中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

智能超算（iHPC）

徐志伟
中国科学院计算技术研究所
http://novel.ict.ac.cn/zxu/
zxu@ict.ac.cn

Xu, Chi, Xiao. High-Performance Computing Environment: A Review of Twenty Years Experiments in China. *National Science Review*, Volume 3, Issue 1, 1 March 2016, Pages 36–48, https://doi.org/10.1093/nsr/nww001

提纲

- 十九大报告：2035年进入创新性国家前列
- 超级计算领域的发展方向是什么？
  - 观点：智能超算是重要方向
  - 论据
    - 历史趋势：三个二十年
    - 科学计算的新需求
    - 本质原因：科学探索需要三类推理
  - 智能超算的内涵

超级计算的方向

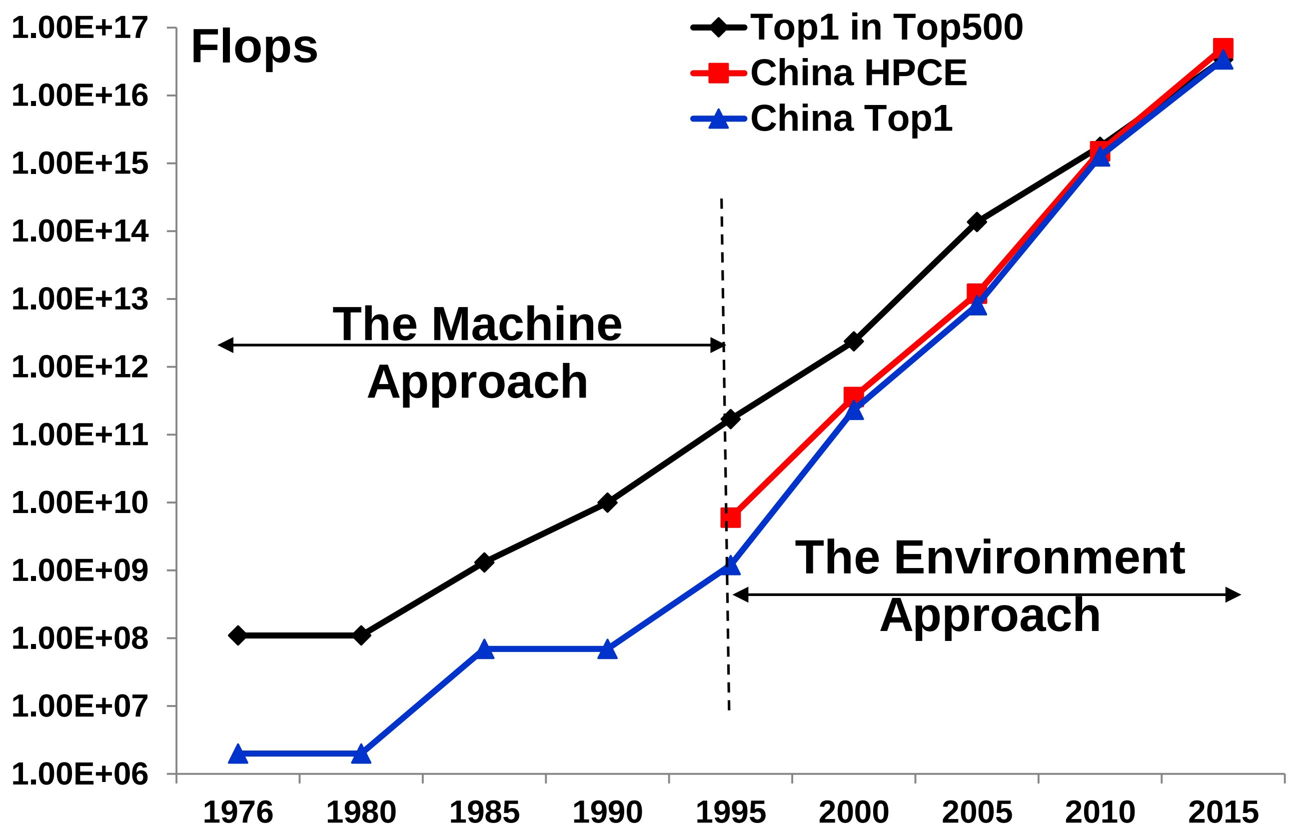
- 近期方向很明确
  - Exascale systems and applications, 50 GOPS/W by 2022
  - 中美欧日都有部署
- 中长期方向尚在探索中
  - ECBD Workshop, 无锡，March 2017
  - 《ECBD Pathway to Convergence》，2017
  - 历史上看，需要提前15-20年布局：Petaflops, HPCS, Exascale
  - **这一次中国应该发挥重要作用**

	2010	**2020**	**2025**	**2035**
Top500	DoE Jaguar 1.76 POPS@6.75MW **0.253 GOPS/W**	1000 POPS@20MW **50 GOPS/W**	?	?
Green500	Dawning Nebula 1.27 POPS@2.58MW 0.492 GOPS/W	200 GOPS/W？	?	?

中国超算的三个20年

Xu, Chi, Xiao. High-Performance Computing Environment: A Review of Twenty Years Experiments in China. *National Science Review*, Volume 3, Issue 1, 1 March 2016, Pages 36–48, https://doi.org/10.1093/nsr/nww001

- 1976-1995
  - Machines
- 1995-2015
  - Machines
  - Environment
- 2015-2035
  - 智能超算环境



Growth trends of top HPC systems in China and in the World: before and after 1995

Flops Growth	1976 to 1995		1995 to 2015	
	Speed Increase	Annual Growth Rate	Speed Increase	Annual Growth Rate
China	600 times	40%	28 million times	136%
World	1550 times	47%	0.2 million times	84%

China's Public HPC Evolution

Attribute	1995	2000	2005	2010	2015
HPCE mode	Isolated	Interconnected	Grid services	Grid services	GS+Domains
No. of Sites	5	7	8	14	17
OS	Self Made	AIX, Linux	Linux	Linux	Linux
Middleware	N/A	NHPCE	CNGrid GOS	CNGrid GOS	CNGrid SCE
Total Speed	0.01 Tflops	0.6 Tflops	18 Tflops	3.4 PFlops	62.6 PFlops
Total Disk	0.08 TB	5.4 TB	200 TB	17.6 PB	34.6 PB
No. of Apps	100	Hundreds	10 domains	450	500
No. of Users	Dozens	Hundreds	Hundreds	Thousands	Thousands
Publications	A few	Dozens	Dozens	Hundreds	Hundreds
Ph.D. Awardees	A few	15	30	45	50
Funding / year	12M yuan	20M yuan	90M yuan	440M yuan	770M yuan

The most important progress is community. 100K users by 2035?

