yardstick for 20 years
  - – Others include HPCS, Green500, etc.
- **Measure advance of R&D: about 1000 times** improvement every 10 years
  - – 1990: Gflops; 2000: Tflops; 2010: Pflops
- **Enable a roadmap:** historical trends can be used as a basis to set future roadmap
  - – The new goal for 2020 is Exa-scale systems

Five Benefits of Having Such a Focused Goal

- **Trickle down**:
  technology trickles down to mass products such as PCs and Pads
- **Facilitate community**:

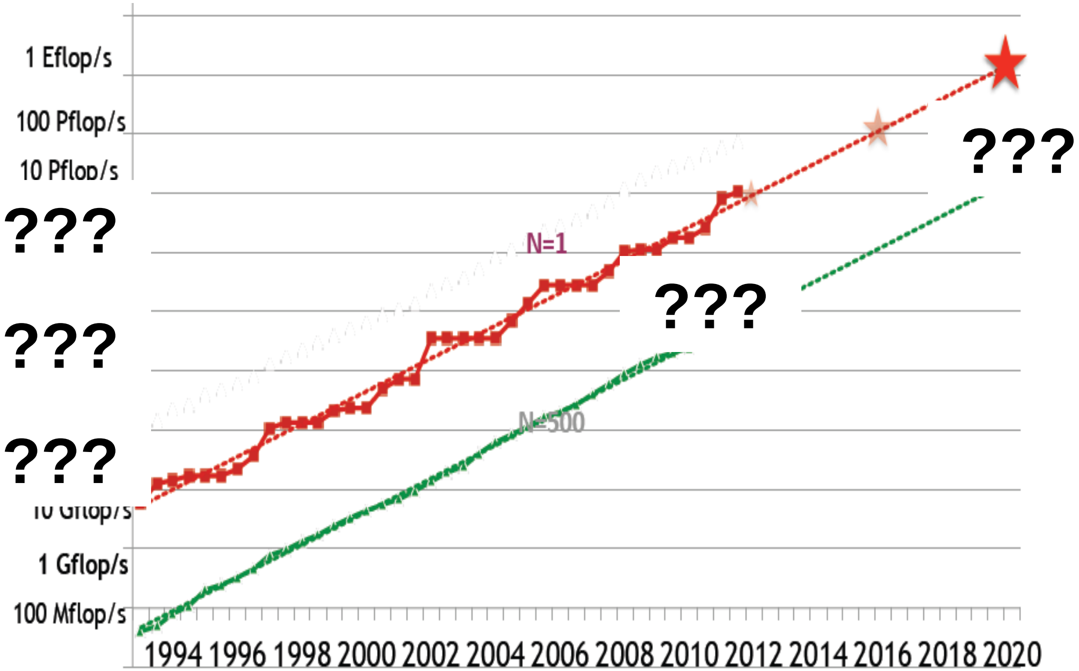
  Parallel processing has a worldwide, broad-based HPC community
  - Brings together devices, systems, software, application people
  - Draws together academia, industry & government
  - Works together to solve funding, R&D, use, and education issues

The Benefit for China
(helps to know the gaps and challenges)



No Such Focused Goal Exists for Distributed Processing

- No focused performance metrics, no Distributed500
- No measurement of advances: X Times/10 Years?
- No concrete roadmaps: 2000: ??; 2010: ??; 2020??
- **No focused community like HPC**

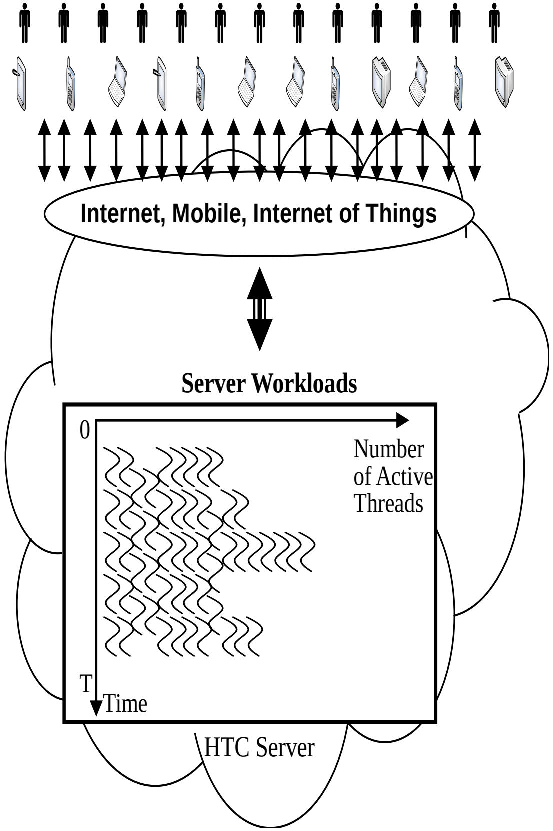


Need a benchmark to characterize workloads of DCCs, and DCC500 !

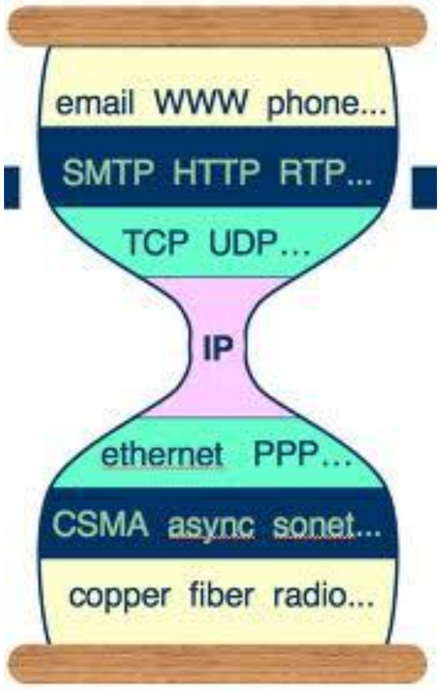
,

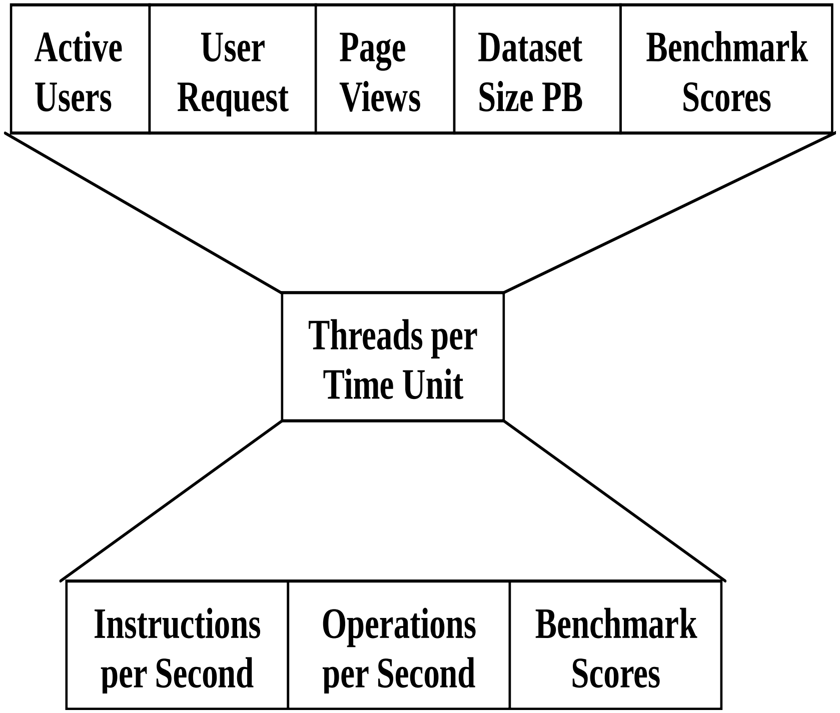
Focus on "Threads Per Second"

- "Threads Per Second" (TPS) is the key metric for datacenter computers (DCCs).
  - A thread is **a schedulable sequence of instruction executions with its own program counter**, like Java thread, CUDA thread of GPU, Hadoop task, etc.
- The definition of thread is not very clear. What we talk about here is soft threads (something like micro-threads in dataflow computer） rather than hardware threads. We may need a new term to denote them.



“Threads Per Second” as hourglass





Active Users | User Request | Page Views | Dataset Size PB | Benchmark Scores

Threads per Time Unit

Instructions per Second | Operations per Second | Benchmark Scores

- “Threads per second” serves as the neck of the performance metrics (thin waist model)
- Like IP is the thin waist of Internet

An Observation:
Volume (number of active threads) matters

- Assume N threads $\{\tau_1,\ldots,\tau_N\}$ are executed in a datacenter computer system in time period $[0, T]$, and power and energy are additive

Throughput =
Volume × Watts per thread × Threads per Joule

  – **Energy** ***E***: Joules consumed by a thread, averaged over $\{\tau_1,\ldots,\tau_N\}$

- Little's Law: $\lambda = L / W$
- New observations:
  – $\lambda = P / E$
  – $\lambda = L \times (E/W) \times (1/E)$

John D. C. Little,
"Little's Law as Viewed on Its 50th Anniversary," OPERATIONS RESEARCH, Vol. 59, No. 3, May–June 2011, pp. 536–549.

A Roadmap Suggestion:
---Using Little’s Law to set a goal
A Billion Threads Peak Volume by 2020

- $\lambda = L \times (E/W) \times (1/E)$

  Throughput = Volume × Watts per thread × Threads per Joule
- For power and energy efficient architecture design
  - Maximize *L* with good enough *W* for user experience
  - Architecture design aims to increasing *L* and 1/*E*, while technology advances controlling *E*/*W*
- How big “peak *L*” was
  - 2000: kilo threads
  - 2010: million threads
  - 2020: billion threads

Attributes of a DCC	**2010**	**2020**
Daily PV (billion)	4-7	20-100
Active threads per PV	1000	10,000
Peak-to-average ratio	2-10	2-15
Peak volume	millions	~1 billion

Godson D: A Data Processor with Thousand of threads

Motivation of Data Processing Unit

Baibu: processing tens of PB every day, store hundreds of PB data , most services require real time processing.

Google : processing 20 PB data every day, 400PB data every months.

In China, there are more than 500 million mobile internet users.

Challenge:

- **Collect, manage, Integrate ,and Analyze massive amounts of data**
- **Handle unstructured data and uncertainty around format variability**
- **Process big data in a timely and energy efficient fashion**

High-Volume Data Processing Unit

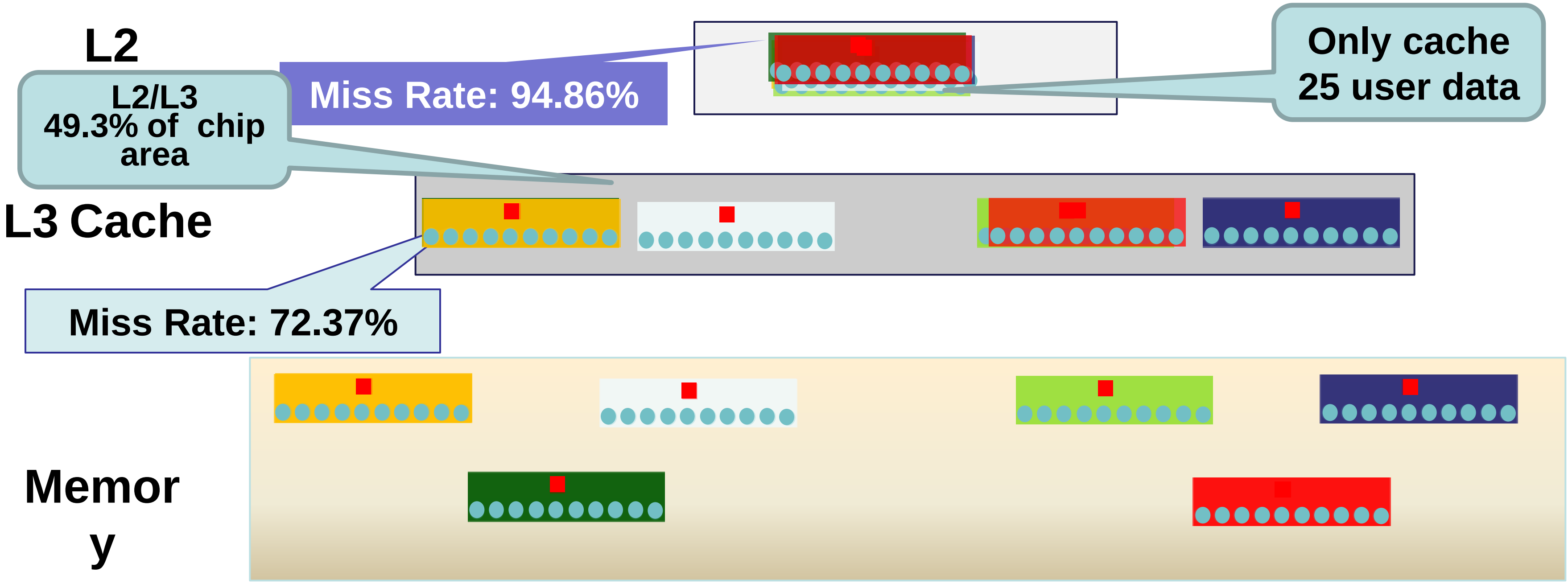
- The essential work is to design a high-volume throughput processor , i.e. High-Volume Data Processing Unit, called Godson-D (D means DPU).
- Through analyzing the special features of cloud service requests, Godson-D will exploit an energy-efficient and scalable micro-architecture for high-volume throughput computing.
- Thread level parallelism with latency hiding technologies and real time technologies for WCET (Worst Case Execution Time) control, will be used on this many-core design, which will satisfy users' interactive and independent requests, irregular memory accesses, and big data processing.

