FLAGSHIP2020プロジェクトと エクサスケールシステムのプログラミングの課題

佐藤　三久　　Team Leader of Architecture Development Team

FLAGSHIP 2020 project
RIKEN Advance Institute of Computational Science (AICS)

自己紹介

- 昭和57年東京大学理学部情報科学科卒業。昭和61年同大学院理学系研究科博士課程中退。
- 同年新技術事業団後藤磁束量子情報プロジェクトに参加。
- 平成3年、通産省電子技術総合研究所入所。
- 平成8年、新情報処理開発機構並列分散システムパフォーマンス研究室 室長。
- 平成13年から平成27年まで、筑波大学 システム情報系 教授。平成19年度より平成24年度まで、同大学計算科学研究センターセンタ長。
- 平成22年より、理化学研究所計算科学研究機構プログラミング環境研究チームリーダ。
- 平成26年より、同機構エクサスケールコンピューティング開発プロジェクト副プロジェクトリーダ。
- 筑波大学連携大学院教授
- 並列処理アーキテクチャ、プログラミングモデルと言語およびコンパイラ、計算機性能評価技術等の研究に従事。情報処理学会、IEEE、日本応用数理学会会員。

Outline of Talk

- **「次世代スーパーコンピュータプロジェクト」**
- **FLAGSHIP 2020 プロジェクト**
  - An Overview of post K system

- **exascale computingの課題**
  - システム
  - プログラミング

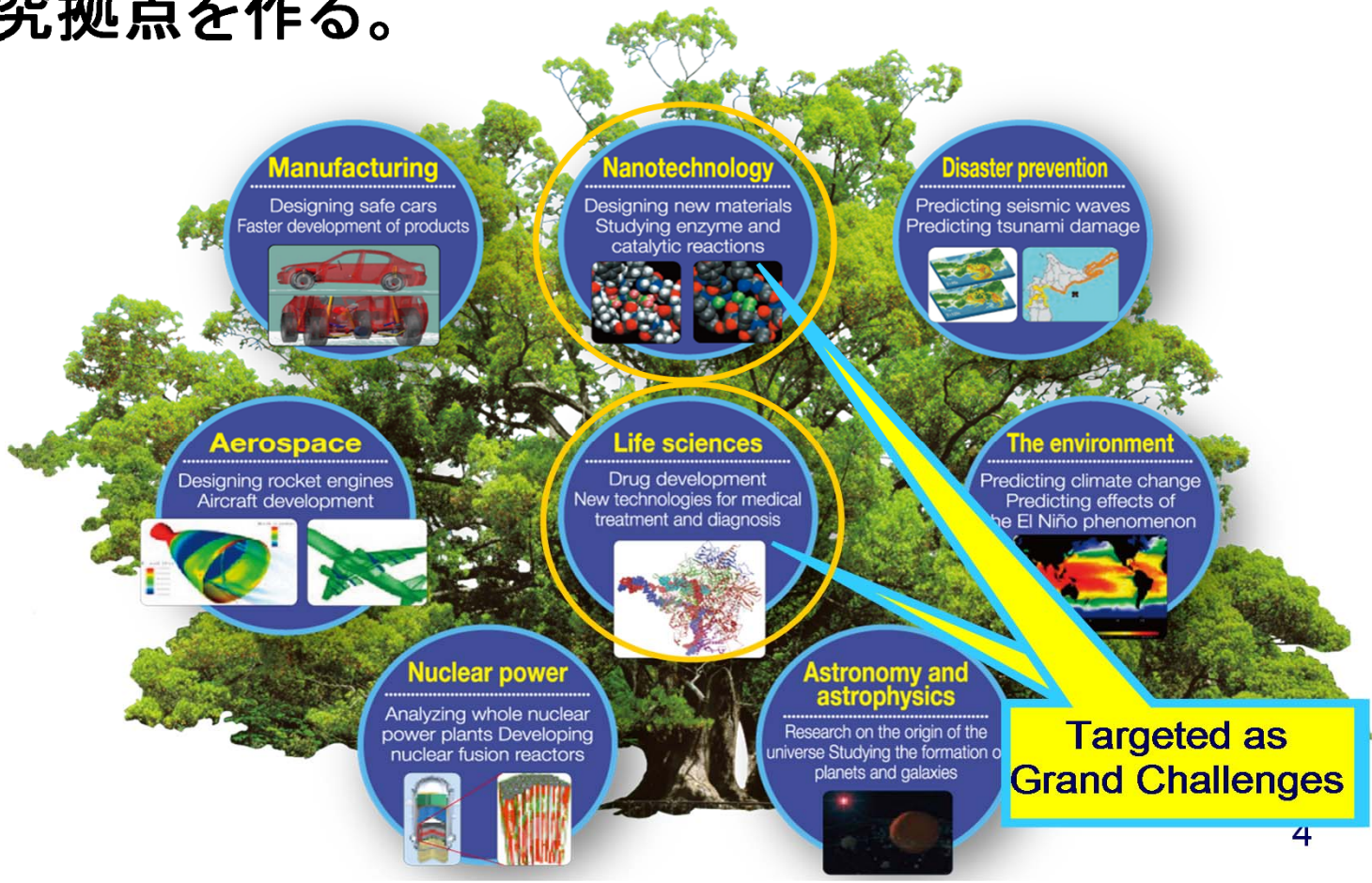
- **Concluding Remarks**

“次世代スーパーコンピュータプロジェクト”

- 世界最高速の性能を持つ汎用スーパーコンピュータの開発する。目標性能は、Linpackベンチマークで10ペタフロップス以上。
- 大規模計算科学アプリケーションを開発し、次世代スパコンで科学のブレークスルーを達成。
- 我が国の計算科学の研究拠点を作る。