A Partial List of Open Source Contributions

- DCFS, http://www.ncic.ac.cn/dcfs/
- OpenBLAS, http://www.openblas.net/
- OpenCVCL, http://opencv.org/
- yaSpMV, https://code.google.com/p/yaspmv/
- Hadoop+, https://github.com/ict-carch/hadoop-plus
- Loongcc, http://svn.open64.net
- FunctionFlow, https://github.com/AthrunArthur/functionflow
- LiveRender, https://github.com/llfjfz/LiveRender
- NightWatch, https://github.com/grtoverflow/PC-Malloc
- Mammoth, https://issues.apache.org/jira/browse/MAPREDUCE-5605
- Frog, https://github.com/AndrewStallman/Frog
- Giraffe, https://github.com/haohonglin/Giraffe
- CCIndex, https://github.com/ICT-Ope/CCIndex_HBase_0.90.0
- RCFile, https://en.wikipedia.org/wiki/RCFile
- DataMPI, http://DataMPI.org

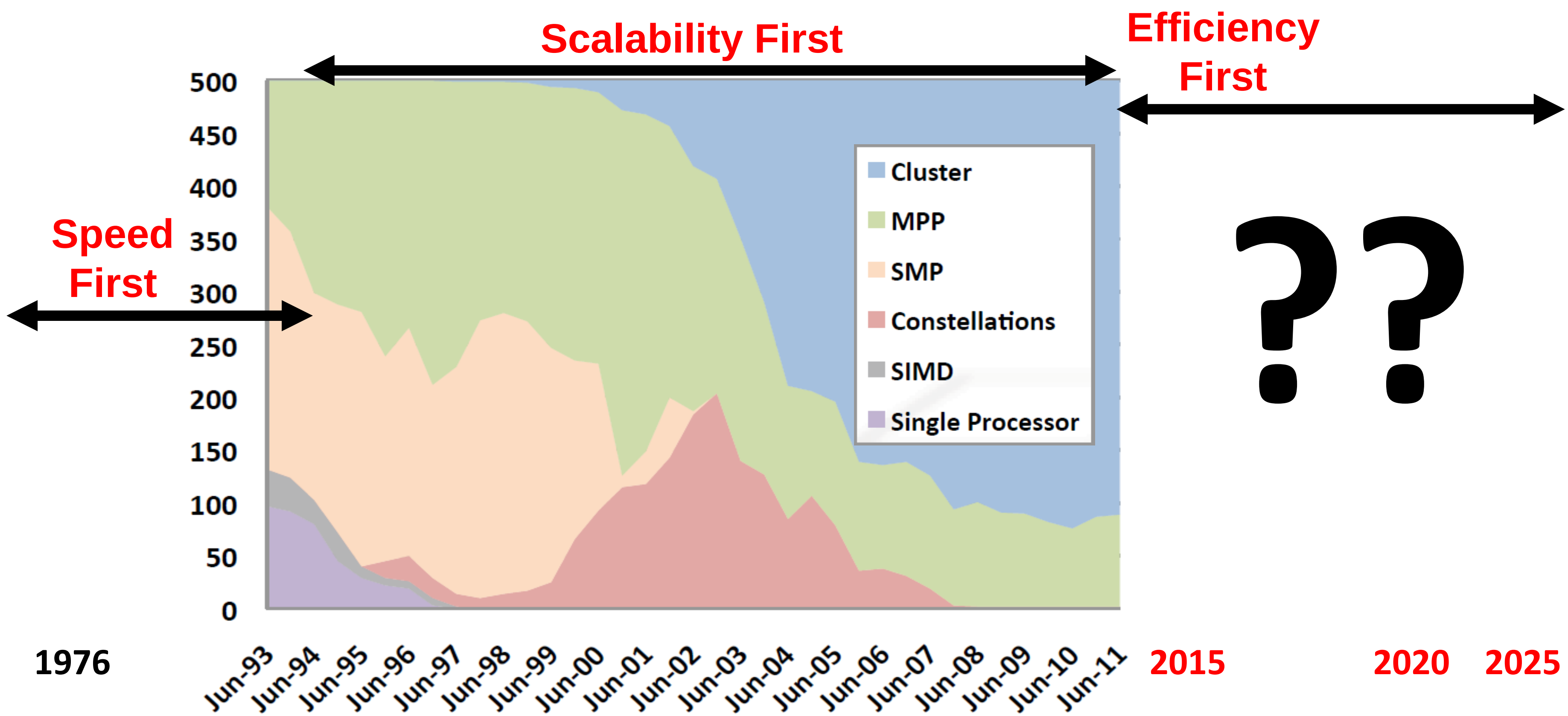
Application Performance Still Slow

Application	Gflops	CPU%	Memory%	DiskIO (MB/s)	EthIO (MB/s)	IBIO (MB/s)
VASP	170.369	87.62	99.57	0.2	0.941	97.918
Castep	136.279	75.66	98.54	1.603	0.13	228.292
MATLAB	135.992	55.55	87.6	0.112	0.253	3.657
Dmol	133.296	85.73	72.89	10.115	2.021	6.561
Gaussian	122.836	84.94	99.7	2.173	1.826	2.159
IMP	89.585	89.33	99.04	0.034	3.287	77.54
Gromacs	37.793	91.68	98.98	0.018	0.027	203.816
Namd	34.822	89.81	46.07	0.048	3.08	305.058
Fluent	32.009	77.12	85.66	0.04	1.673	16.377
OceanM	21.397	82.69	97.46	0.089	0.488	125.791
Qcprog	17.467	90.79	86.61	0.802	0.426	100.003
Relion	16.617	61.81	92.18	0.294	1.078	4.759
WRF	14.276	54.24	98.77	0.065	1.07	100.959
CCSM	9.639	80.64	66.26	0.126	0.046	71.654
AstroFoam	7.073	85.37	11.37	0.001	0.041	N/A
CESM	6.566	82.56	27.83	0.144	1.181	127.338

Source: one week sample (2015.6) by Paratera of ~100 private HPCs (~100 nodes each)