A Case Study of workload

That's a true story…

- A system with hundreds of general purpose high performance processors
- Serves about 1,000,000 users concurrently
- Zoom in:
  -- one state-of-art processor can process about 7000 users' requests in specific period
  -- each user's data size is about 10KB
- What will happen?

Memory Access Latency is the most important factor for Mobile Internet



🟥Dynamic user data ● Static user data 🟨🟩… ⬜ 7000 users' data compete for memory space

Phenomenon

➢L2 and L3 caches occupied 49.3% of chip area, but contribute little because of high miss rate

➢ Bandwidth is much wider than requirement

➢ Float point Unit is useless for this mobile internet scenario

Main Cause

➢ Massive independent user request compete for limited on-chip memory, and *memory access latency* dominates the performance

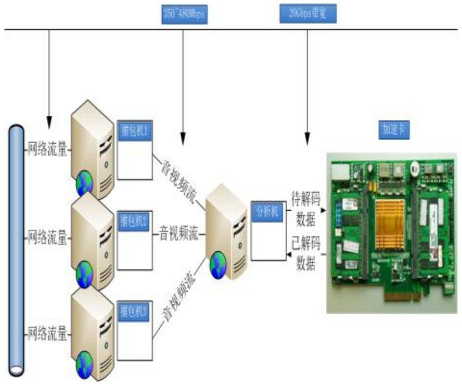
Problem 1: Mismatch between Data Format and Data Structure in CPU

- Example：Simulation of Complex Problems like Security and Social Network--Graph500 application
  - Application Characteristics
    - Needs of Traversing Large Scale Data Structure
    - Irregular Computational Dependency and Memory Access
    - Dynamic Discontinuous Memory Access
    - Low Data Locality and Data Reuse

- Current CPU is Weak for Irregular Data Processing
- Current cache is inefficient for network application
- Need special support for memory on chip
  - ➢ format conversion in data transmission on chip

Problem 2: Mismatch between Data Flow and Data Path in CPU

- Example：Network Video Service
  - ➢ Application Characteristics
    - Stream processing
    - Less Data Reuse
    - Several Steps to Process Data



- Current data flow is always from L3 to L2 to LI cache, but computation unit is not always the core unit
- Computing and memory resource should match up the data flow, the efficient approach is processing data during data transfer.

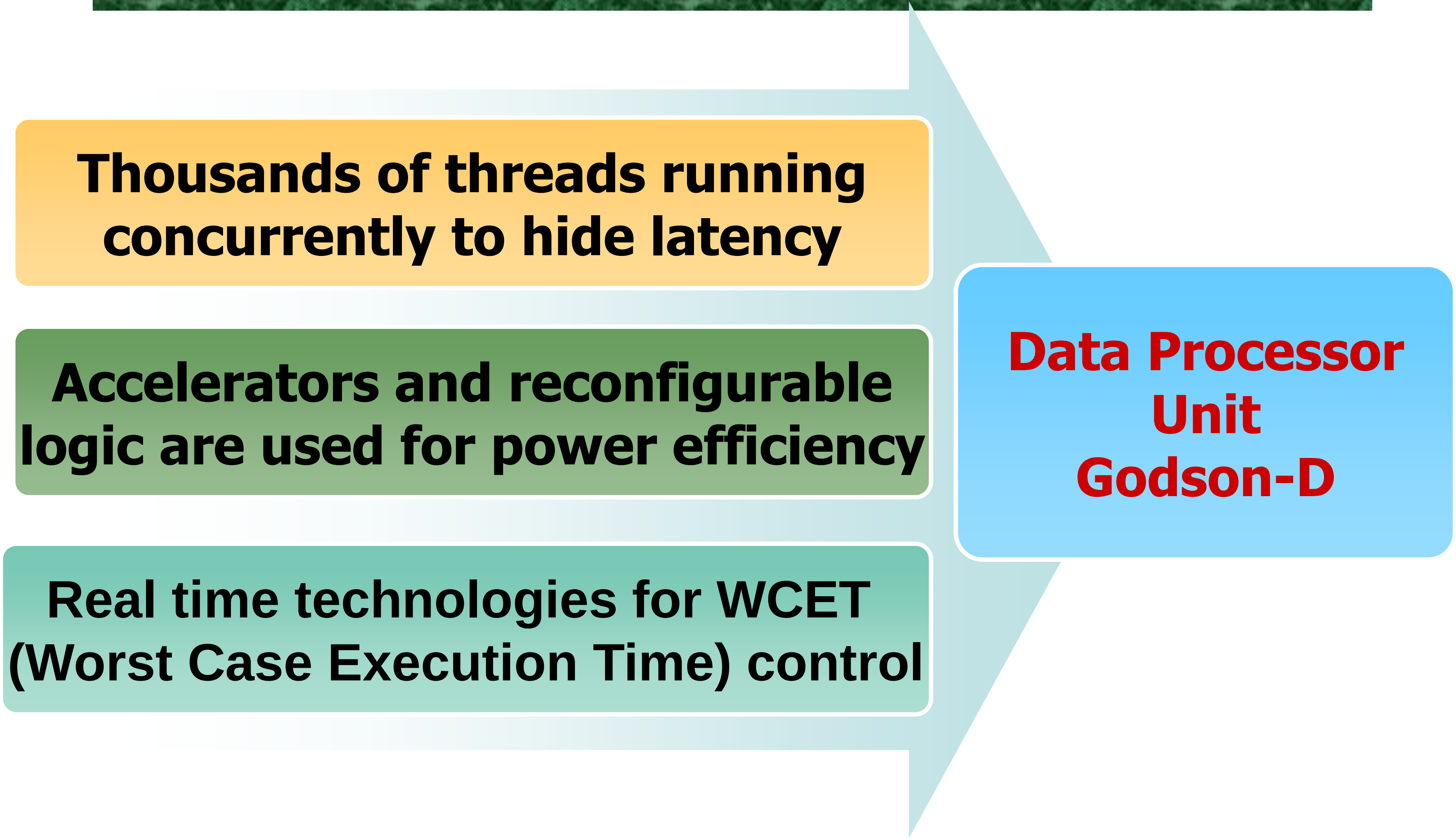
Problem 3： Mismatch between Real Time Data Processing Requirement and the Unpredictable Architecture of CPU

- App spec
  - Mobile devices require real-time processing
  - Network game and network security’s real-time compress/decompress requirement
  - The ETL（Extract/Transform/Load) of Facebook is required to be reduced from 24-28 hours to 10 seconds, to meet real-time analyzing requirements.
- State-of-art processors adopt unpredictable micro-architecture, e.g. out-of-order, branch prediction, pipeline, DMA, and Mesh, so the real-time efficiency is poor.

Problem 4： Mismatch between special processing requirement and the general architecture of CPU

- Existing mainstream CPUs are general purpose processor.
- For various network applications, the special accelerators are needed.
- Obvious mismatch between special processing requirement and the general architecture of CPU
- Will discuss this issue in the following part of elastic chip .

Approaches of Godson-D



Thousands of threads running concurrently to hide latency

Accelerators and reconfigurable logic are used for power efficiency

Real time technologies for WCET (Worst Case Execution Time) control

Data Processor Unit
Godson-D

Draft Architecture of Godson-D

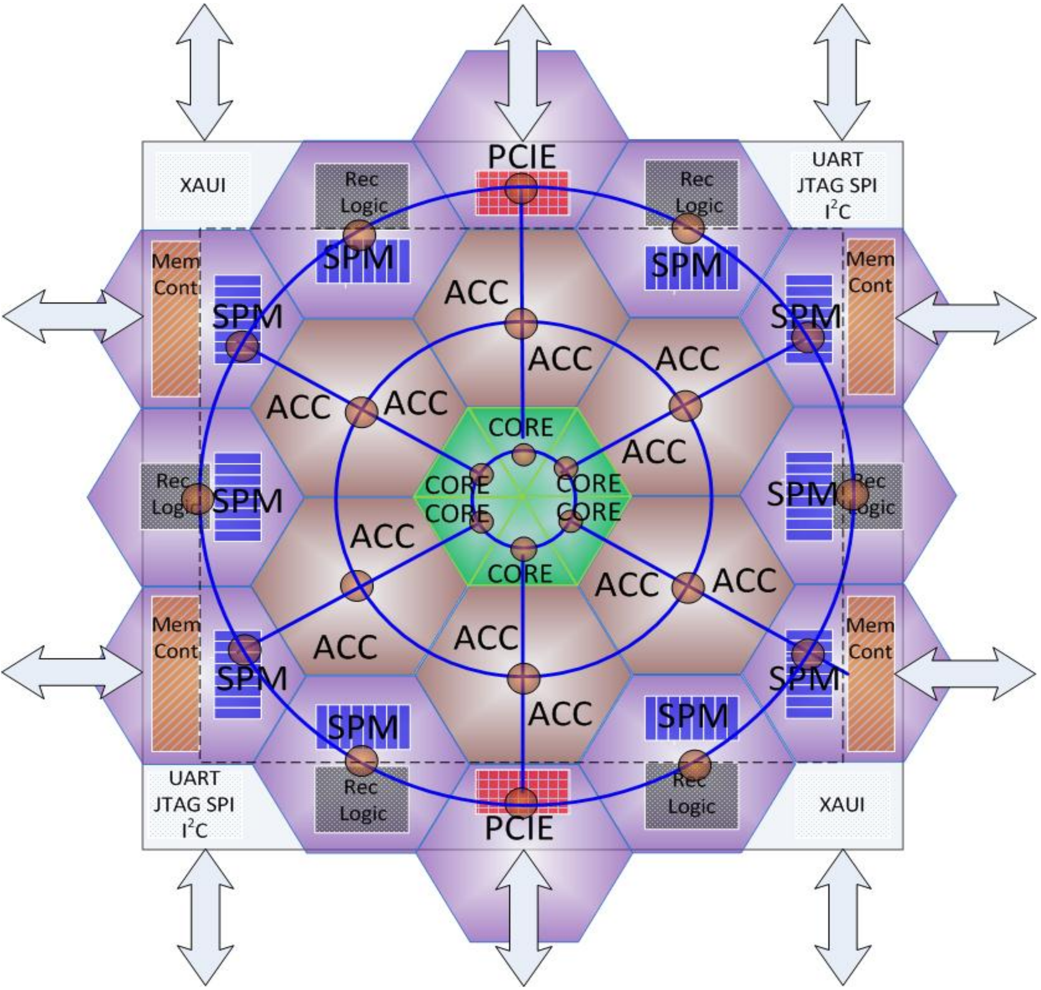
- To resolve four problems mentioned above
- Tile+Special Purpose PE+RC

1-Data format

2-Data Flow

3- ASIC requirement

4-Real time requirement



Features of Godson-D

- High throughput
  - General cores in center
  - Accelerators in periphery
  - Tightly couple accelerators with I/O
- Low latency
  - Tightly couple among accelerators and on-chip memory
  - Shorten the on-chip transfer of data

- High scalability
  - Various components are scalable by adding corresponding “ring”
  - Scalable with new technology (e.g. 3D)
- Real time
  - Predictable WCET
  - Predictable latency of NOC and scheduling components

Three level thread	Description	Number
General Thread	**General purpose cores**	**10x**
Accelerate Thread	**AISC accelerators**	**100x**
Reconfigurable Thread	**Reconfigurable accelerators**	**1000x**

High Volume Throughput Computer （HVC）

What is HVC

- High volume throughput computing(HVC): a datacenter based computing paradigm focusing on throughput-oriented workloads
  - Characteristics: a large amount of loosely coupled jobs.
  - Metrics: volume in terms of requests, data, or the maximum number of simultaneous subscribers
  - Nature: throughput computing
  - Target : high volume

Main Research Topics (1)

- Dynamic allocation and efficient organization of datacenter resources to meet diversified needs
- Workload-aware optimization to ensure each workload can run on a suitable server .
- Software infrastructure for efficient scheduling and monitoring of data center jobs
- Virtualization of data center resources
- RAS((Reliability， Availability， Serviceability) features to ensure stable and sustainable service

Main Research Topics (2)

- Direct interconnection among energy-efficient processors
- Modularized and customizable technology to meet the demand of diversified applications
- Scalable datacenter network to meet massive service requirement
- Integration of a large number of heterogeneous processing units to increase system-level concurrency
- Scalable memory system, high throughput and hybrid storage system

Study on Workload Characterization

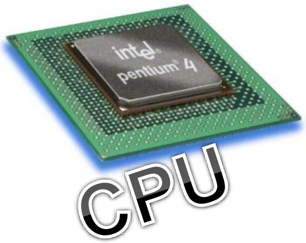
- As an example of our research on HVC, we show here some results on HVC workload .
- Jianfeng Zhan et al, High Volume Throughput Computing: Identifying and Characterizing Throughput Oriented Workloads in Data Centers. Workshop on Large-Scale Parallel Processing in conjunction with 26th IEEE International Parallel and Distributed Processing Symposium(IPDPS 2012).
- *Huafeng Xi,* Jianfeng Zhan, et al, Characterization of Real Workloads of Web Search Engines. 2011 IEEE International Symposium on Workload Characterization (IISWC 2011).