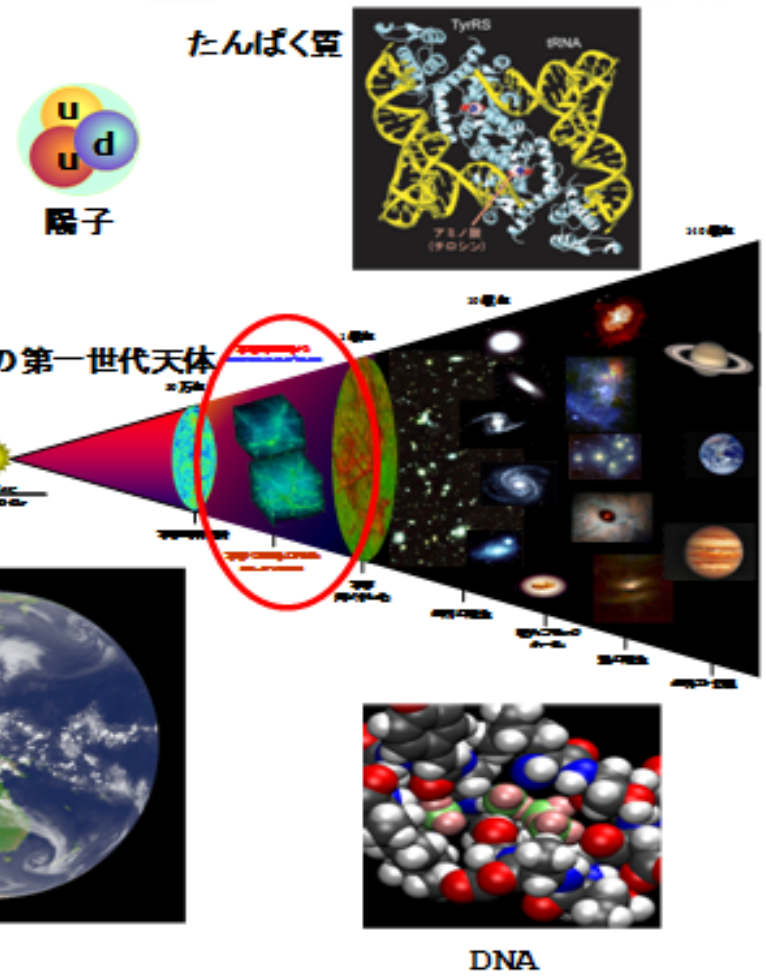
  ⇒　計算科学研究機構

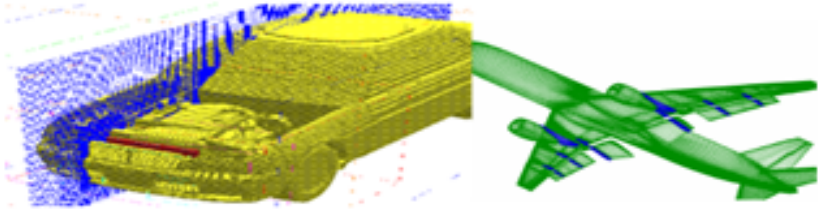
- 2006年度から2012年度
- 7年間のプロジェクト
- 予算総額1100億円



計算科学の重要性：何に役立つのか

- 「紙と鉛筆」では解けないような複雑な現象の探求
  - 物質の根源である素粒子の成り立ち
  - DNAやたんぱく質等数百万個の原子の集団の示す性質
- 実験ができない現象の探求
  - 宇宙における最初の天体の起源
  - 地球規模の気候変動と温暖化予測
- 膨大な大規模データの探索
  - ゲノムインフォマティクス
- 実験の代替や開発コストの低減
  - 自動車の衝突シミュレーション
  - 航空機設計





第一原理的手法を使用すれば，実験不可能なことでも，シミュレーションによって解明される，であろうことが明らかになりつつある。
•バイオ，ナノテクノロジー
•現在の計算機リソースでは不可能なものも多い

スパコンのハードウェアの歴史

- 1983年：1 GFLOPS，1996年：1 TFLOPS…

- 1990年以前は、特別なスパコン（ベクトル型）が主流
- 1990年代以降は、多数のコンピュータを結合した並列計算機が主流に。
- PCに使われているマイクロプロセッサ（1つのチップでできたコンピュータ）の急激な進歩
  - 1.5年に2倍の割合でトランジスタの集積度が増加（ムーアーの法則）
  - 4004(世界初、1971年、750KHz) 8008(1972年、500KHz、インテル) 8080(1974年、2MHz、インテル)
  - Pentium 4 (2000年、～3.2GHz)

- 30年間で、1MHzから1GHz、1000倍の進歩

スパコンのハードウェアの歴史

- 2000年以降は、PCに使われてマイクロプロセッサを使ったが並列計算機（PCクラスタ）が主流に。
- 2008年にはIBM RoadRunner, 1Peta Flops を達成
- そして、「京」が世界1に！

TOP 500 List スパコン・ランキング
http://www.top500.org/

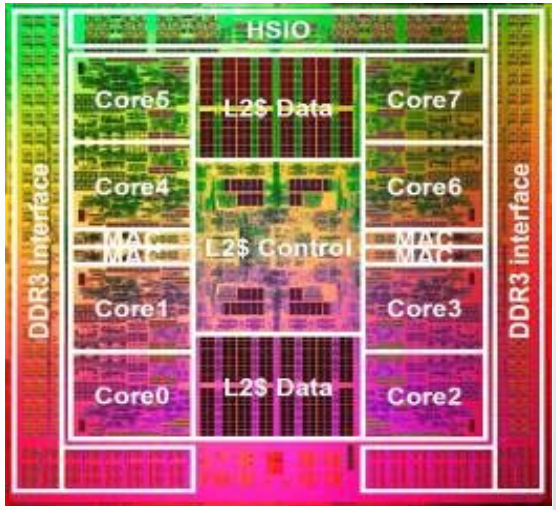
- LINPACKと言われるベンチマークプログラムの性能を性能の基準とする。
  - 超大規模な連立一次方程式を解く
  - 1千万次元の連立1次方程式
- 実際のアプリケーションの性能とは違う
  - 実際のアプリケーションではこれほどの性能は出ない
- 2008年から、電力消費量を表示するようになった
  - これからのスパコンは、電力が大切