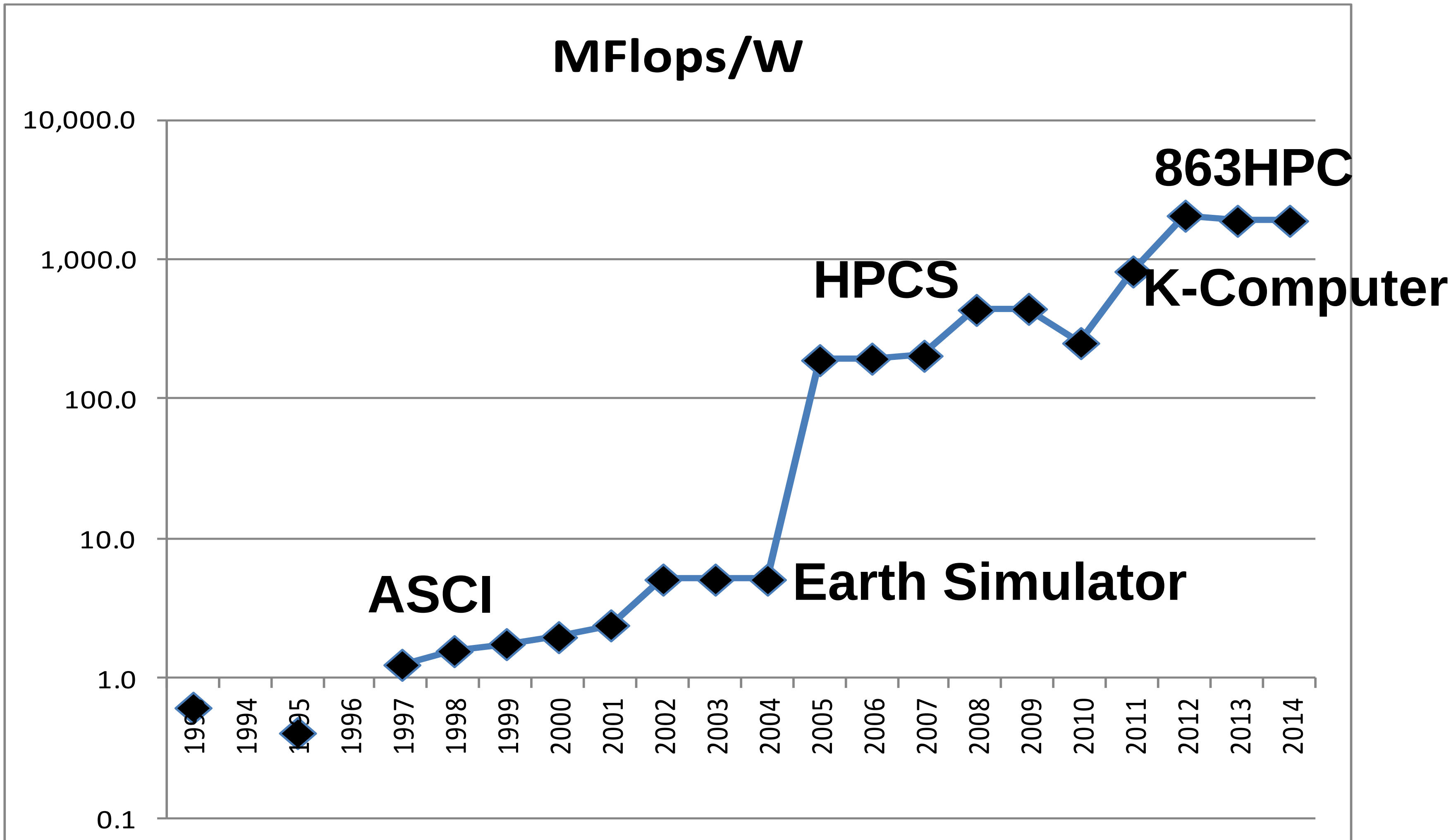
HPC's Three Phases

- Top priority went through two phases
  - Speed (flops), aka performance
  - Scalability: market scalability, problem scalability



An Important Efficiency Metric

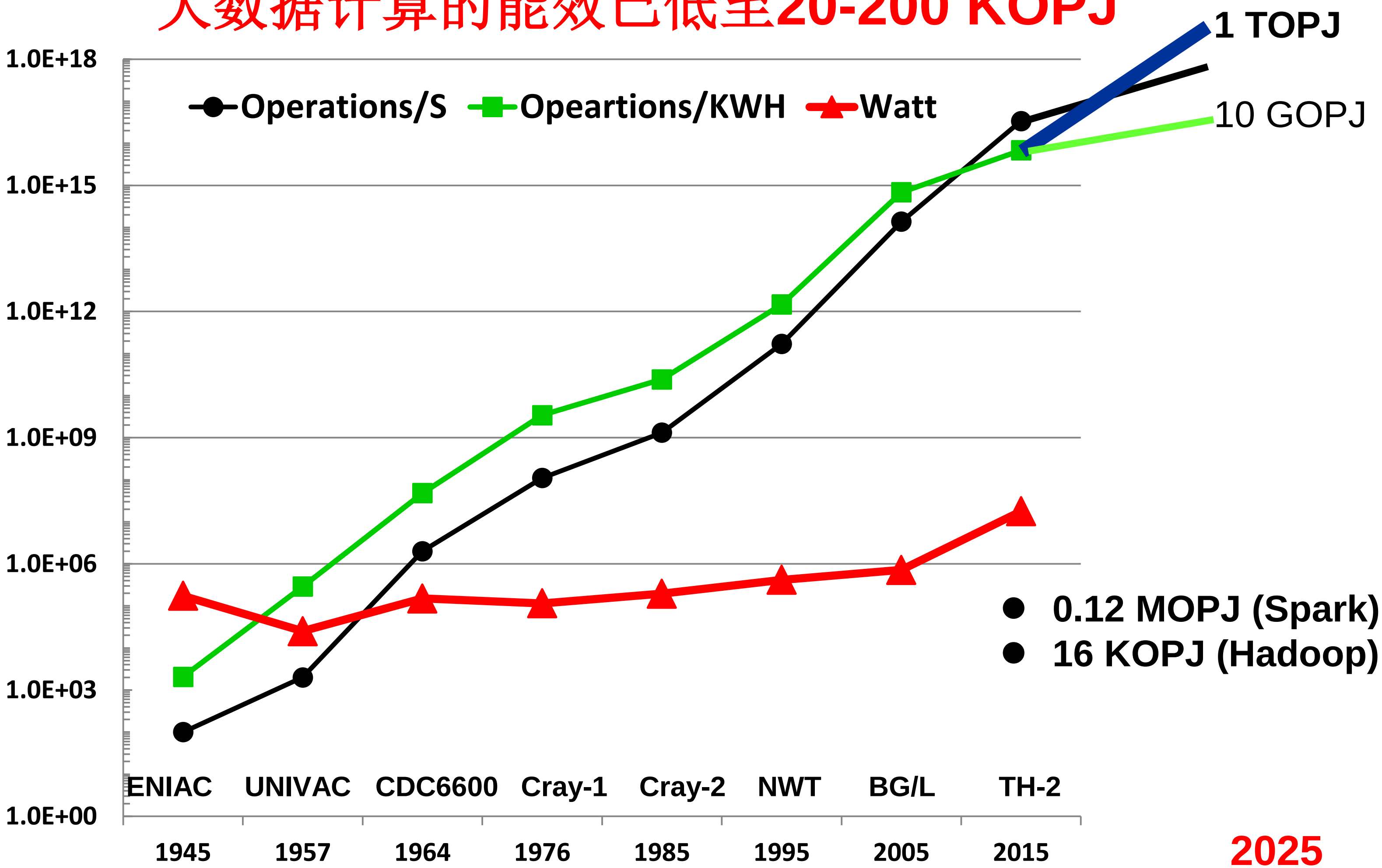
- Energy Efficiency: GOPS/W≈GOPJ



70年未有之大变局！

产业70年发展首次出现：**能效增长远低于速度增长**

大数据计算的能效已低至20-200 KOPJ



Need Both Progressive and Aggressive Approaches for 2035

- Progressive approaches
  - 50 GOPS/W by 2022
  - 1000 GOPS/W by 2035
- Aggressive approaches
  - 1000 GOPS/W by 2022
  - 1000 TOPS/W by 2035

- In all three areas of
  - Theory
  - Hardware
  - Software

	2010	2013	2022	2035
Top500	DoE Jaguar 1.76 POPS@6.75MW **0.253 GOPS/W**	**1.9 GOPS/W**	1000 POPS@20MW **50 GOPS/W**	100 EOPS@20MW **5 TOPS/W**
Green500	Dawning Nebula 1.27 POPS@2.58MW 0.492 GOPS/W	**3.13 GOPS/W**	250 GOPS/W？	50 TOPS/W？

重要HPC用户的诉求

- CERN High-Luminosity LHC in 202
  - 计算速度 50-100X
  - 存储容量 exabytes
  - 传统HPC不够，探索新方法
    - 商业云计算，新型计算机体系结构，高阶数据分析技术，深度学习