Three categories of workloads in HVC

- Services : A service is a group of applications that to receive user requests and return responses to end users.
- Data processing applications: mainly loosely coupled data-intensive computing.
- Interactive real-time applications: an interactive real-time application will maintain a user session of a long period while guaranteeing the real time quality of service.

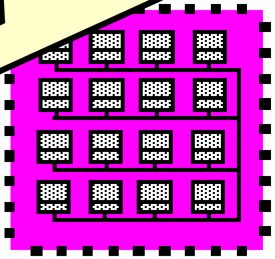
Current Benchmarks

SPEC CPU

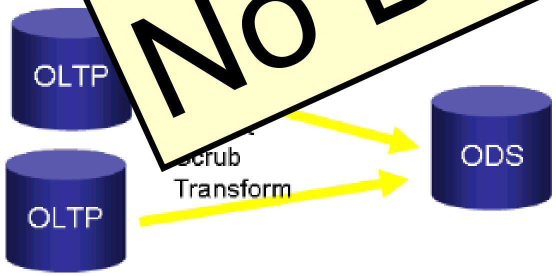


SPEC Web

HPCC



TPCC

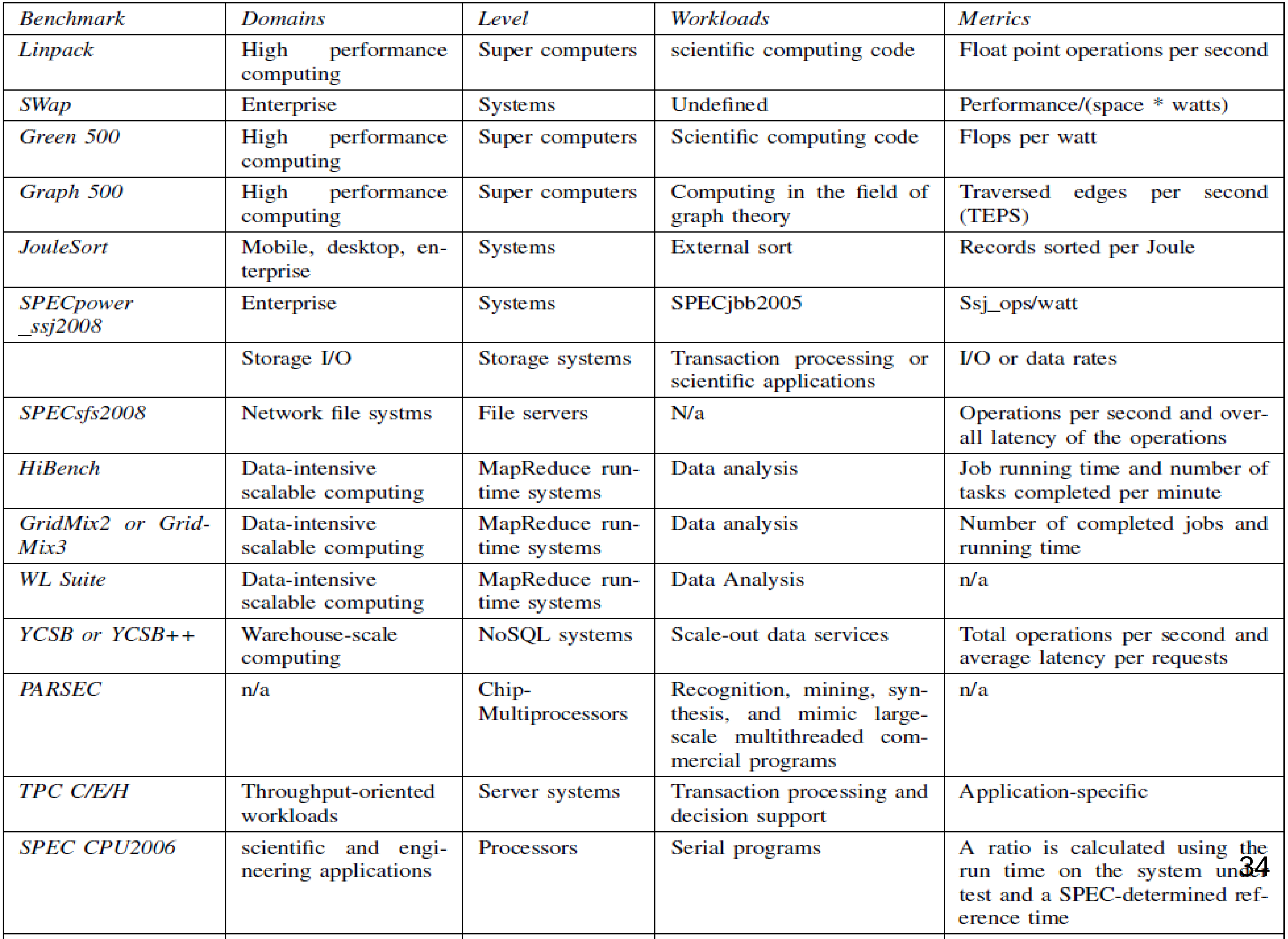


Gridmix

YCSB

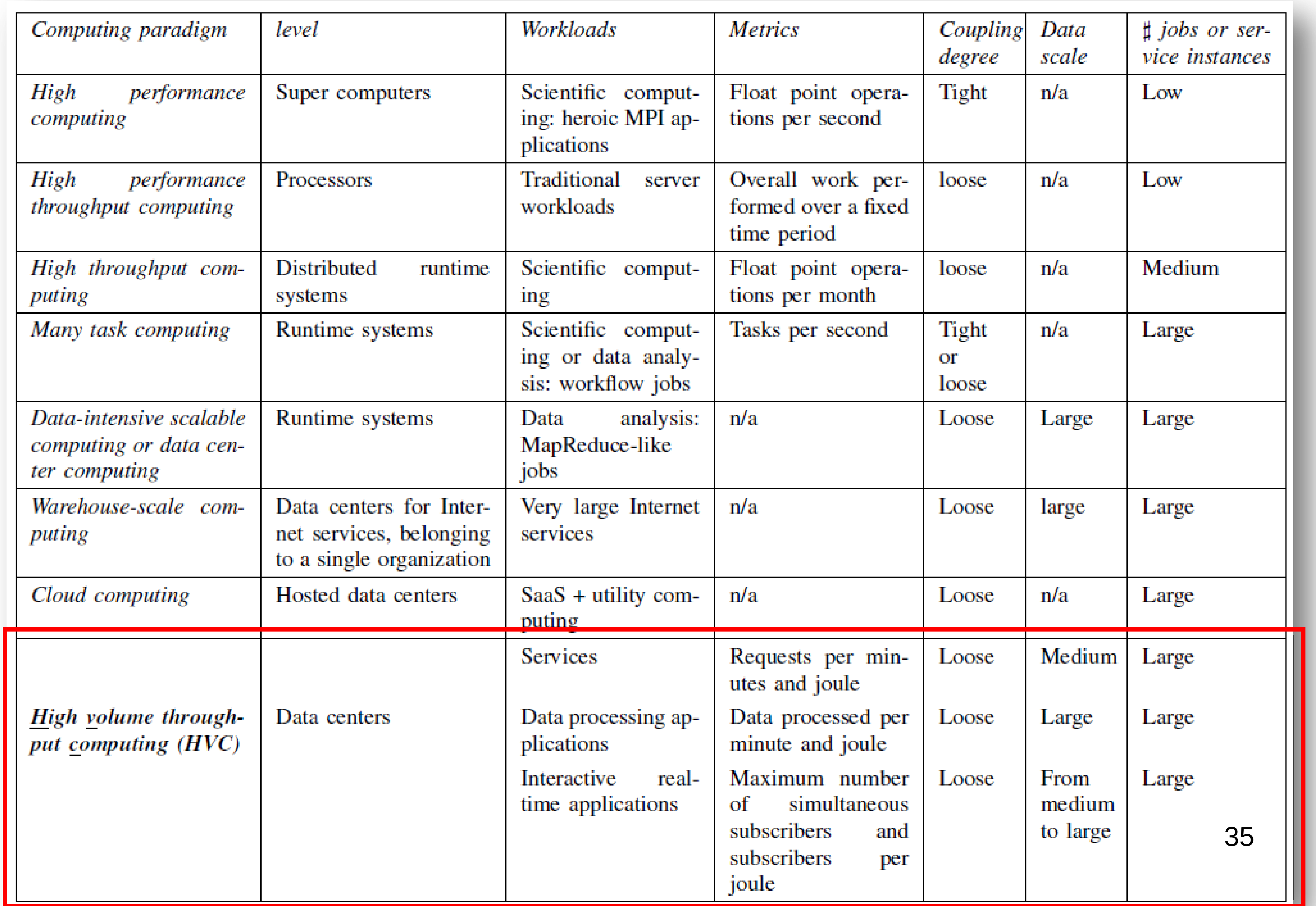
No Benchmark For HVC

Different benchmarks and metrics



Benchmark	*Domains*	*Level*	*Workloads*	*Metrics*
Linpack	High performance computing	Super computers	scientific computing code	Float point operations per second
SWap	Enterprise	Systems	Undefined	Performance/(space * watts)
Green 500	High performance computing	Super computers	Scientific computing code	Flops per watt
Graph 500	High performance computing	Super computers	Computing in the field of graph theory	Traversed edges per second (TEPS)
JouleSort	Mobile, desktop, enterprise	Systems	External sort	Records sorted per Joule
SPECpower _ssj2008	Enterprise	Systems	SPECjbb2005	Ssj_ops/watt
	Storage I/O	Storage systems	Transaction processing or scientific applications	I/O or data rates
SPECsfs2008	Network file systms	File servers	N/a	Operations per second and overall latency of the operations
HiBench	Data-intensive scalable computing	MapReduce runtime systems	Data analysis	Job running time and number of tasks completed per minute
GridMix2 or GridMix3	Data-intensive scalable computing	MapReduce runtime systems	Data analysis	Number of completed jobs and running time
WL Suite	Data-intensive scalable computing	MapReduce runtime systems	Data Analysis	n/a
YCSB or YCSB++	Warehouse-scale computing	NoSQL systems	Scale-out data services	Total operations per second and average latency per requests
PARSEC	n/a	Chip-Multiprocessors	Recognition, mining, synthesis, and mimic large-scale multithreaded commercial programs	n/a
TPC C/E/H	Throughput-oriented workloads	Server systems	Transaction processing and decision support	Application-specific
SPEC CPU2006	scientific and engineering applications	Processors	Serial programs	A ratio is calculated using the run time on the system under test and a SPEC-determined reference time

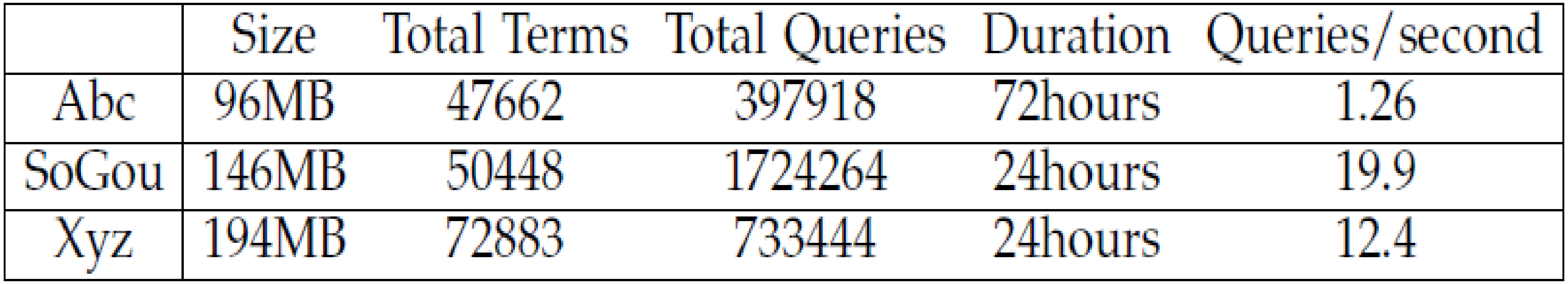
HVC is different from other paradigms



Computing paradigm	*level*	*Workloads*	*Metrics*	*Coupling degree*	*Data scale*	*♯ jobs or service instances*
High performance computing	Super computers	Scientific computing: heroic MPI applications	Float point operations per second	Tight	n/a	Low
High performance throughput computing	Processors	Traditional server workloads	Overall work performed over a fixed time period	loose	n/a	Low
High throughput computing	Distributed runtime systems	Scientific computing	Float point operations per month	loose	n/a	Medium
Many task computing	Runtime systems	Scientific computing or data analysis: workflow jobs	Tasks per second	Tight or loose	n/a	Large
Data-intensive scalable computing or data center computing	Runtime systems	Data analysis: MapReduce-like jobs	n/a	Loose	Large	Large
Warehouse-scale computing	Data centers for Internet services, belonging to a single organization	Very large Internet services	n/a	Loose	large	Large
Cloud computing	Hosted data centers	SaaS + utility computing	n/a	Loose	n/a	Large
***H**igh **v**olume through-put **c**omputing (HVC)*	Data centers	Services	Requests per minutes and joule	Loose	Medium	Large
		Data processing applications	Data processed per minute and joule	Loose	Large	Large
		Interactive real-time applications	Maximum number of simultaneous subscribers and subscribers per joule	Loose	From medium to large	Large