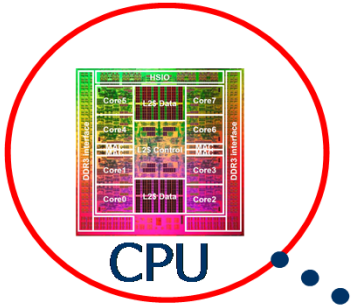
June/2011

Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect / 2011 Fujitsu	548352	8162.00	8773.63	9898.56
2	National Supercomputing Center in Tianjin China	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C / 2010 NUDT	186368	2566.00	4701.00	4040.00
3	DOE/SC/Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz / 2009 Cray Inc.	224162	1759.00	2331.00	6950.60
4	National Supercomputing Centre in Shenzhen (NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU / 2010 Dawning	120640	1271.00	2984.30	2580.00
5	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU Linux/Windows / 20 NEC/HP				
6	DOE/NNSA/LANL/SNL United States	Cielo - Cray XE6 8- Cray Inc.				
7	NASA/Ames Research Center/NAS United States	Pleiades - SGI Altix Xeon HT QC 3.0/Xe Infiniband / 2011 SGI				
8	DOE/SC/LBNL/NERSC United States	Hopper - Cray XE6 Cray Inc.				
9	Commissariat a l'Energie Atomique (CEA) France	Tera-100 - Bull bull S6010/S6030 / 201 Bull SA				
10	DOE/NNSA/LANL United States	Roadrunner - Blade Cluster, PowerXCel 1.8 GHz, Voltaire In IBM				

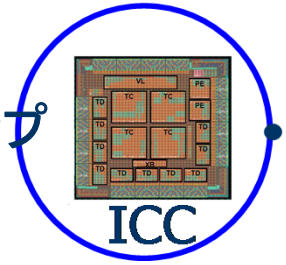
京コンピュータの構成

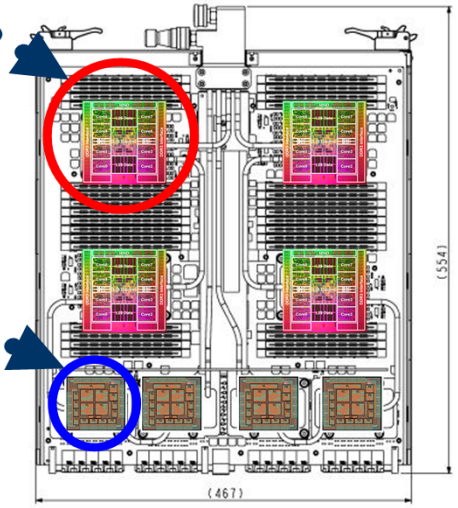
- 1つのチップに8個のコンピュータ(コア)
- 1つのコンピュータの性能は、16GFLOPS (2GHz), チップあたり、128GFLOPS
- PCとかわらない？





通信チップ





システムボード

Courtesy of FUJITSU Ltd.