Dr. Maria Girone
CERN Openlab CTO

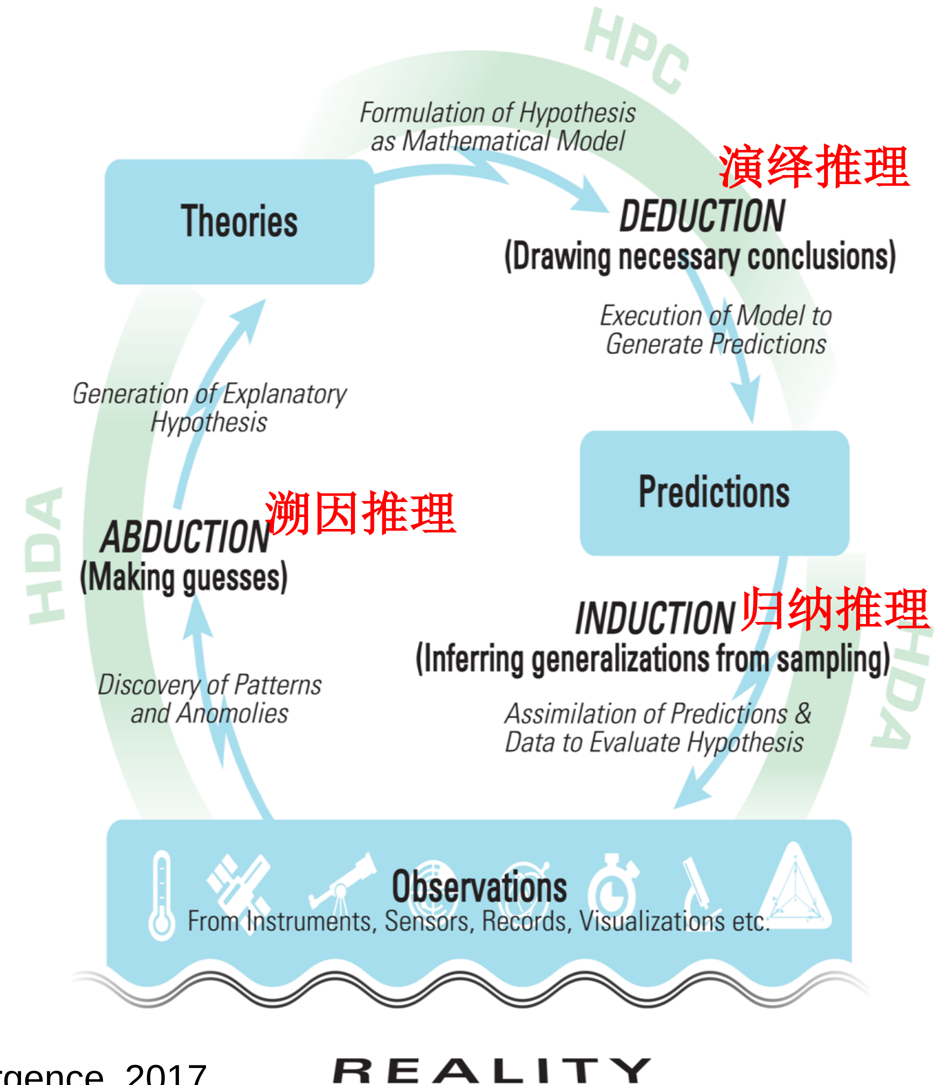
- 阿岗实验室：
  - 5年之内AI会长出另一个HPC市场，阿岗已启动200个AI项目
- Gatner: AI will be among the top 3 workloads in 2022

Argonne National Laboratory, Associate Director for Computing, Environment and Life Sciences

融合智能
演绎推理+归纳推理+溯因推理

科学研究界的科研活动 (process of scientific inquiry) 涉及三类推理 (inference, reasoning)

- Deduction 演绎推理
- Induction 归纳推理
- Abduction 溯因推理



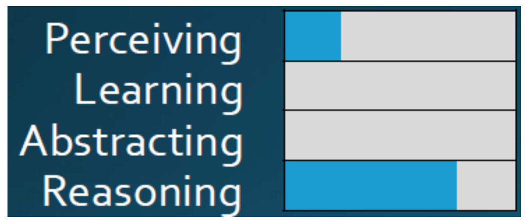
ECBD Pathway to Convergence, 2017

DARPA的人工智能愿景

J. Launchbury, I2O Director, 2017

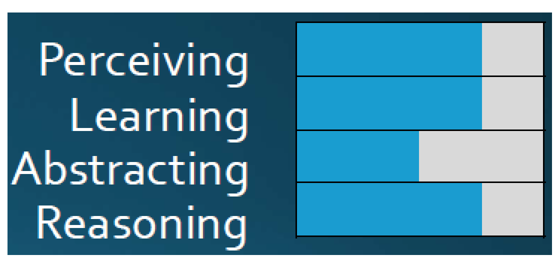
- 人工智能即将进入第三次浪潮

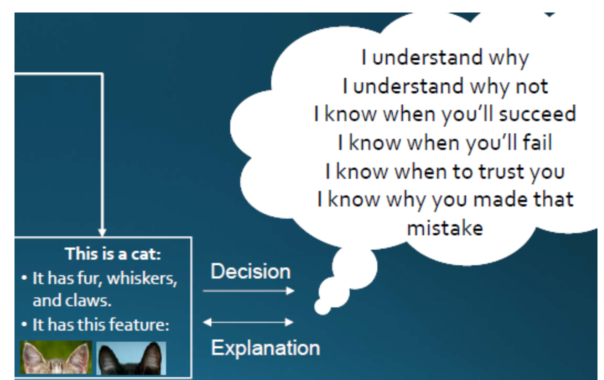
- The 1st Wave
  - Handcrafted Knowledge