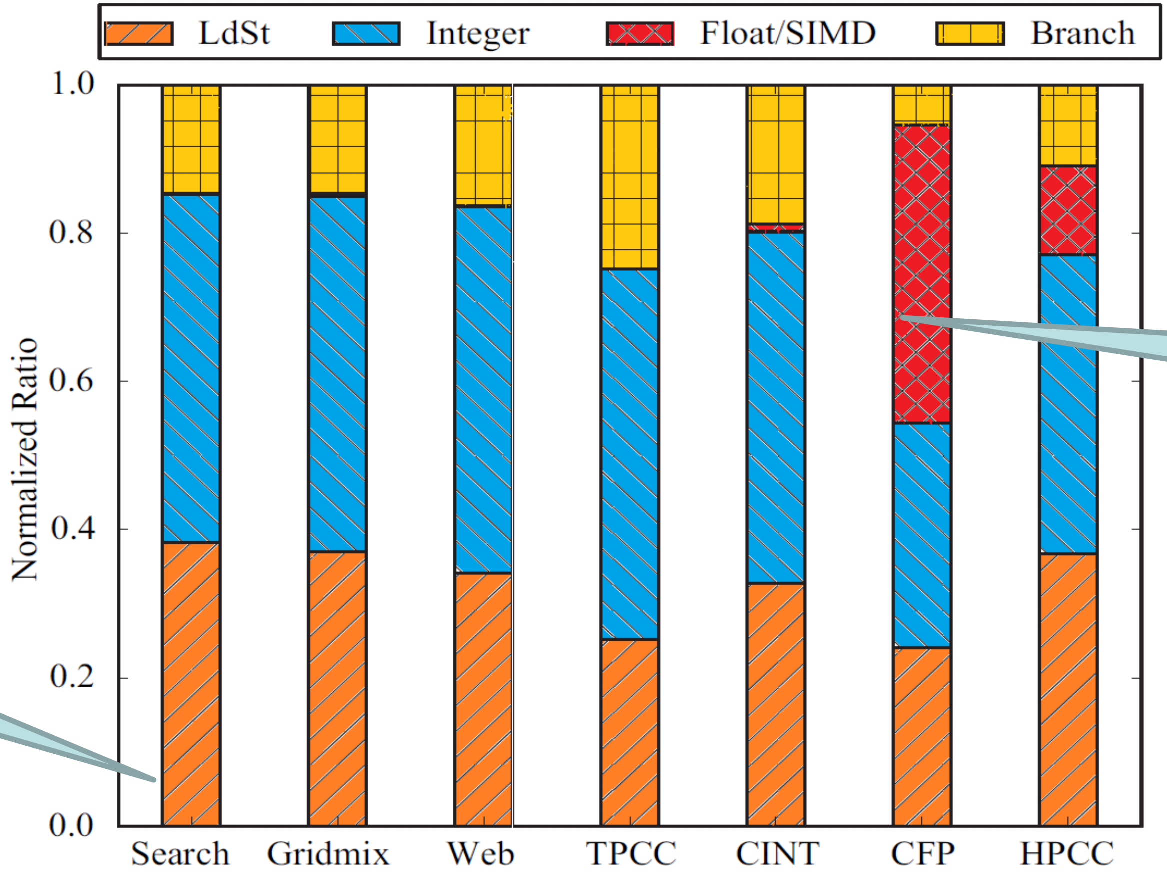
Case Study: Search
(Real workload traces)

- Understanding the behaviors of web search engines is becoming increasingly important to the design and deployment of data center systems。
- Three real traces (access logs) from search service providers.



	Size	Total Terms	Total Queries	Duration	Queries/second
Abc	96MB	47662	397918	72hours	1.26
SoGou	146MB	50448	1724264	24hours	19.9
Xyz	194MB	72883	733444	24hours	12.4

Instruction Mix



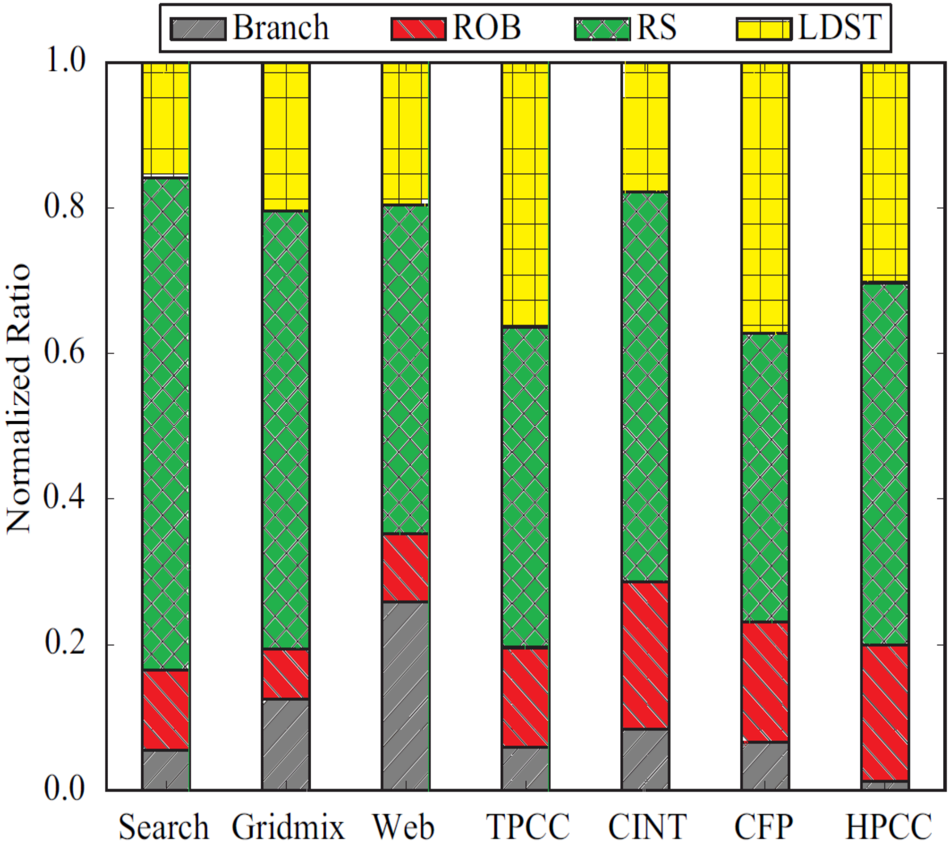
Float/ SIMD

load/ store

- **Search dose not have Float/SIMD operations**
- **Search has the highest percentage of load/store instructions**

Pipeline Stall

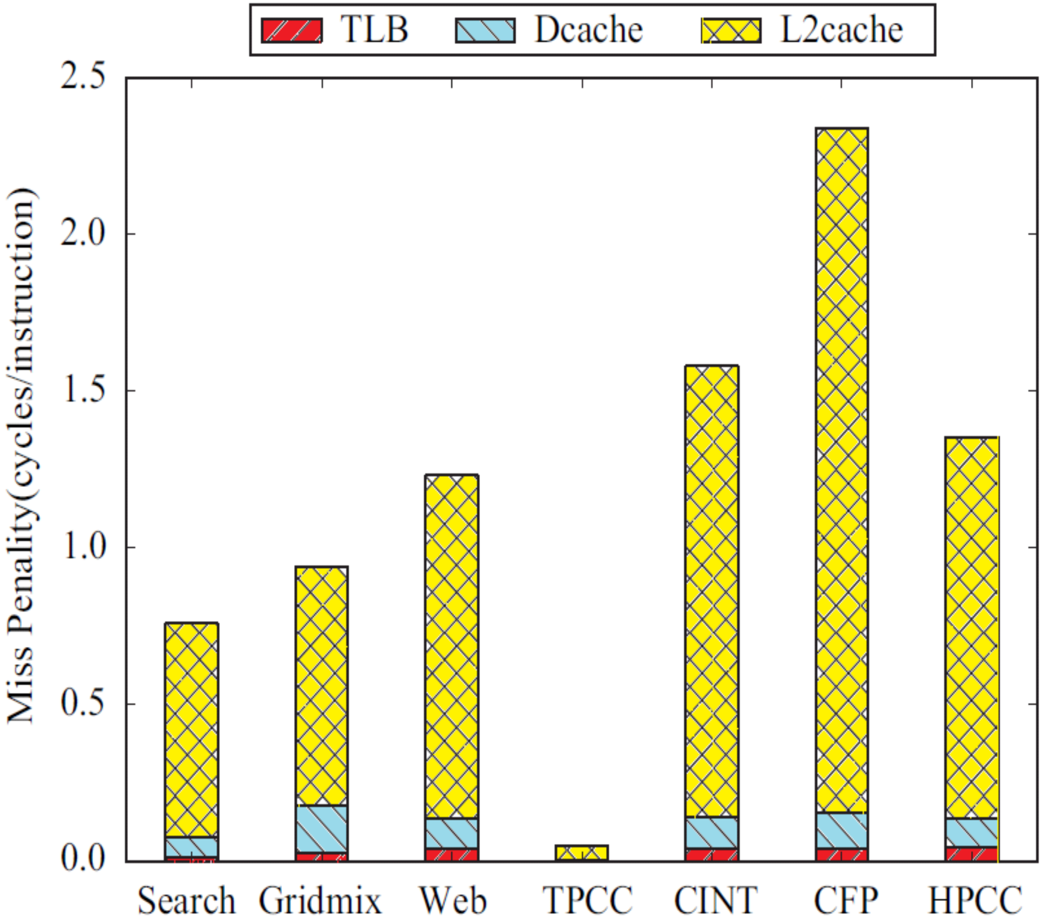
- Search has a low percentages of branch stalls (the smallest percentage of branch operations.)
- Search typically has a low percentage of load/store Stalls. (Search has a good cache performance)
- All benchmarks suffer significantly for the shortage of reservation stations.



***Branch*: Branch Miss Prediction**
ROB: Reorder Buffer Full
***RS:* Reservation Station Full**
***LDST:* Load/Store Buffer Full**

Cache Performance

- L2 cache misses have larger impact than L1 cache misses and TLB misses.
- The cache performance of the search engine is better than others but TPCC



Our Finding

- Our analysis reveals that, real-world query traces do not follow well-defined probability models, such as Poisson distribution and log-normal distribution.
- Synthetic traces do not accurately reflect the real traces. Need to abstract from the real workloads
- HVC workloads exhibit very different behavior with the existing benchmarks

Need New HVC Benchmarks

- Main-Stream systems work poorly for some HVC workloads
  - High i-cache miss rates
  - Ineffective “out of order” pipeline
  - Poor cache performance
- Require a set of benchmarks
  - to guide the design of architectures suitable for HVC workloads
  - Challenges
    - Difficult to obtain real data
    - Difficult to obtain real user behavior

What We Have Done on HVC Benchmarks

- We have built DCAngel: a workload characterization tool
  - Collect, analyze, and visualize a large number of performance metrics
  - Performance counters: CPI, average memory access latency, etc.
  - QoS measurements: response time of an individual query, etc.
- We have built a TestBed named Athena (http://prof.ncic.ac.cn/htc-testbed/) , which has been online since Auguest, 26, 2011.
  - The **TestBed** will provide real big data, application, and live workloads for research communities

Data Processing Unit

- Architecture for massive concurrent processing and support for a new execution model
  - Ability to save the status of a massive number of threads
  - Reduce the number of thread status at architecture-level
- Pursue data processing capability
  - Capability of processing a single datum is not a priority
  - Capability of processing a large volume of data in a unit time is the priority
- Novel heterogeneous multithreaded parallelism
  - On-chip heterogeneous parallelism: various processing cores to boost computing throughput
  - Reconfigurable parallel design: customized accelerators to reduce computing delay

Data Center Network

- Architecture for large-scale datacenter networks
  - Lightweight and simplified protocol
  - Simplified interconnection
  - Increasing the degree of coupling with processors
  - Increasing the degree of coupling with storage
- Approach
  - A large-scale full-system datacenter network simulator
  - Deep and direct support for applications in various network layers, such as protocol, NIC and switch
  - Novel NIC: simplifying and accelerating protocols to reduce end-to-end delay
  - Novel Switch/Router: aware of dataflow traffic and adjust accordingly

Heterogenous Operating System

- Rebuilding OS with the principles of distributed systems, and providing new OS abstraction
  - Reducing complexity of system management
  - Increasing system programmability
  - Enhancing heterogeneous resources sharing
- Incubating software ecosystem from the very beginning
  - Operating system
  - Data management system
  - Resource management system
  - Environment for application demonstration