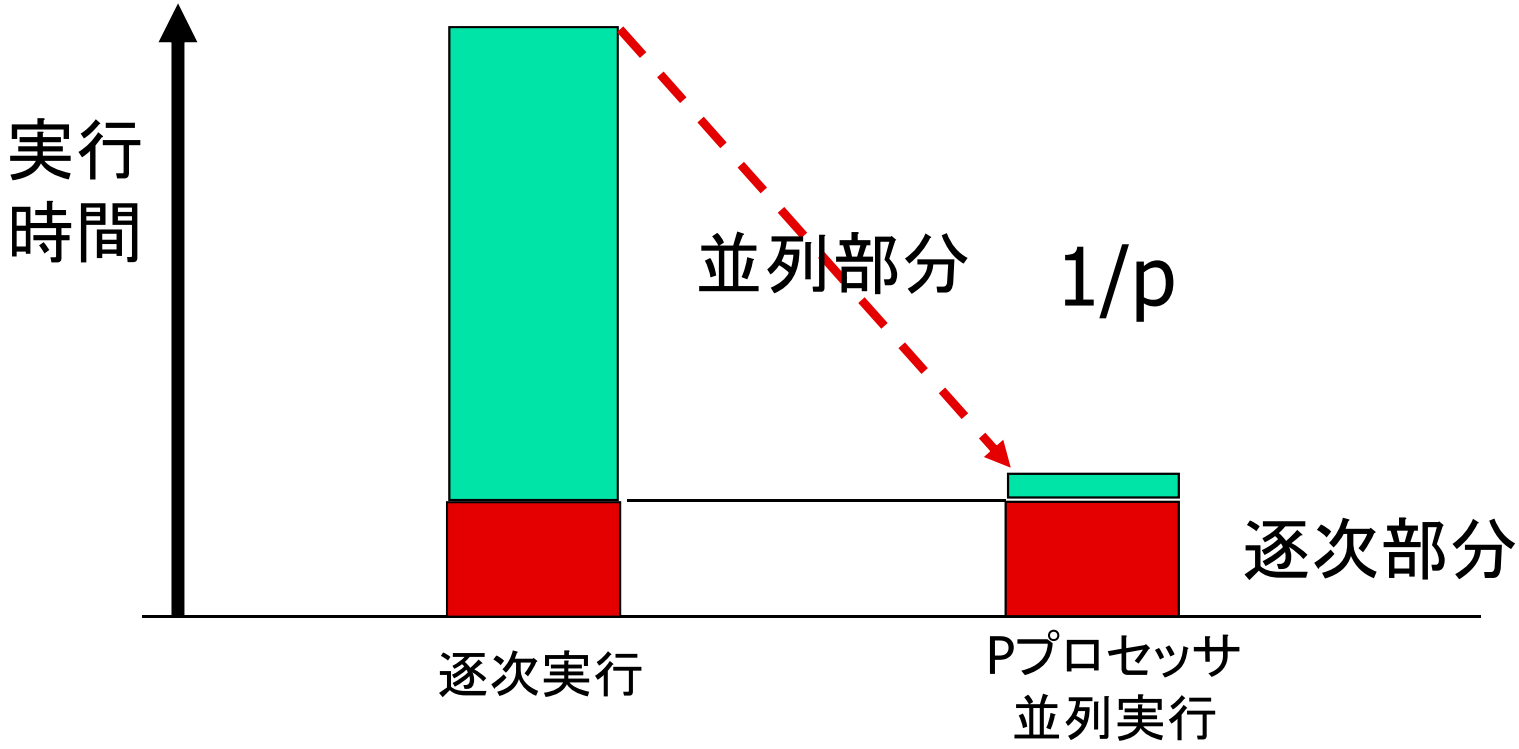
京コンピュータ　全体のデータ

- 筐体数　864
- チップ数:　82,944
- コンピュータ数:　663,552
- 性能　Linpack
  10.51PF
  (電力12.66MW)
  2011/11月

平成24年10月1日現在
つくば市人口:**217,315**人
男:**111,288**人
女:**106,027**人
世帯:**90,151**世帯

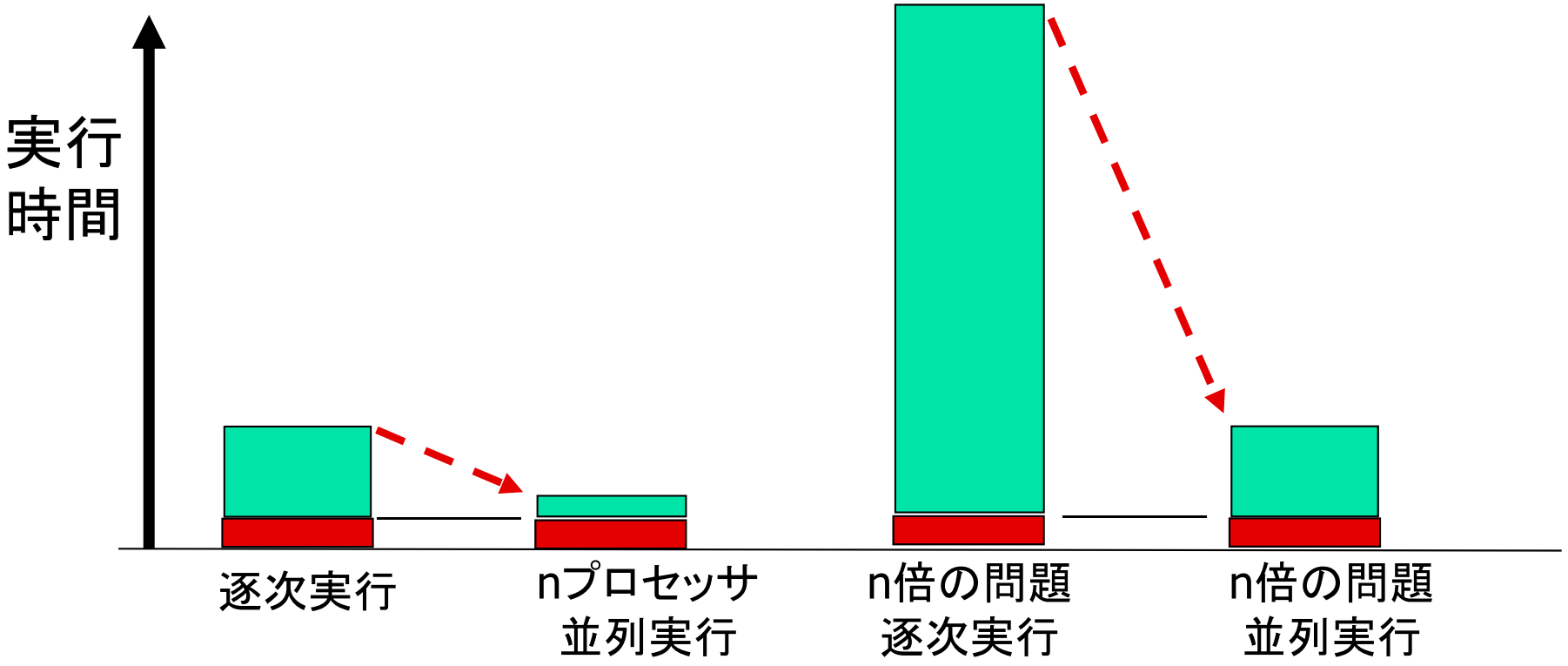
並列処理の問題点:「アムダールの法則」の呪縛

- アムダールの法則
  - 逐次処理での実行時間をT_1, 逐次で実行しなくてはならない部分の比率がαである場合、pプロセッサを用いて実行した時の実行時間(の下限)T_pは、$T_p = \alpha * T_1 + (1-\alpha) * T_1/p$
  - つまり、逐次で実行しなくてはならない部分が10%でもあると、何万プロセッサを使っても、高々10倍にしかならない。