- The 2nd Wave
  - Statistical Learning

- The 3rd Wave
  - Contextual Explanation





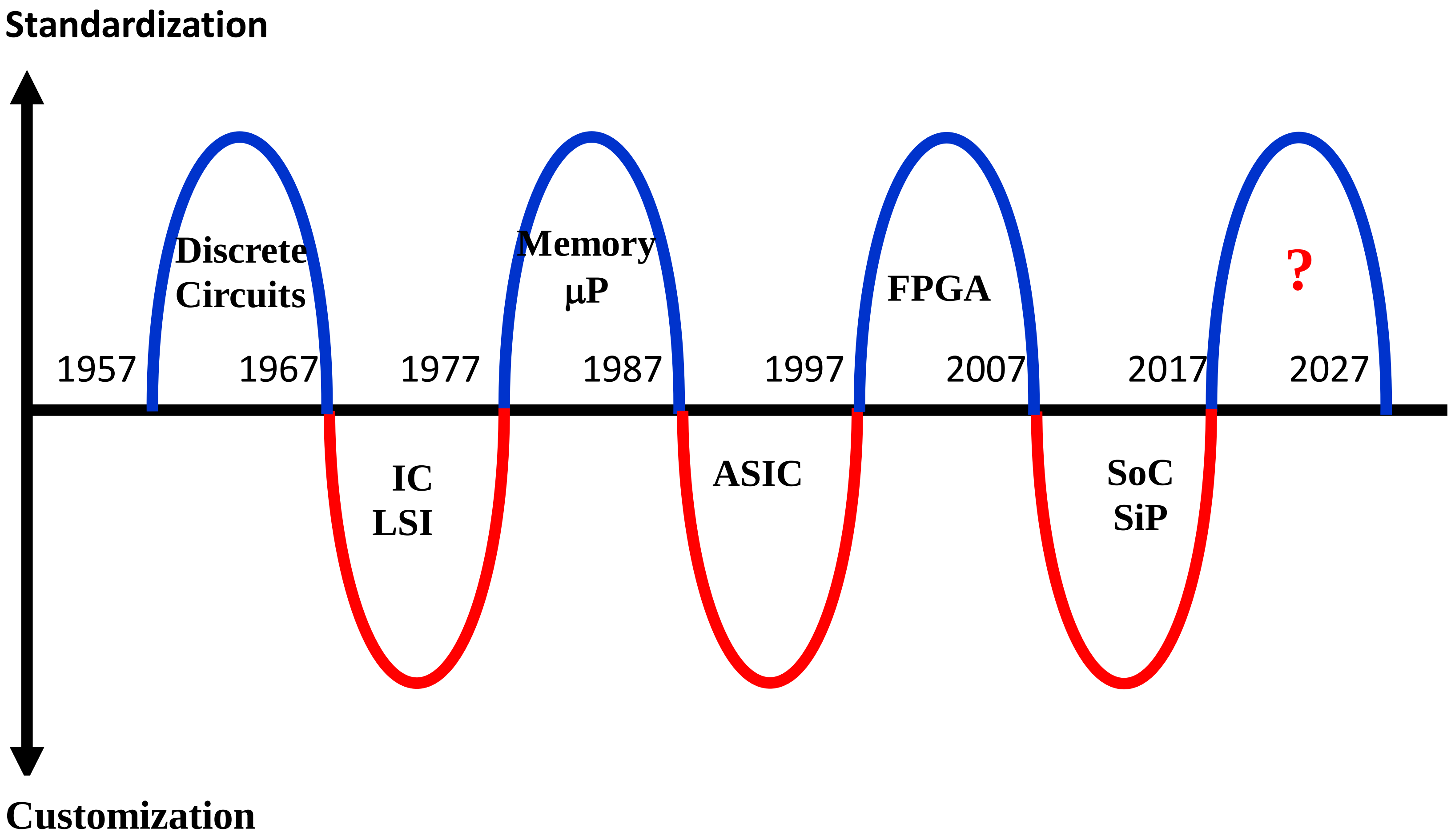
智能超算的内涵

- 传统超算仍有强大生命力
  - **智能超算拓展传统超算的能力与市场**
    - 智能超算=传统超算+新超算
    - 100亿美元 → 300亿美元
- 新在何处？
  - 新的典型应用与基准程序（不只是Linpack）
  - 新的使用模式与新的用户社区（30万用户）
  - 新的应用编程框架
  - 新的系统软件
  - 新的体系结构
  - **新的志愿者社区！**

大数据计算、智能计算框架的易用性
+
传统超算的高效率

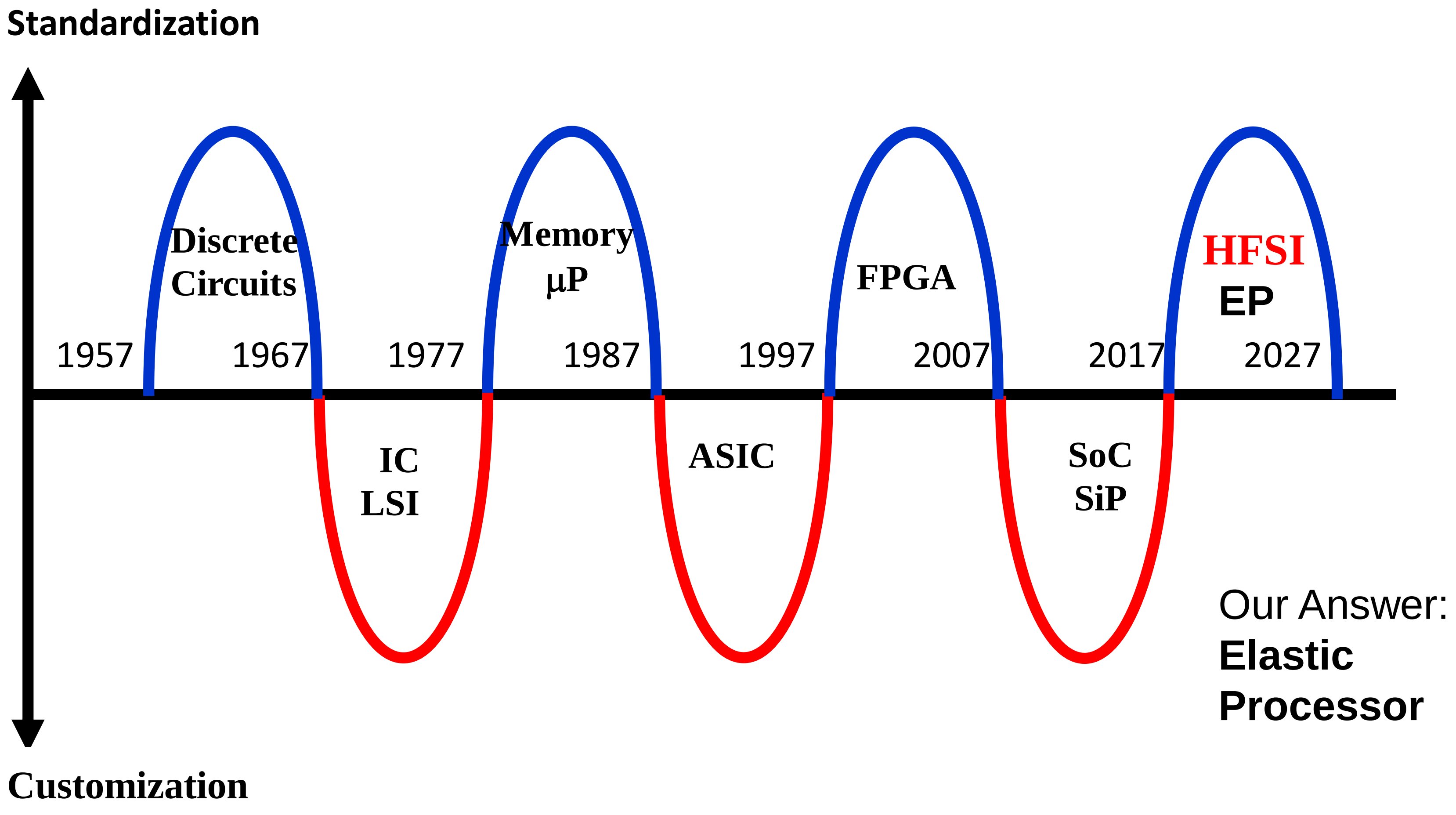
Makimoto's Wave

- Semiconductor technology will soon enter another phase change. But what is it?



Makimoto's Wave

- HFSI: Highly Flexible Super Integration
- Redundant circuits can be shut off when not in use

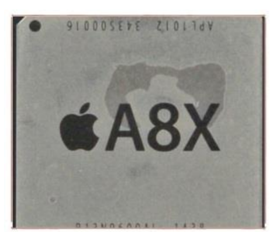


异构计算：可重塑处理器

- 设计目标：性能功耗比达到 **1000 GOPS/W = 1 TOPJ**
- 体系结构特色：
  - **函数指令集体系结构FISC**，一条指令可完成一个函数（如机器学习）
  - **可重塑ASIC**，一个定制电路可在一个应用领域内动态重塑，实现多个加速函数
  - 多独立环片上网络互连m个RISC-V通用核以及n个**可重塑ASIC加速核**
  - 可重塑加速核调用延迟仅4拍，不到FPGA百分之一

FISC=Function Instruction Set Computer

复杂指令集	精简指令集	函数指令集
CISC Intel X86	**RISC** ARM	**FISC**
数十种芯片 10W+ 1芯片 对 千万应用	数百种芯片 1W+ 1芯片 对 百万应用	一个体系结构，数千种芯片 0.1W+ 1芯片 对 千应用