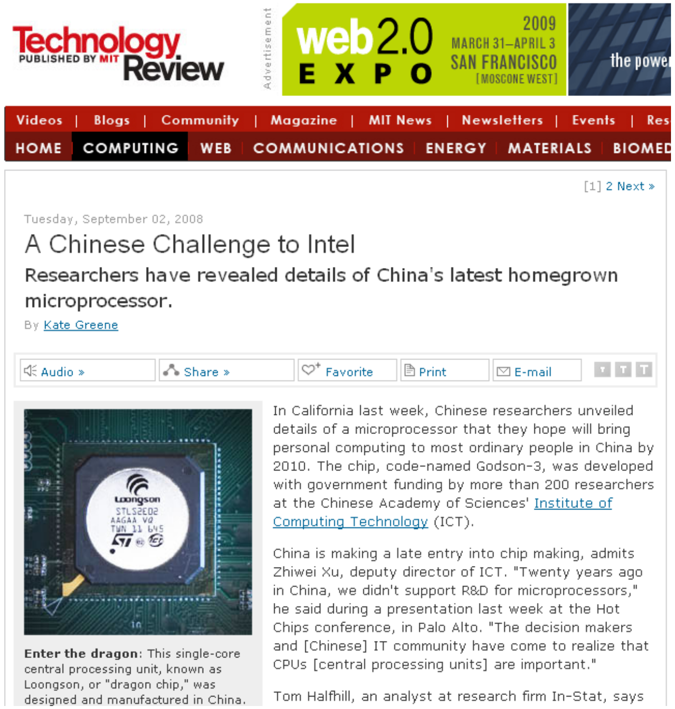
Unified Memory System

- Unified Memory System with DRAM, emerging storage media and disk
  - Unifying API for multiple memory and storage devices
  - Raising the position of the storage system in the hierarchy
  - Shortening the distance between the processor and the memory/storage
  - Broadening the path between processor and memory/storage
- Research topics
  - Analyzing characteristics of typical I/O workloads in cloud environment
  - Understanding new memory/storage devices (Flash, SSD, PCM, etc.)
  - Designing new data accessing interface
  - Unified Memory System in one single node
  - Distributed Memory System

Elastic Processor

Godson (Loongson) CPU

- Godson （the commercial brand is Loongson) is general purpose CPU designed by ICT/CAS and Loongson Corp, which is one of 16 Major program of National Mid-long Term S&T Plan, each fund USD 5-10B during 2006-2020
- Loongson-3B： 8-cores, 2*256-bit Vector Ext. per core, 65nm, 585M transistors , 300mm^2, 1GHz, 128GFlops@40W, product chip for supercomputer.
- Loongson-3C, 32nm, 1.1Billion transistors 1.5GHz, 192GFlops, 25.6GBps bandwidth. 8M Cache, tape-out now
- Research results on Godson CPU have been published in ISSCC, ISCA, HPCA, SPAA, DATE, HotChips, IEEE Trans. on Computer, IEEE Micro, etc


Technology Review PUBLISHED BY MIT
Videos | Blogs | Community | Magazine | MIT News | Newsletters | Events | Res
HOME COMPUTING WEB COMMUNICATIONS ENERGY MATERIALS BIOMED
[1] 2 Next »
Audio » Share » Favorite Print E-mail

Tuesday, September 02, 2008

A Chinese Challenge to Intel

Researchers have revealed details of China's latest homegrown microprocessor.

By Kate Greene

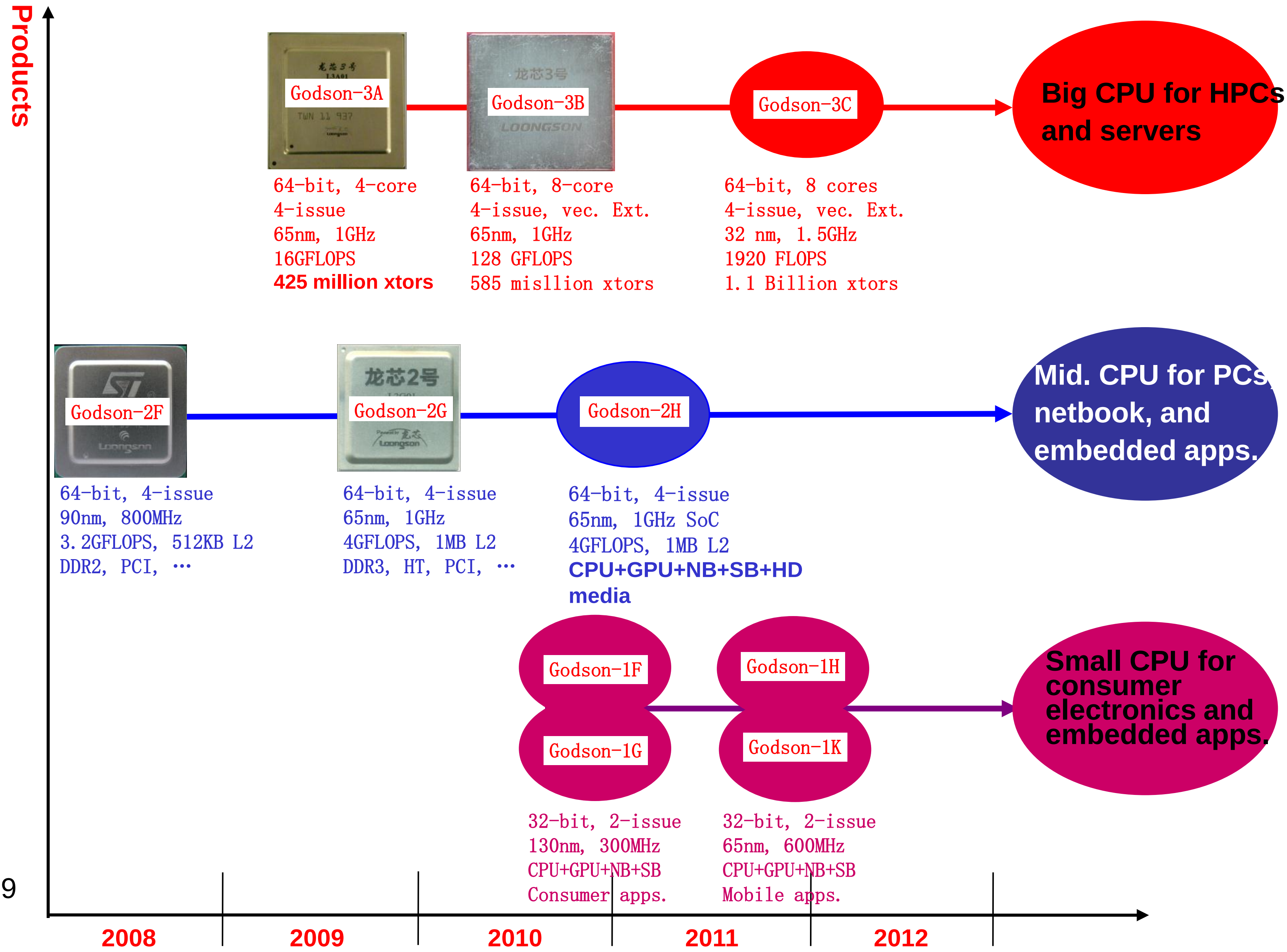
Enter the dragon: This single-core central processing unit, known as Loongson, or "dragon chip," was designed and manufactured in China.

In California last week, Chinese researchers unveiled details of a microprocessor that they hope will bring personal computing to most ordinary people in China by 2010. The chip, code-named Godson-3, was developed with government funding by more than 200 researchers at the Chinese Academy of Sciences' Institute of Computing Technology (ICT).

China is making a late entry into chip making, admits Zhiwei Xu, deputy director of ICT. "Twenty years ago in China, we didn't support R&D for microprocessors," he said during a presentation last week at the Hot Chips conference, in Palo Alto. "The decision makers and [Chinese] IT community have come to realize that CPUs [central processing units] are important."

Tom Halfhill, an analyst at research firm In-Stat, says

Godson CPU Roadmap



Big CPU for HPCs and servers

64-bit, 4-core
4-issue
65nm, 1GHz
16GFLOPS
425 million xtors

64-bit, 8-core
4-issue, vec. Ext.
65nm, 1GHz
128 GFLOPS
585 misllion xtors

64-bit, 8 cores
4-issue, vec. Ext.
32 nm, 1.5GHz
1920 FLOPS
1.1 Billion xtors

Mid. CPU for PCs, netbook, and embedded apps.

64-bit, 4-issue
90nm, 800MHz
3.2GFLOPS, 512KB L2
DDR2, PCI, …

64-bit, 4-issue
65nm, 1GHz
4GFLOPS, 1MB L2
DDR3, HT, PCI, …

64-bit, 4-issue
65nm, 1GHz SoC
4GFLOPS, 1MB L2
CPU+GPU+NB+SB+HD media

Small CPU for consumer electronics and embedded apps.

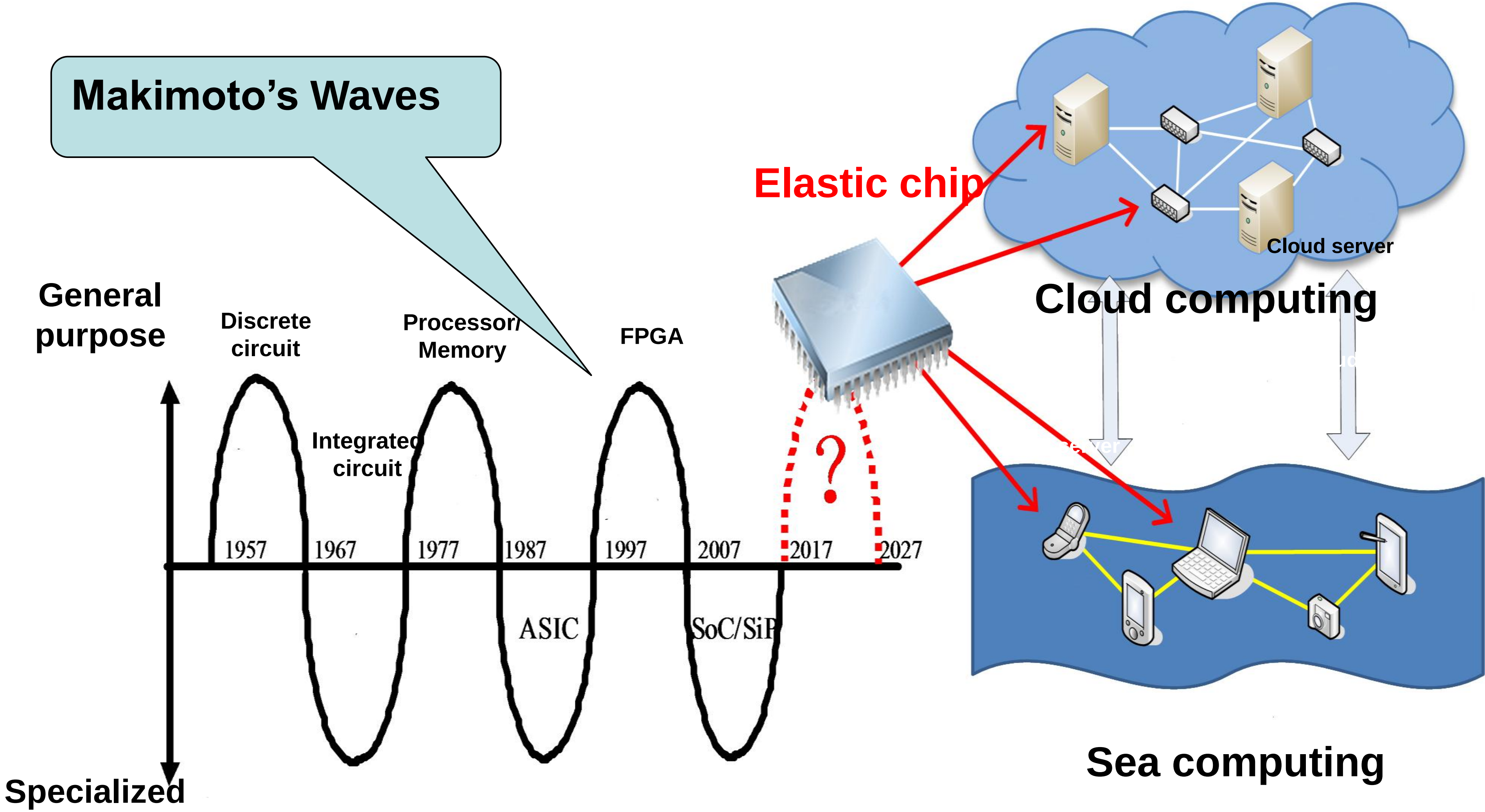
32-bit, 2-issue
130nm, 300MHz
CPU+GPU+NB+SB
Consumer apps.

32-bit, 2-issue
65nm, 600MHz
CPU+GPU+NB+SB
Mobile apps.

Elastic chip for solving "the Class Insecta Paradox"

- Much application classes for Internet of Things and mobile applications
  - ◆ Dedicated energy-efficient chip for each class? No!
  - ◆ General purpose processor for all class? No!
- How to deal with the various niche applications, so called "the Class Insecta Paradox"?
  - – Current IT: mammals-like (about 5000 species)
  - – Future IT: insects-like (about 50 millions species)
- Elastic chip: Achieving high energy-efficiency for most applications through adapting to concrete application class.