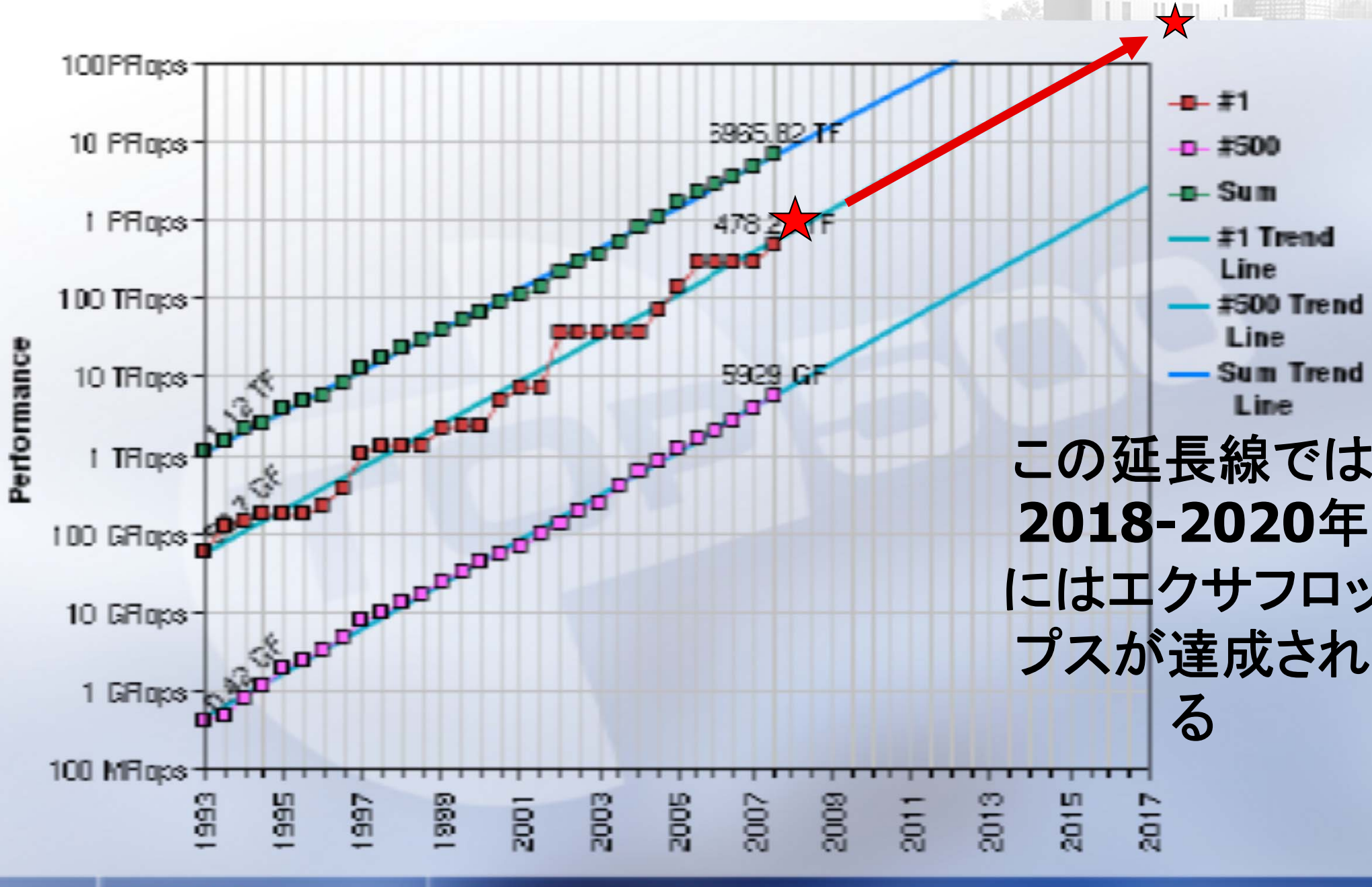


並列処理の問題点:「アムダールの法則」の呪縛

- 「Gustafsonの法則」:では実際のアプリではどうか？
  - 並列部分は問題規模によることが多い
  - 例えば、ノード数nの場合、n倍の大きい問題を解けばよい。n倍の問題は、計算量がnになると、並列処理部分は一定
  - Weak scaling – プロセッサあたりの問題を固定←大規模化は可能
  - Strong scaling ―問題サイズを固定　←こちらはプロセッサが早くなくてはならない。