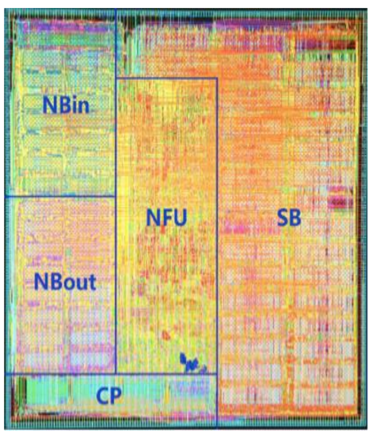


寒武纪神经网络加速器

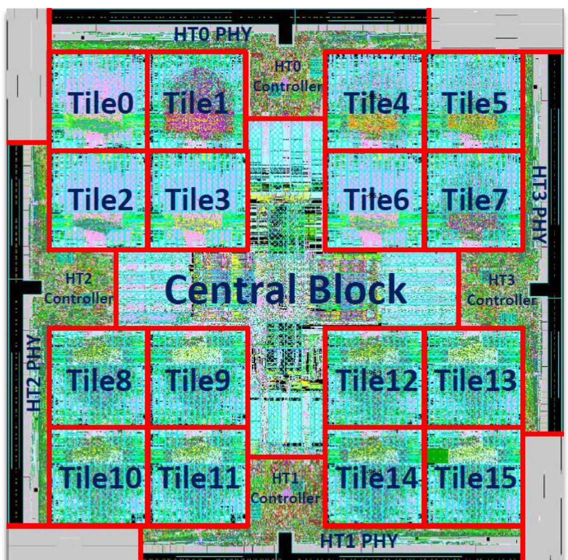
可重塑处理器有希望达到1 TOPJ

- 大电脑：0.6GHz, 5.58 TOPS, 68mm^2, 16W@28nm
  - **相对Nvidia K20 GPU：21x性能提升，330x能耗降低**

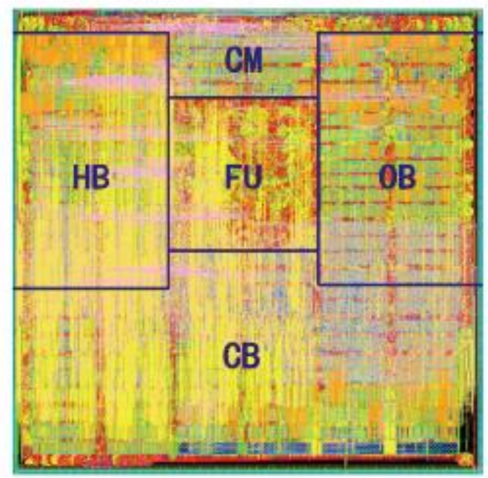
Yunji Chen, Tianshi Chen, Zhiwei Xu, Ninghui Sun, Olivier Temam. DianNao family: energy-efficient hardware accelerators for machine learning. *Communications of the ACM* 59(11): 105-112 (2016) Research Highlight paper