Development Cycles of IC



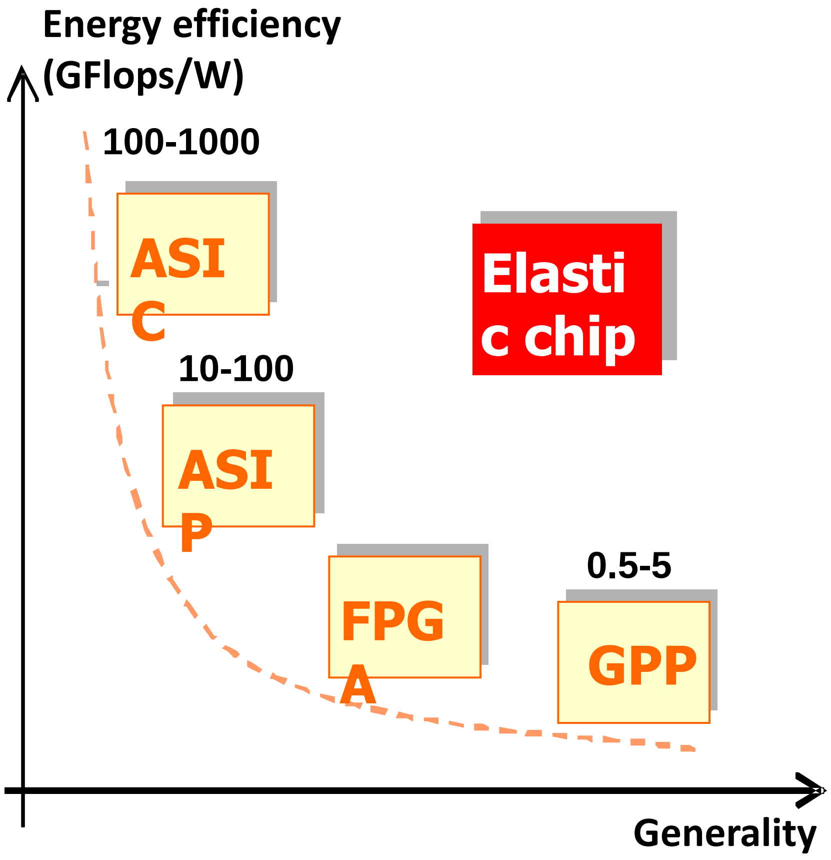
Elastic Chip: Objective

Objective

- A single chip which can cope with most applications with 100-1000x energy efficiency of current GPP

Scientific Problems

- How to achieve run-time elasticity?
- What elastic architecture can adaptive to most applications?
- How to select the most suitable hardware configuration for an application?
- How to characterize the software-hardware relativity of elastic chip?



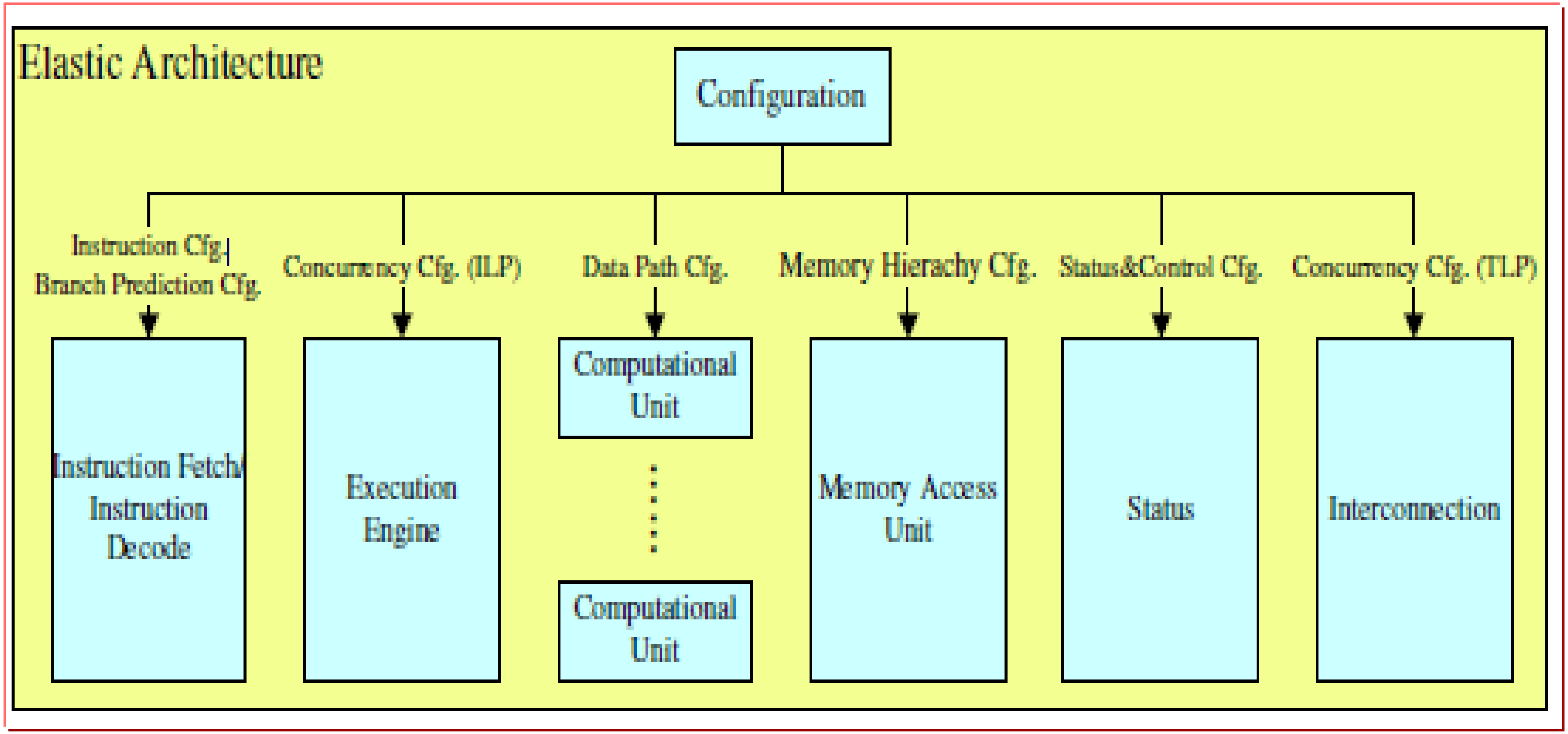
Elastic Architecture

- Reconfiguration is a pervasive characteristics in existing architectures
  - e.g., Dynamic Frequency Scaling, Core Enable, etc.
- Elasticity of reconfigurable architectures
  - Ratio of the worst- and best-case responses(e.g., performance/power/EDP)
- A flexible architecture can offer large elasticity to adapt to different applications
- Reconfigure architecture for different applications
  - Reconfigurable components
  - ISA, Branch Predictor, Data Path, Cache, etc.

Design Methodology

- Application-specific Design Space Exploration
  - – Constructing predictive models to find the best/worst architecture for each application
    - Training phase for model construction
    - Predicting phase to find the appropriate architecture
- Implementation Issues
  - – Configurable buffers/queues
  - – Flexible RAM for configurable cache/BHT/BTB etc.

Elastic Architecture



More Than 70 Millions Choices

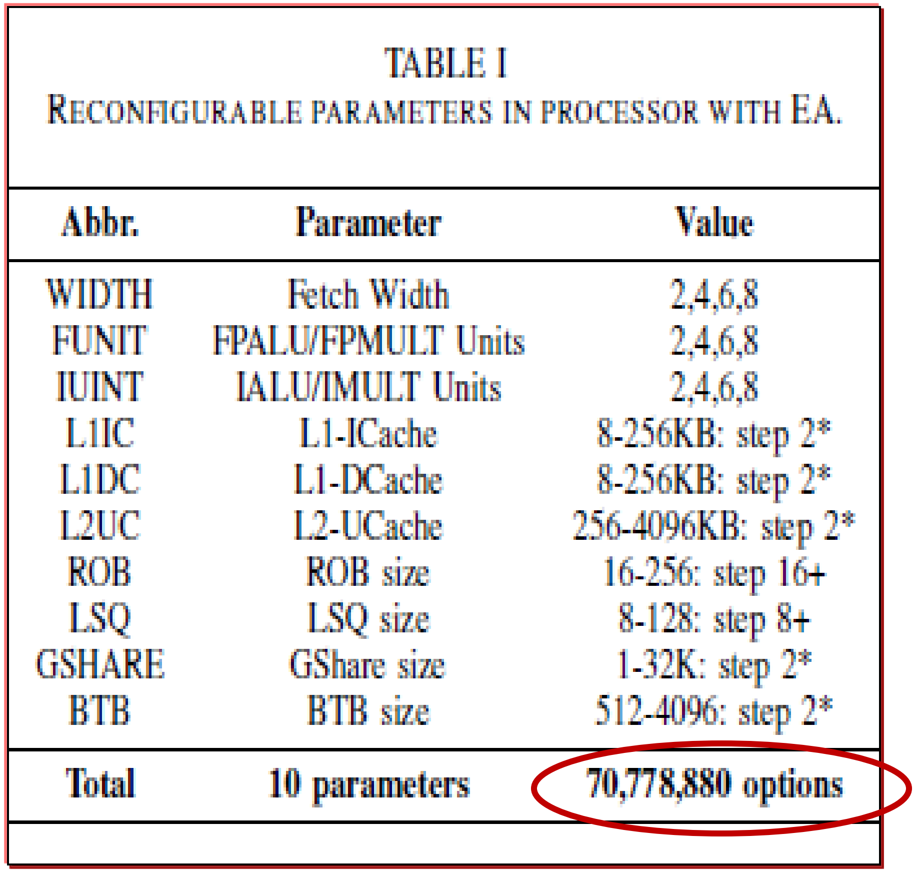


TABLE I
RECONFIGURABLE PARAMETERS IN PROCESSOR WITH EA.

Abbr.	Parameter	Value
WIDTH	Fetch Width	2,4,6,8
FUNIT	FPALU/FPMULT Units	2,4,6,8
IUINT	IALU/IMULT Units	2,4,6,8
L1IC	L1-ICache	8-256KB: step 2*
L1DC	L1-DCache	8-256KB: step 2*
L2UC	L2-UCache	256-4096KB: step 2*
ROB	ROB size	16-256: step 16+
LSQ	LSQ size	8-128: step 8+
GSHARE	GShare size	1-32K: step 2*
BTB	BTB size	512-4096: step 2*
Total	**10 parameters**	**70,778,880 options**

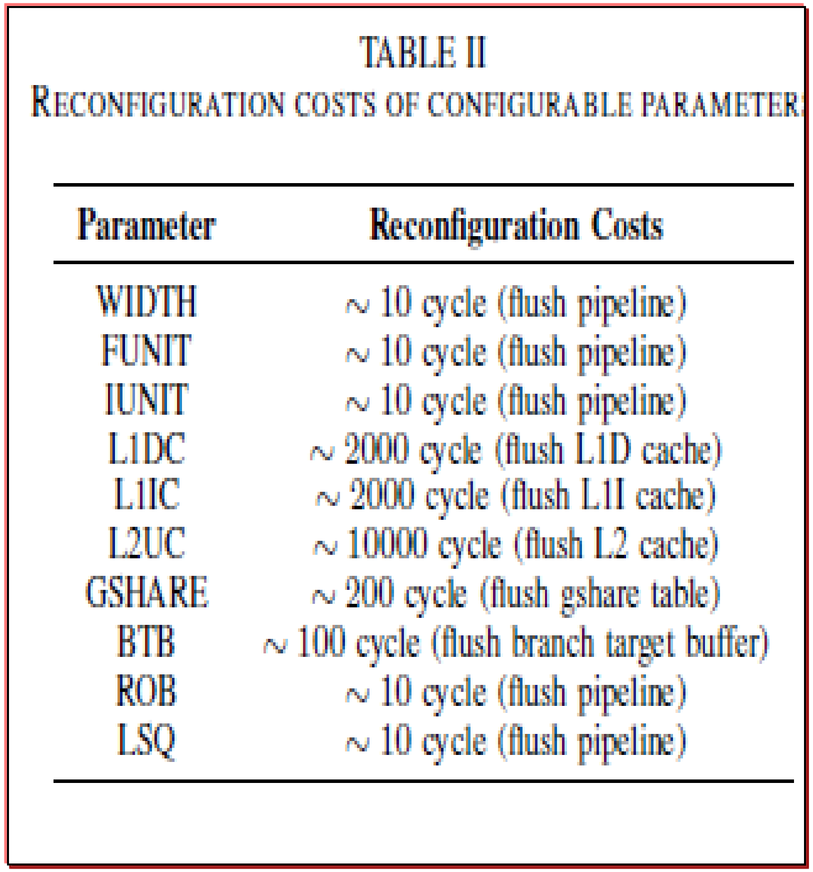
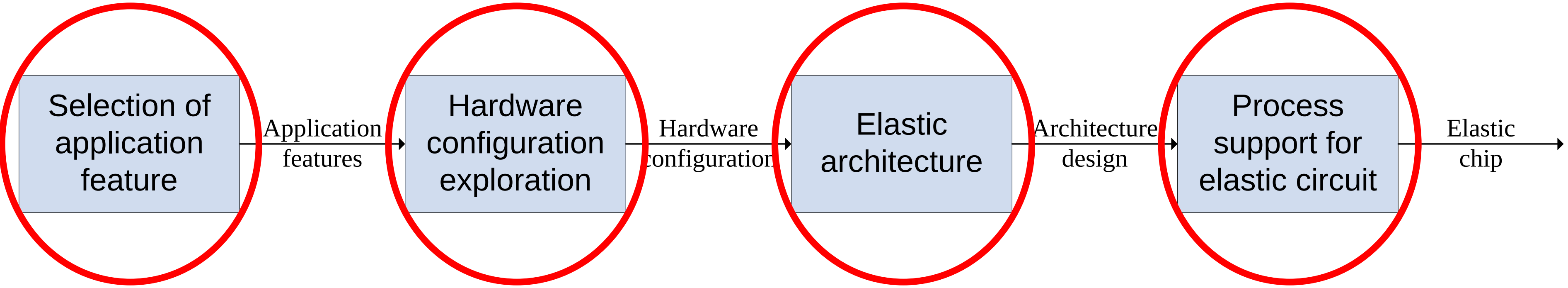