これからのトレンド



この延長線では**2018-2020年**にはエクサフロップスが達成される

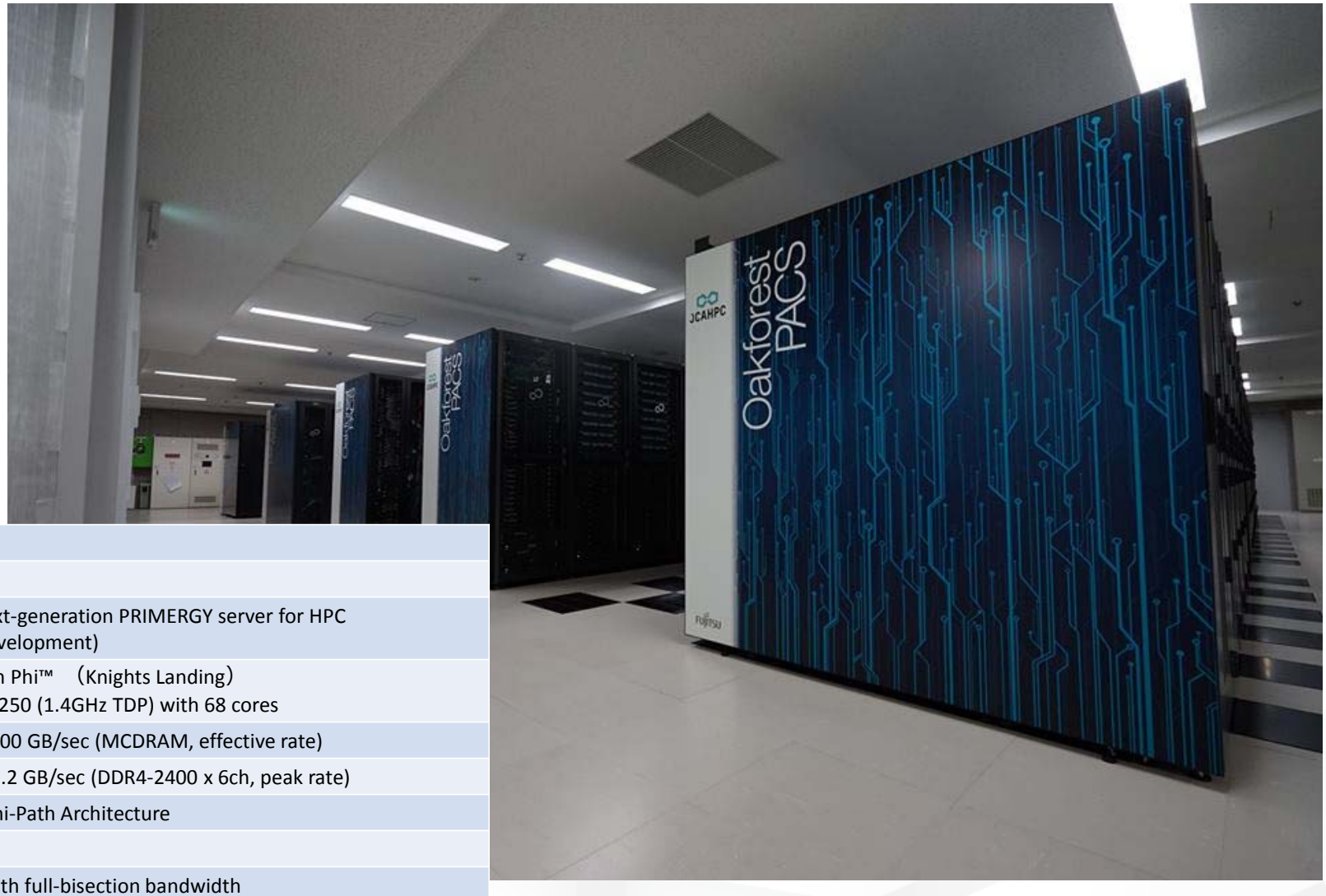
いま、最先端のスパコンを作る時の問題は、…

- いまのスパコンの性能は、並列処理から
- すなわち、コンピュータ数
- ということは、性能は結合するコンピュータの数を増やせばいい
- が、電力が限界

Oakforest-PACS

JCAHPC

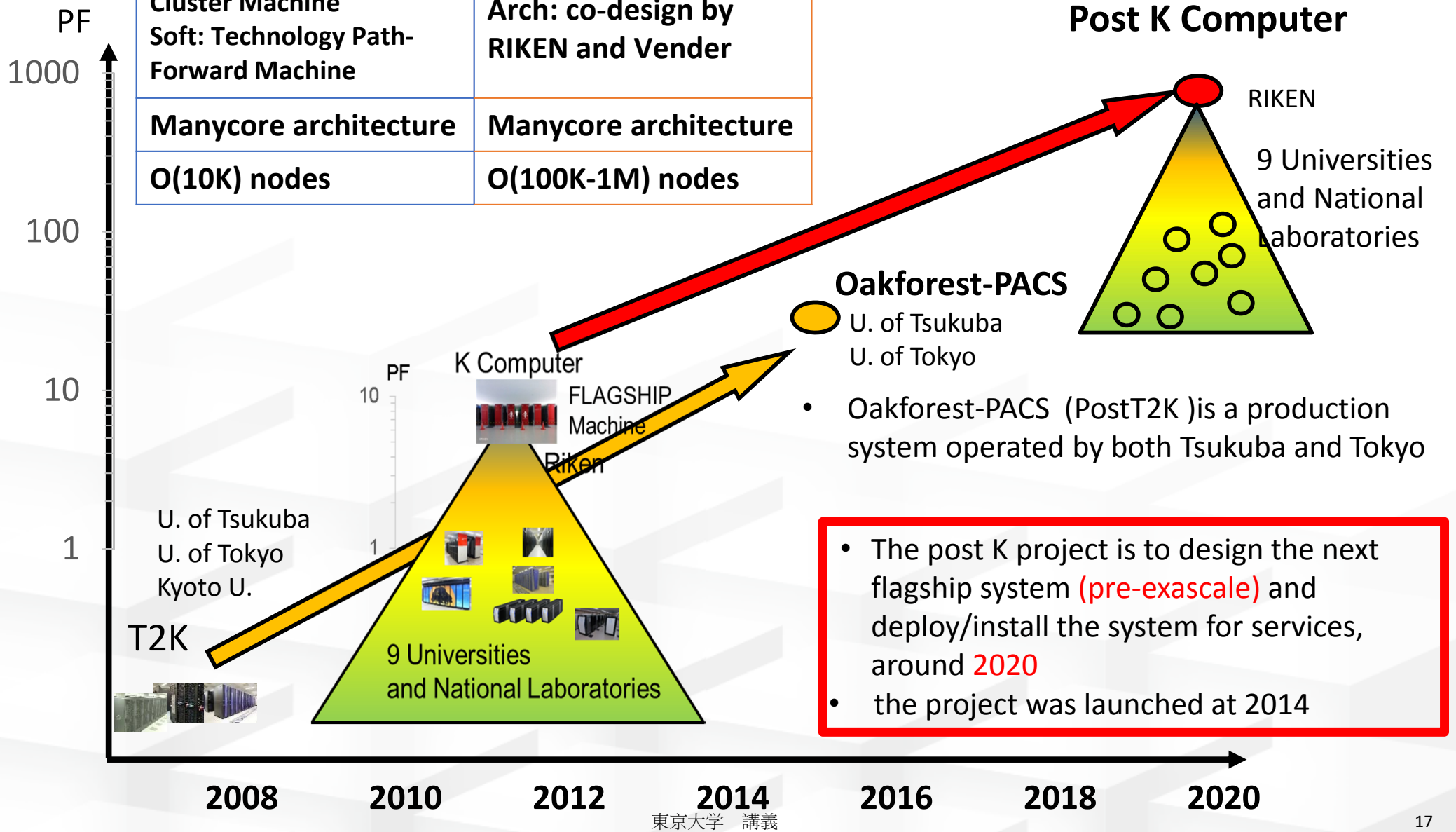
Japan's fastest supercomputer in Top500 of Nov. 2016 (13.55PF)