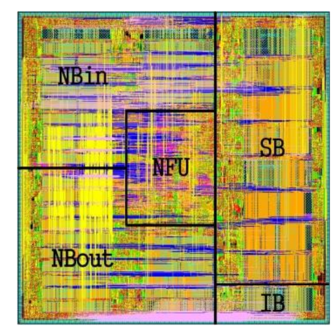
“电脑”
神经网络加速器
931 GOPS/W
ASPLOS’14最佳论文



“大电脑”
64芯片的神经网络超级计算机
100-250 GOPS/W
MICRO’14最佳论文

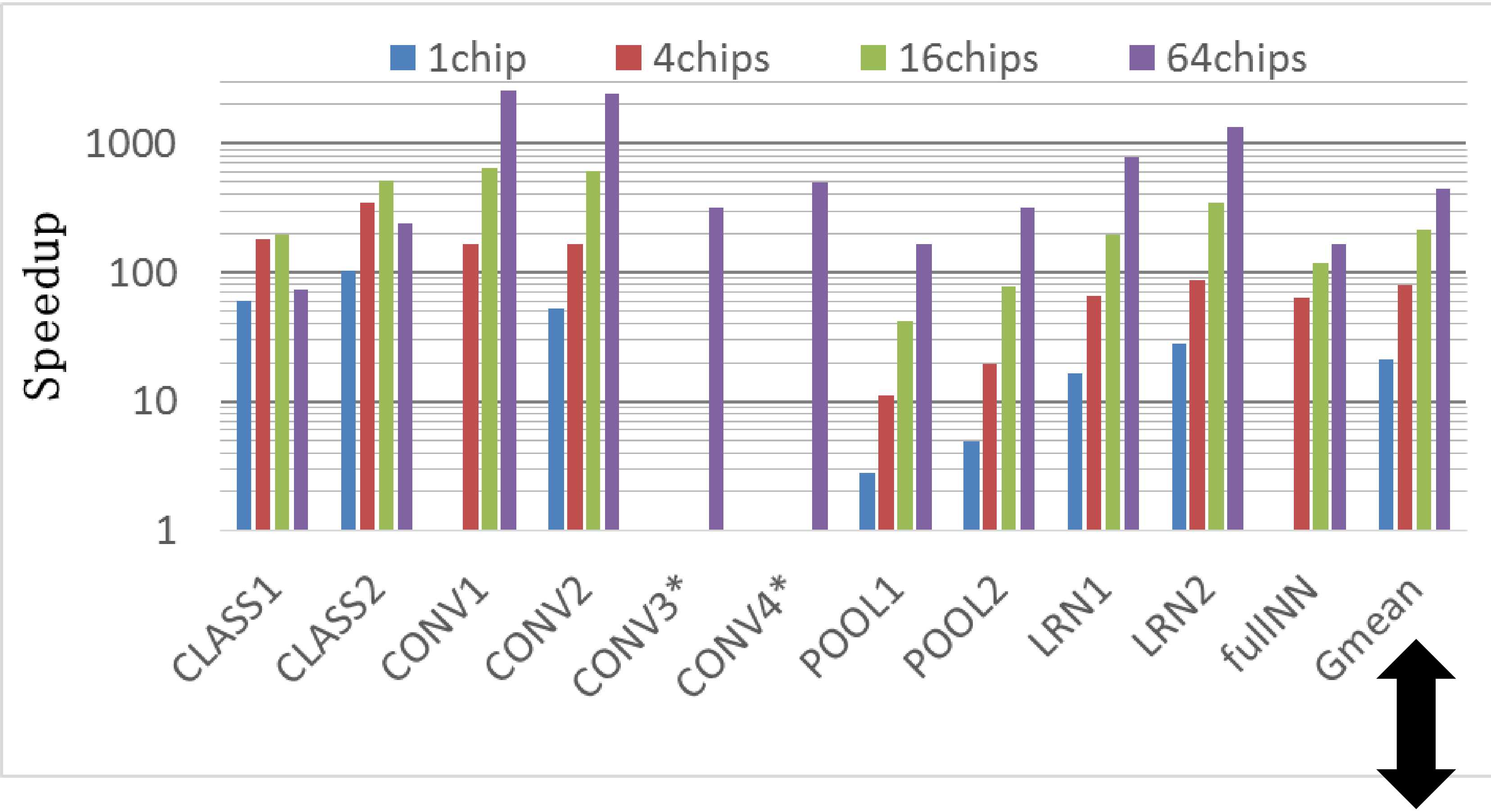


“普电脑”
通用机器学习加速器
300-1200 GOPS/W
ASPLOS’15



“视电脑”
海端视频处理
2-4 TOPS/W
ISCA’15

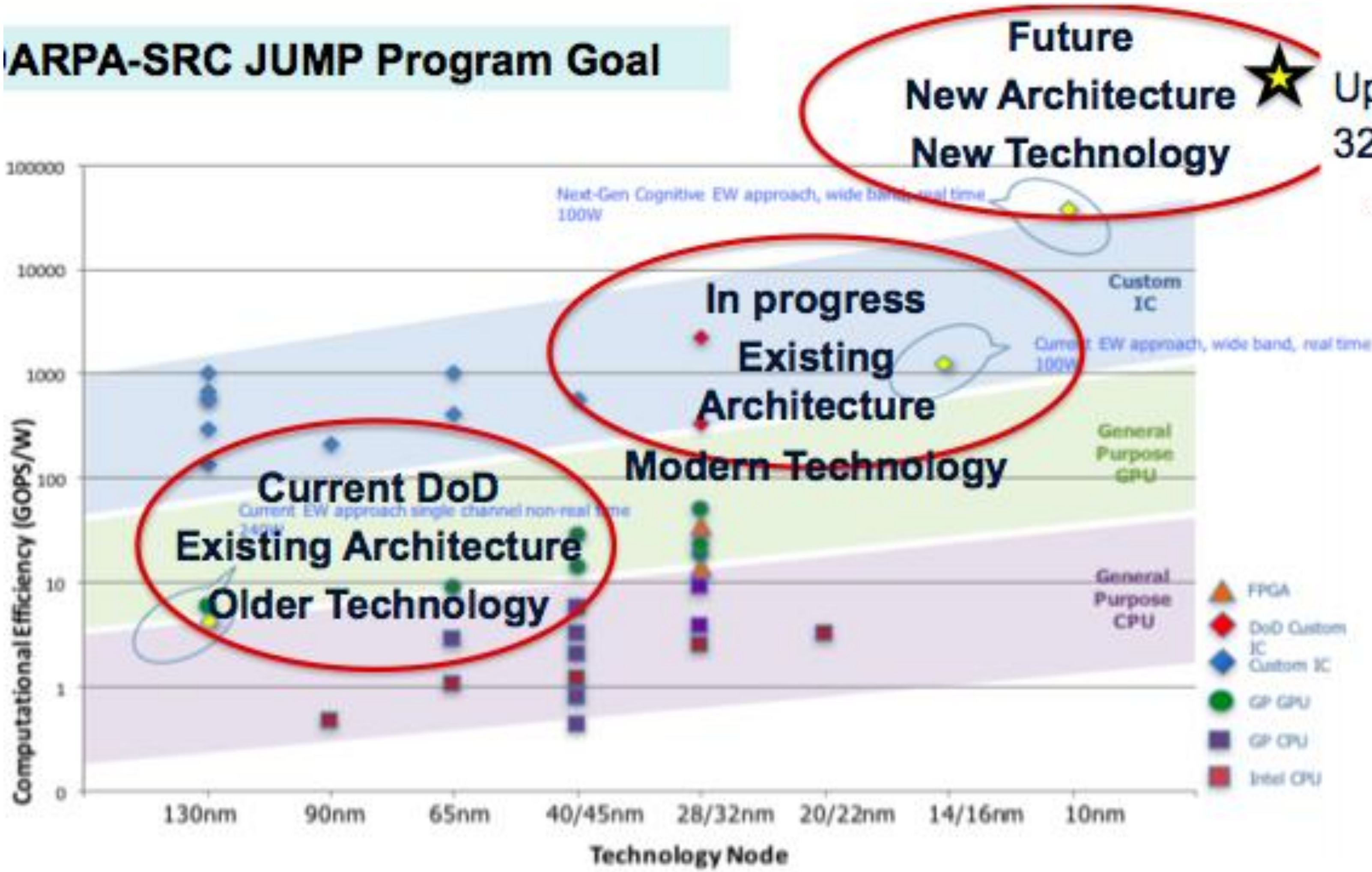
DaDianNao: An NN Supercomputer



- In average, 450x speedup and 150x energy saving over K20 GPU

MICRO 2014 Best Paper

POPS/W目标实例



Upside goal
3200 TOPS/W
~0.3 fJ/op

Linton Salmon,
JUMP Proposer
Day Workshop,
Jan 2017.