TABLE II
RECONFIGURATION COSTS OF CONFIGURABLE PARAMETER

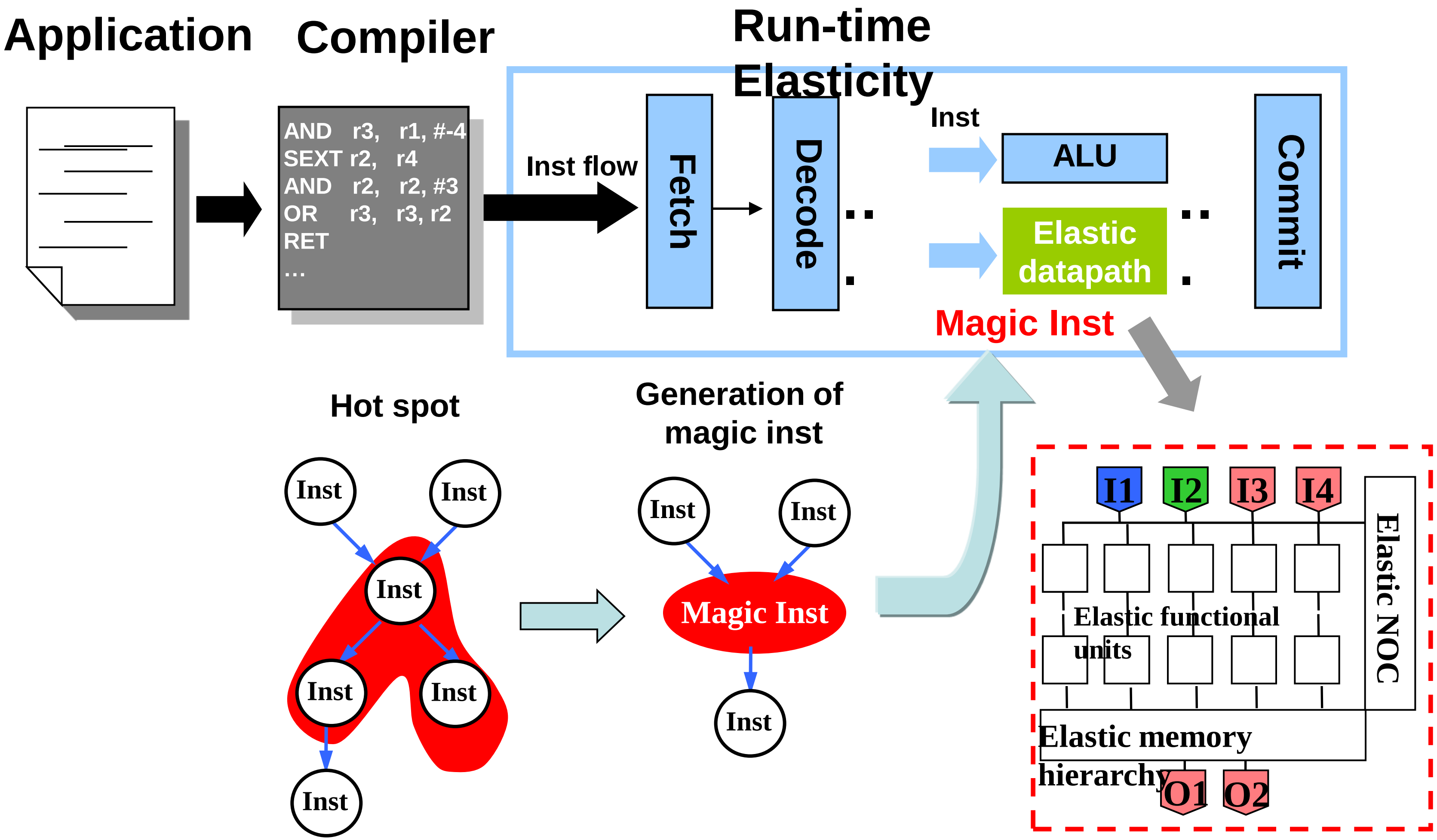
Parameter	Reconfiguration Costs
WIDTH	∼ 10 cycle (flush pipeline)
FUNIT	∼ 10 cycle (flush pipeline)
IUNIT	∼ 10 cycle (flush pipeline)
L1DC	∼ 2000 cycle (flush L1D cache)
L1IC	∼ 2000 cycle (flush L1I cache)
L2UC	∼ 10000 cycle (flush L2 cache)
GSHARE	∼ 200 cycle (flush gshare table)
BTB	∼ 100 cycle (flush branch target buffer)
ROB	∼ 10 cycle (flush pipeline)
LSQ	∼ 10 cycle (flush pipeline)

Elastic Chip: Roadmap


Selection of application feature
Application features
Hardware configuration exploration
Hardware configuration
Elastic architecture
Architecture design
Process support for elastic circuit
Elastic chip

- Finding out the most important hardware-independent features of applications
- Based on application features, exploring the hardware configuration space to find out the most suitable hardware configuration for the current applications
- The hardware configuration is enabled with run-time elasticity
- Dedicated process support to implement run-time elasticity

Elastic Chip: Run-time Elasticity

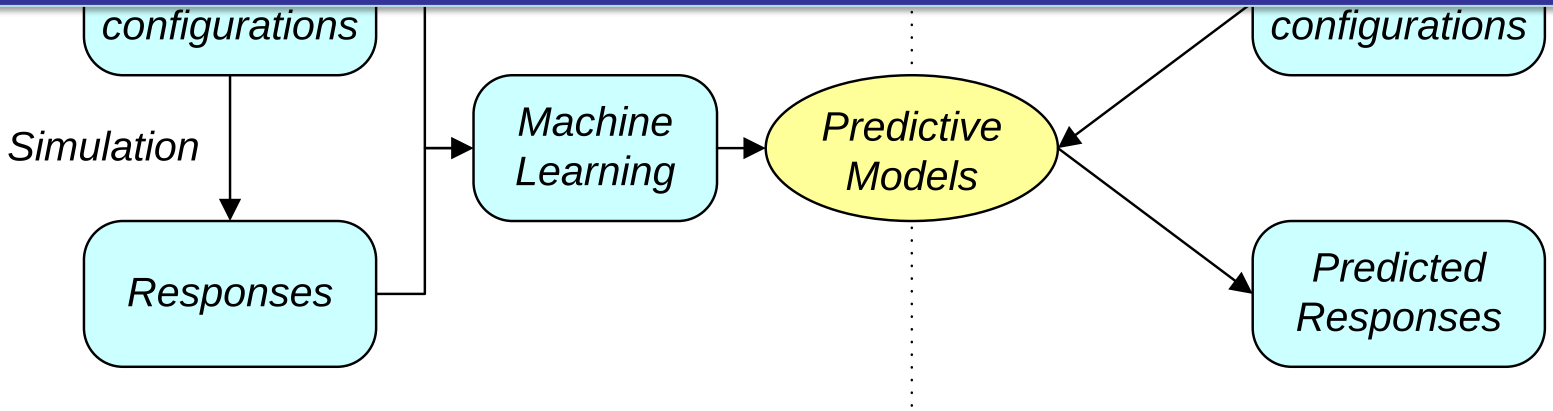


```
AND   r3,   r1, #-4
SEXT r2,   r4
AND   r2,   r2, #3
OR     r3,   r3, r2
RET
…
```

Elastic Chip: Hardware Configuration Selection

- Billions of possible hardware configurations for an application
  - Thread number, cache size, functional units, memory bandwidth, magic inst, ……

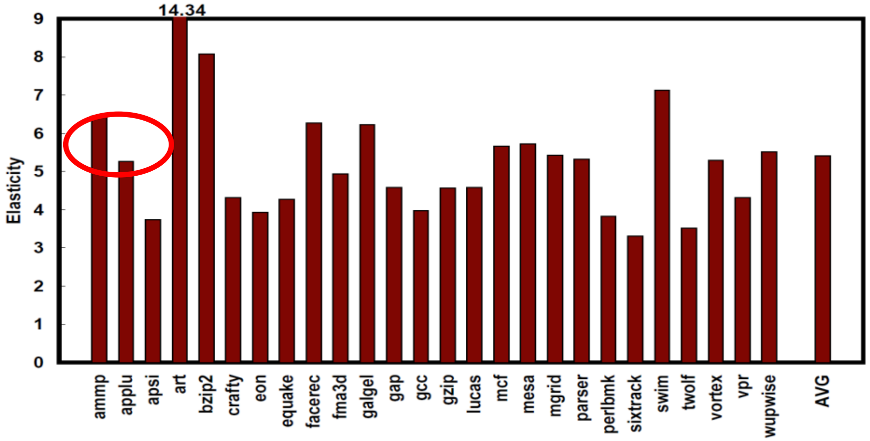
The performance/energy of hardware configurations can be predicted with effective machine learning techniques.



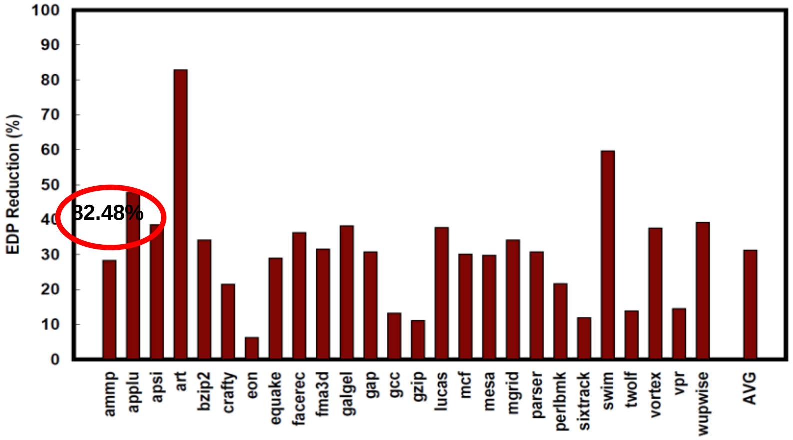
Preliminary Results

- 10 reconfigurable components
  - Issue Width, F/I-UNIT, L1/L2 Caches etc.
  - >70 million architecture instances

Elasticity Reduction



EDP

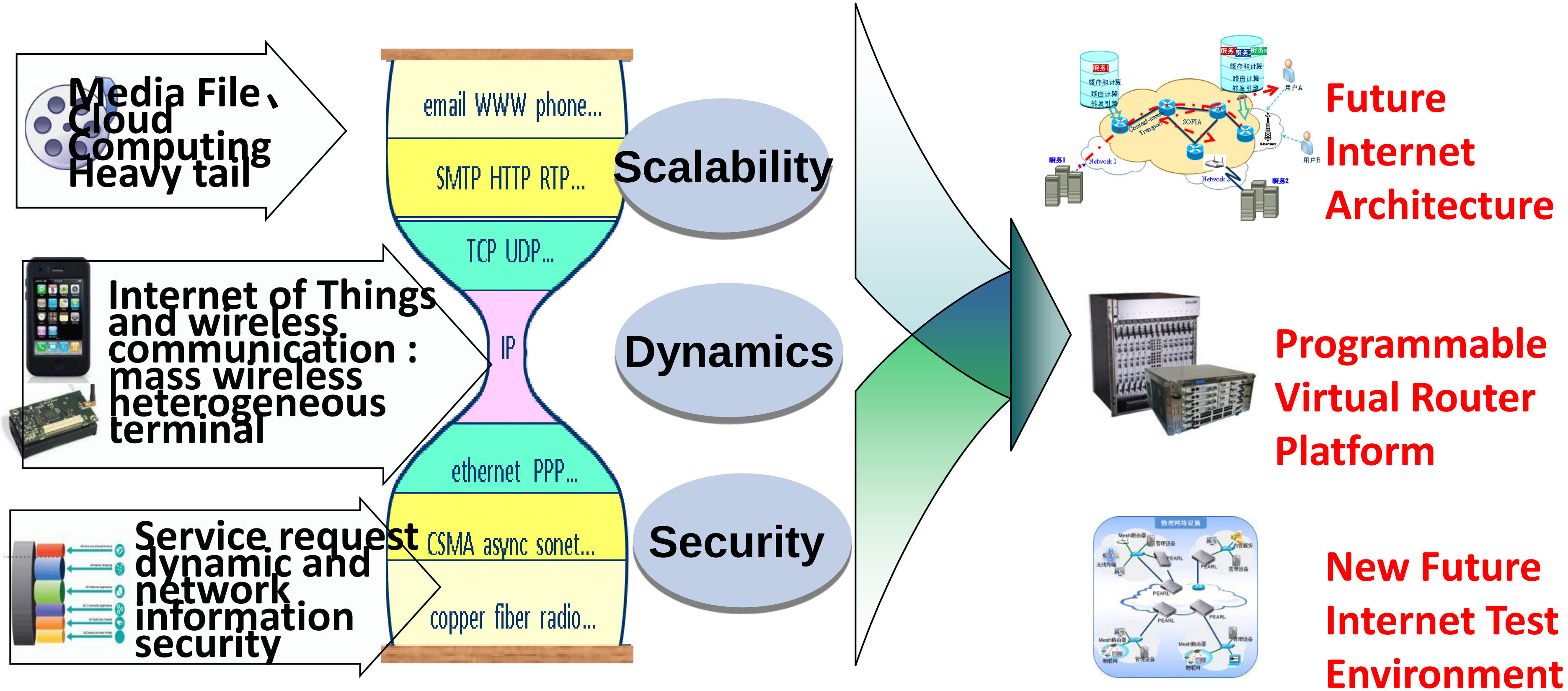


Q. Guo, T. Chen, Y. Chen, Z. Zhou, W Hu, Z. Xu, Effective and Efficient Microprocessor Design Space Exploration Using Unlabeled Design Configurations, IJCAI 2011.