Total peak performance			25 PFLOPS
Total number of compute nodes			8,208
Compute node	Product		Fujitsu Next-generation PRIMERGY server for HPC (under development)
	Processor		Intel® Xeon Phi™ （Knights Landing） Xeon Phi 7250 (1.4GHz TDP) with 68 cores
	Memory	High BW	16 GB, > 400 GB/sec (MCDRAM, effective rate)
		Low BW	96 GB, 115.2 GB/sec (DDR4-2400 x 6ch, peak rate)
Inter-connect	Product		Intel® Omni-Path Architecture
	Link speed		100 Gbps
	Topology		Fat-tree with full-bisection bandwidth
Login node	Product		Fujitsu PRIMERGY RX2530 M2 server
	# of servers		20
	Processor		Intel Xeon E5-2690v4 (2.6 GHz 14 core x 2 socket)
	Memory		256 GB, 153 GB/sec (DDR4-2400 x 4ch x 2 socket)

Towards the Next Flagship Machine

PostT2K	PostK
Arch: Upscale Commodity Cluster Machine Soft: Technology Path-Forward Machine	Flagship Machine Arch: co-design by RIKEN and Vender
Manycore architecture	Manycore architecture
O(10K) nodes	O(100K-1M) nodes



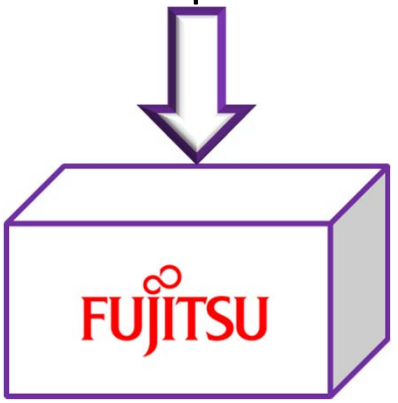
- Oakforest-PACS (PostT2K)is a production system operated by both Tsukuba and Tokyo

- The post K project is to design the next flagship system (pre-exascale) and deploy/install the system for services, around 2020
- the project was launched at 2014

An Overview of Flagship 2020 project

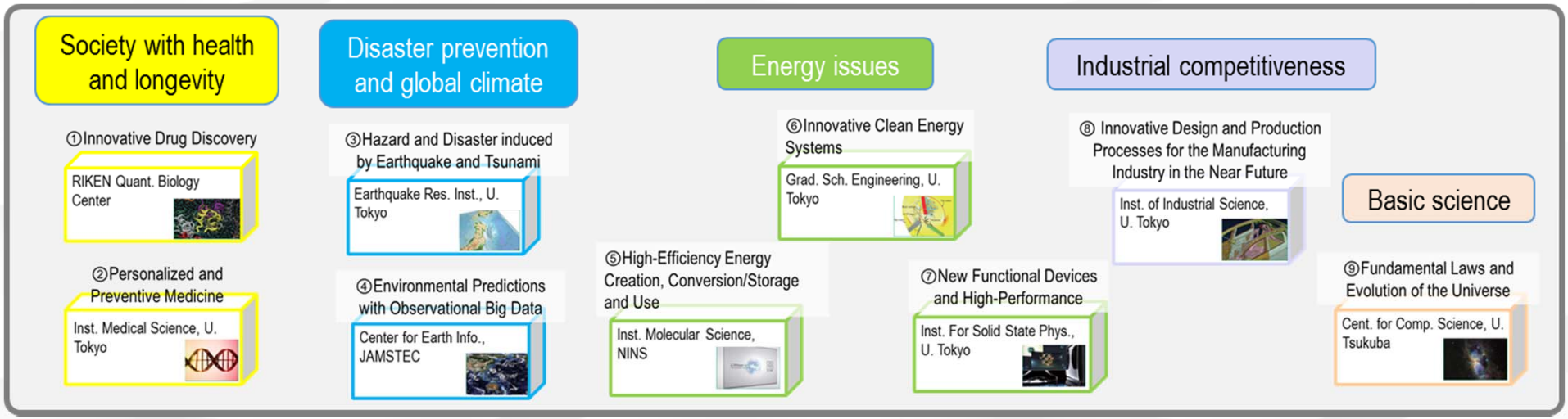
- **Developing the next Japanese flagship computer, temporarily called "post K"**