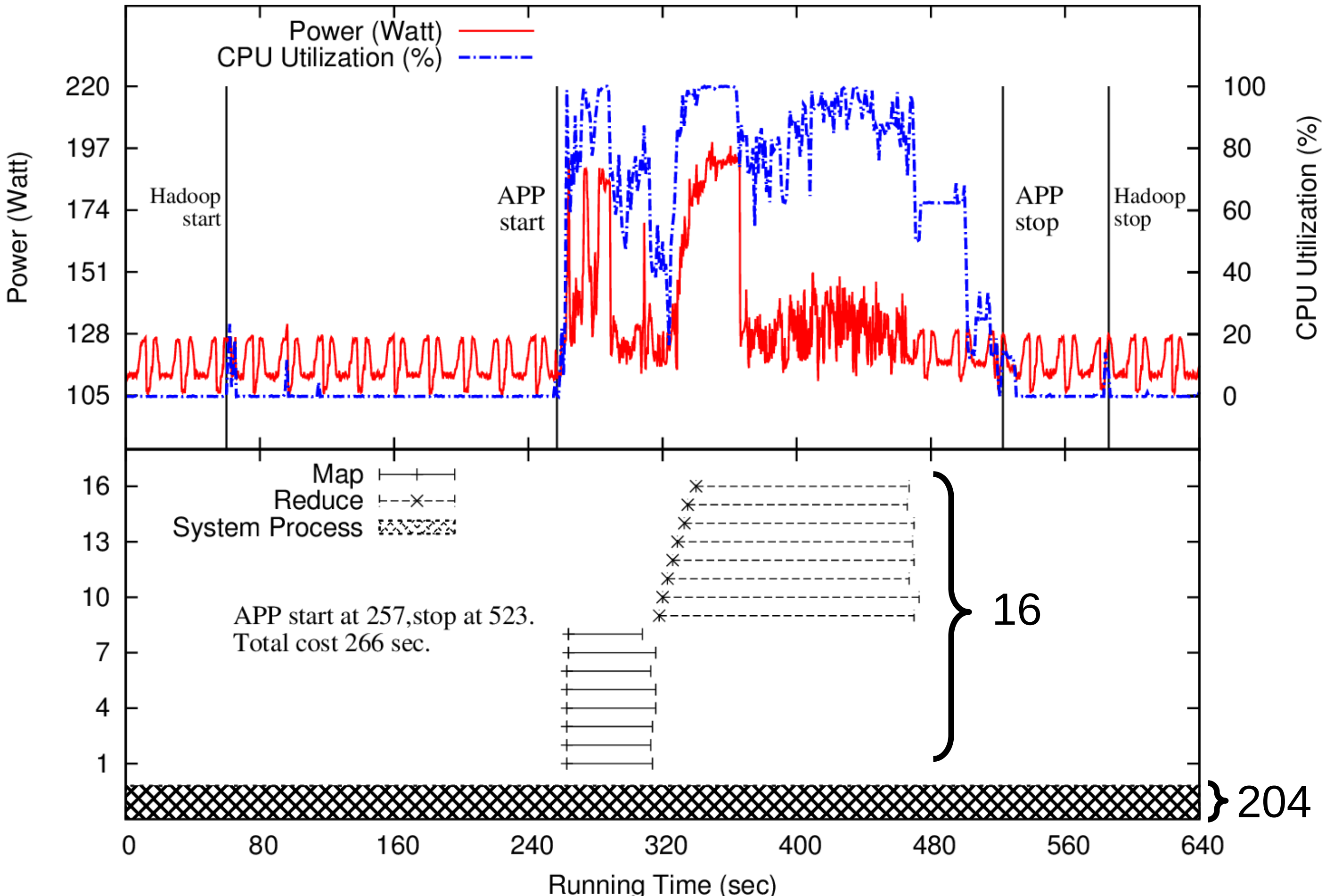
References

- Elastic Processor
  - Yunji Chen, Tianshi Chen, Zhiwei Xu, Ninghui Sun, and Olivier Temam: DianNao Family: Energy-efficient Hardware Accelerators for Machine Learning. to appear in Communications of the ACM (Technical Highlight paper).
  - Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Tao Luo, Xiaobing Feng, Yunji Chen, and Olivier Temam, "ShiDianNao: Shifting Vision Processing Closer to the Sensor", ISCA'15.
  - Daofu Liu, Tianshi Chen, Shaoli Liu, Jinhong Zhou, Shengyuan Zhou, Olivier Temam, Xiaobing Feng, Xuehai Zhou, and Yunji Chen, "PuDianNao: A Polyvalent Machine Learning Accelerator", ASPLOS'15.
  - Yunji Chen, Tao Luo, Shaoli Liu, Shijin Zhang, Liqiang He, Jia Wang, Ling Li, Tianshi Chen, Zhiwei Xu, Ninghui Sun, and Olivier Temam, "DaDianNao: A Machine-Learning Supercomputer", MICRO'14.
  - Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam, "DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning", ASPLOS'14.
- DataMPI
  - Lu Chao, Chundian Li, Fan Liang, Xiaoyi Lu, Zhiwei Xu: Accelerating Apache Hive with MPI for Data Warehouse Systems. ICDCS 2015: 664-673
  - Xiaoyi Lu, Fan Liang, Bing Wang, Li Zha, Zhiwei Xu: DataMPI: Extending MPI to Hadoop-Like Big Data Computing. IPDPS 2014: 829-838
  - Xiaoyi Lu, Bing Wang, Li Zha, Zhiwei Xu: Can MPI Benefit Hadoop and MapReduce Applications? ICPP Workshops 2011: 371-379
- Energy Efficient Ternary Computing
  - Zhiwei Xu, Xuebin Chi, Nong Xiao: High-Performance Computing Environment: A Review of Twenty Years Experiments in China. National Science Review, to appear
  - Zhiwei Xu: High-Performance Techniques for Big Data Computing in Internet Services. Invited speech at SC12, SC Companion 2012: 1861-1895
  - Zhiwei Xu: Measuring Green IT in Society. IEEE Computer 45(5): 83-85 (2012)
  - Zhiwei Xu: How much power is needed for a billion-thread high-throughput server? Frontiers of Computer Science 6(4): 339-346 (2012)
  - Zhiwei Xu, Guojie Li: Computing for the masses. Commun. ACM 54(10): 129-137 (2011)
- **http://novel.ict.ac.cn/zxu/#PAPERS**

zxu@ict.ac.cn   **http://novel.ict.ac.cn/zxu/**

Hadoop Efficiency Is Low

- Lacks a high-performance communication substrate
  - Use HTTP, RPC, direct Sockets over TCP/IP to communicate
  - Can MPI be used for big data?



Speed Efficiency

(Sustained/Peak)

Payload:	**0.002%**
Linpack:	**94.5%**
Total op:	4.22%
Instruction:	4.72%

Energy Efficiency

(Operations per Joule)

Payload:	**1.55×10^4**
Linpack:	**7.26×10^8**
Total op:	2.20×10^7
Instruction:	2.45×10^7

Desired Sort Code via DataMPI: Scalable and Easy to Write

```
public class Sort {
    public static void main(String[] args) {
        try {
            int rank, size;
            Map<String, String> conf = new HashMap<String, String>();
            conf.put(MPI_D_Constants.KEY_TYPE, java.lang.String.class.getName());
            conf.put(MPI_D_Constants.VALUE_TYPE, java.lang.String.class.getName());
            MPI_D.Init(args, MPI_D.Mode.Common, conf);
            if (MPI_D.COMM_BIPARTITE_O != null) {
                rank = MPI_D.Comm_rank(MPI_D.COMM_BIPARTITE_O);
                size = MPI_D.Comm_size(MPI_D.COMM_BIPARTITE_O);
                String[] keys = loadKeys(rank, size);
                if (keys != null) {
                    for (int i = 0; i < keys.length; i++) {
                        MPI_D.Send(keys[i], "");
                    }
                }
            } else {
                rank = MPI_D.Comm_rank(MPI_D.COMM_BIPARTITE_A);
                size = MPI_D.Comm_size(MPI_D.COMM_BIPARTITE_A);
                Object[] keyValue = MPI_D.Recv();
                while (keyValue != null) {
                    System.out.println("Task " + rank + " of " + size + ", key is "
                        + ((String) keyValue[0]) + ", value is " + ((String) keyValue[1]));
                    keyValue = MPI_D.Recv();
                }
            }
            MPI_D.Finalize();
        } catch (MPI_D_Exception e) {
            e.printStackTrace();
        }
    }
}
```

init

rank/size

send

recv

finalize

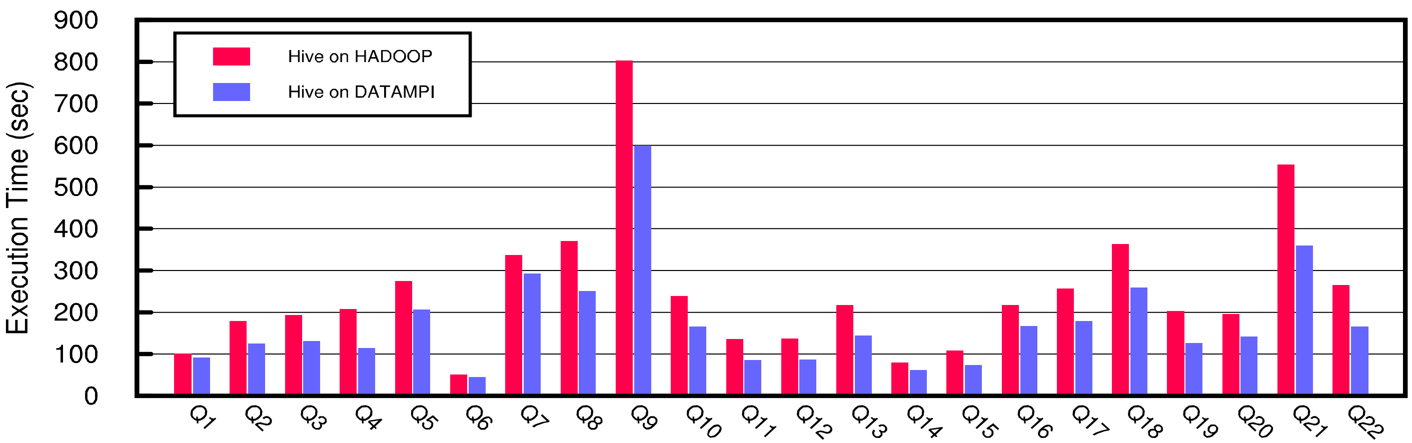
33 lines of code

1 GB, 1 TB, 1PB

Hive on DataMPI

A first attempt to propose a general design for fully supporting and accelerating data warehouse systems with MPI

- **Functionality & Productivity & Performance**
  - Support Intel **HiBench** (2 micro benchmark queries) & **TPC-H** (22 app queries)
  - Only **0.3K** LoC modified in Hive
  - HiBench: **30%** performance improvement on average
  - TPC-H: **32%** improvement on average, up to **53%**



Performance Benefits with 40 GB data for 22 TPC-H queries

Lu Chao, Chundian Li, Fan Liang, Xiaoyi Lu, Zhiwei Xu. *Accelerating Apache Hive with MPI for Data Warehouse Systems*. ICDCS 2015, Columbus, Ohio, USA, 2015