Future Internet Research in ICT/CAS

Future Internet Research at a Glance

Requirements **Problems** **Our Research**



Media File、Cloud Computing Heavy tail

Internet of Things and wireless communication : mass wireless heterogeneous terminal

Service request dynamic and network information security

Scalability

Dynamics

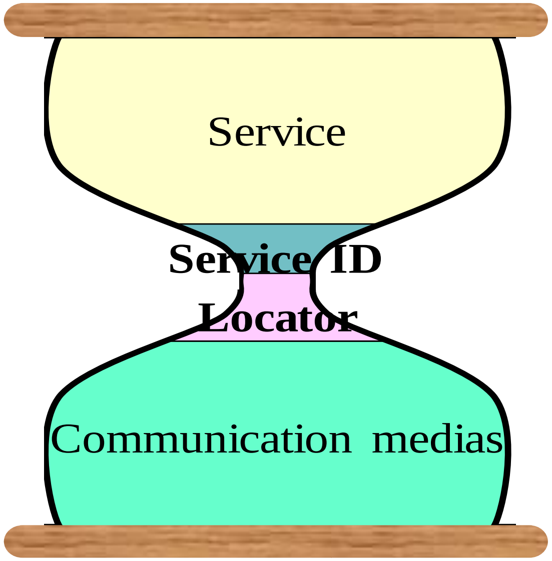
Security

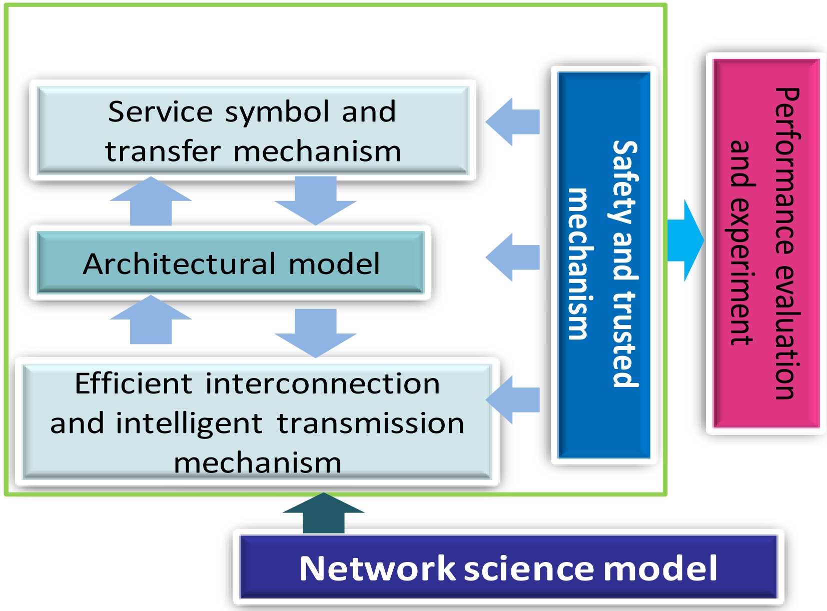
Future Internet Architecture

Programmable Virtual Router Platform

New Future Internet Test Environment

Service Oriented Future Internet Architecture (SOFIA)

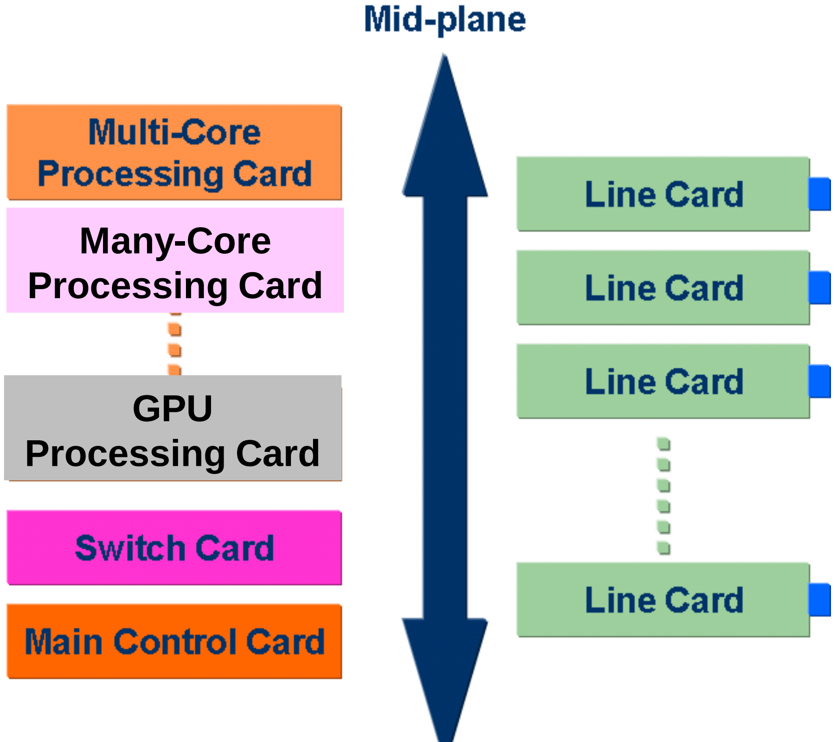


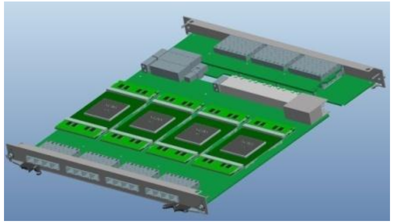

Service symbol and transfer mechanism
Architectural model
Efficient interconnection and intelligent transmission mechanism
Safety and trusted mechanism
Performance evaluation and experiment
Network science model

- **Based on cloud computing and the development trend of communications, propose Service Oriented Future Internet Architecture (SOFIA)**
  - **Abandon TCP/IP, directly connect user and service**
  - **Separate identity and location, provide the network with the capability of storage and processing**
  - **Perception between service and network, provide service level security and**

Service-Oriented Future Internet Architecture (SOFIA), IEEE INFOCOM (Poster), 2011

Programmable Virtual Router Platform (PEARL）

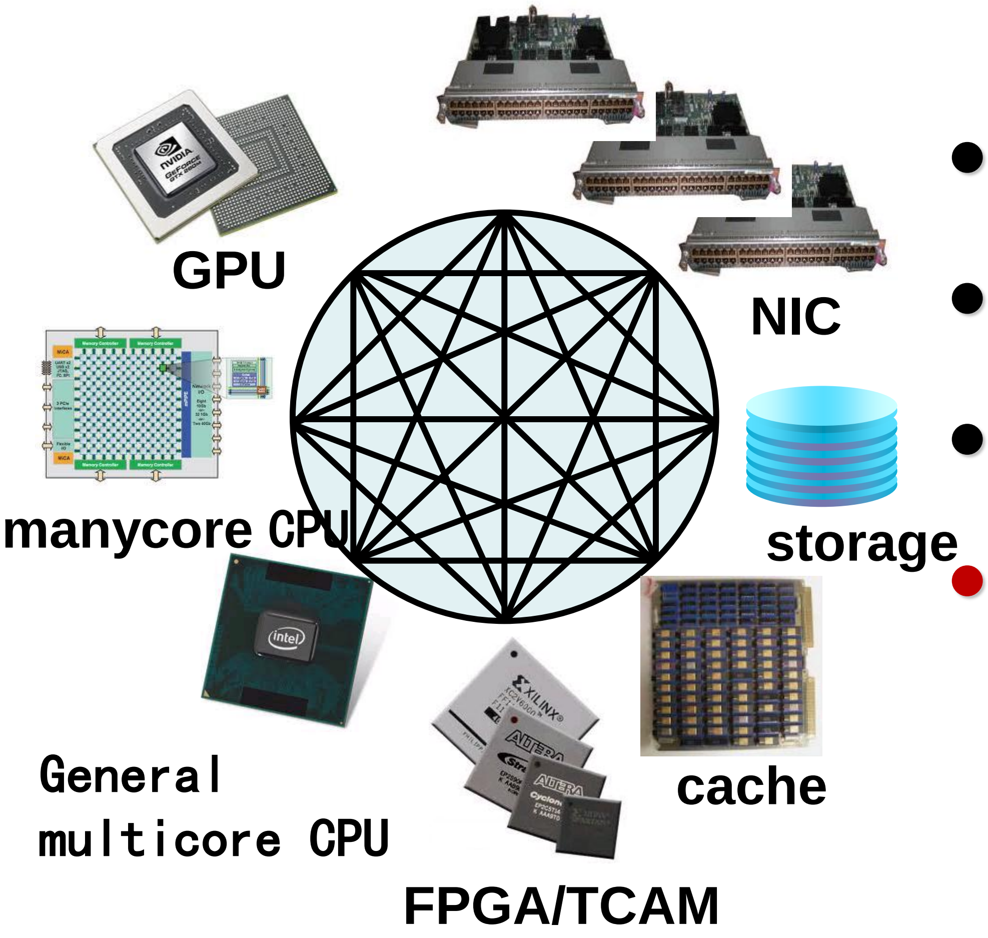




- Application
  - Future Internet Testbed
  - Date Center Network
- Overview
  - 14 slots per shelf， dual-star fabric (base & fabric)
  - Backbone bandwidth: 480 Gbps
  - More than 128 virtual router Instances can be supported
  - High performance, good scalability, and strong isolation
  - Support: OpenFlow、IPv4/v6、SOFIA、NDN
- Challenges
  - New programmable router architecture
  - Programmable and high-performance packet processing approaches
- Verification system has already be finished，PEARL II will be tested in 2012

PEARL: A Programmable Virtual Router Platform, IEEE Communications Magazine, Future Internet Architectures: Design and Deployment Perspectives,

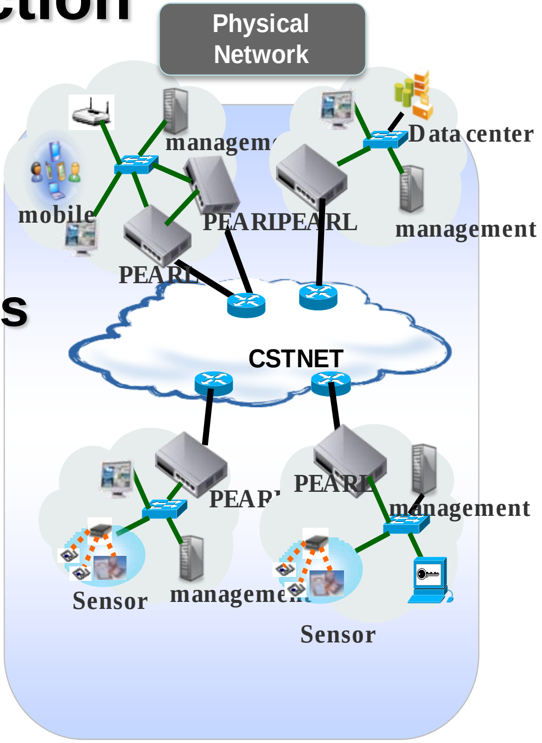
PEARL II: Main Features



- High-throughput packet switching, and various high speed network interfaces
- Flexible and programmable packet processing in the data plane
- Programmable interconnected structure
- Programmable protocol processing in the control plane
- GPU、Many Core and other accelerated chips can be equipped flexibly， achieving scalable high performance and low power consumption

China Experiment Environment for Future Network Innovation (CENI)

- Set up a virtual, programmable, measurable, scalable network experiment platform to speedup the future network research and national innovative environmental infrastructure construction
  - Explore future network architecture
  - Research of crucial technology
  - Equipment development and business innovation
  - Uncover the mechanism of the interactions between systems
- We can test
  - Future network architecture
  - Internet of Things
  - Cloud Computing
  - Optical network
  - Network merging
  - Network Security



Acknowledgements

Thanks to the following colleague for providing valuable information:

- CAS FIT project
  - zxu@ict.ac.cn
- High volume throughput computer
  - zhanglixin@ict.ac.cn
- Thousands threads processor
  - fandr@ict.ac.cn
- Elastic processor
  - chenyj@ict.ac.cn
- New router of future Internet
  - xie@ict.ac.cn

Zhiwei Xu

Lixin zhang

Dongrui Fan

Yunji Chen

Gaogang Xie

Thank you !

Institute of Computing Technology
Chinese Academy of Sciences