Vendor partner

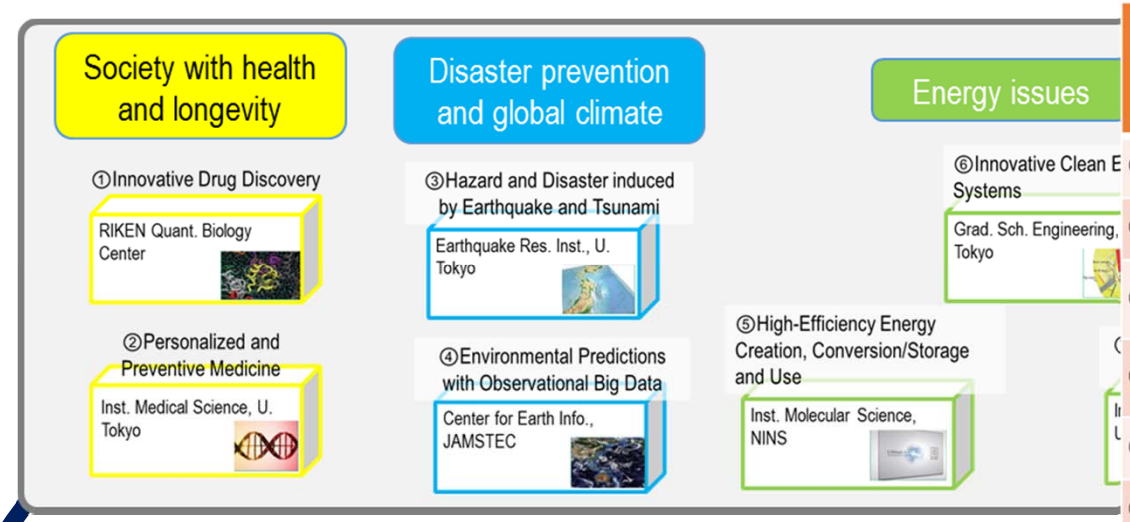


- **Developing a wide range of application codes, to run on the "post K", to solve major social and science issues**

The Japanese government selected 9 social & scientific priority issues and their R&D organizations.

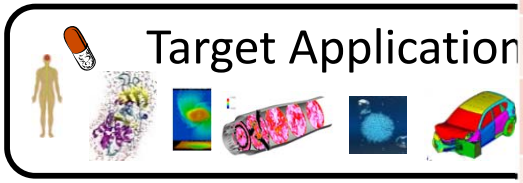


Co-design



Architectural Parameters

- #SIMD, SIMD length, #core, #NUMA node
- cache (size and bandwidth)
- memory technologies
- specialized hardware
- Interconnect
- I/O network

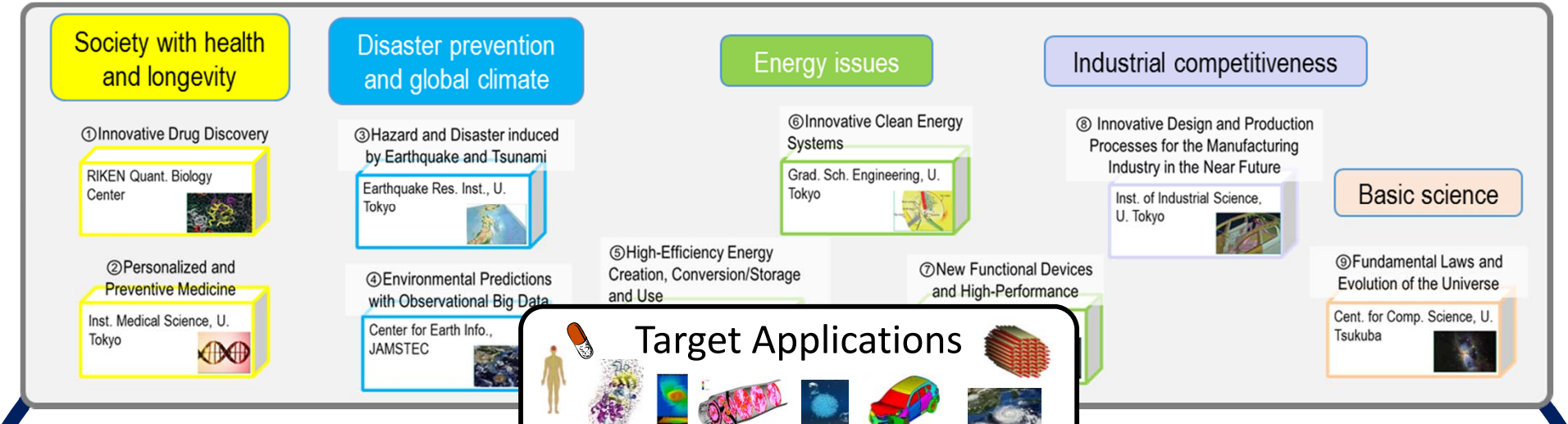


Target Application		
	Program	Brief description
①	GENESIS	MD for proteins
②	Genomon	Genome processing (Genome alignment)
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)
⑤	NTChem	molecular electronic (structure calculation)
⑥	FFB	Large Eddy Simulation (unstructured grid)
⑦	RSDFT	an ab-initio program (density functional theory)
⑧	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)
⑨	CCS-QCD	Lattice QCD simulation (structured grid Monte Carlo)

Target Applications' Characteristics

	Target Application		
	Program	Brief description	Co-design
①	GENESIS	MD for proteins	Collective comm. (all-to-all), Floating point perf (FPP)
②	Genomon	Genome processing (Genome alignment)	File I/O, Integer Perf.
③	GAMERA	Earthquake simulator (FEM in unstructured & structured grid)	Comm., Memory bandwidth
④	NICAM+LETK	Weather prediction system using Big data (structured grid stencil & ensemble Kalman filter)	Comm., Memory bandwidth, File I/O, SIMD
⑤	NTChem	molecular electronic (structure calculation)	Collective comm. (all-to-all, allreduce), FPP, SIMD,
⑥	FFB	Large Eddy Simulation (unstructured grid)	Comm., Memory bandwidth,
⑦	RSDFT	an ab-initio program (density functional theory)	Collective comm. (bcast), FPP
⑧	Adventure	Computational Mechanics System for Large Scale Analysis and Design (unstructured grid)	Comm., Memory bandwidth, SIMD
⑨	CCS-QCD	Lattice QCD simulation (structured grid Monte Carlo)	Comm., Memory bandwidth, Collective comm. (allreduce)

Co-design



Architectural Parameters

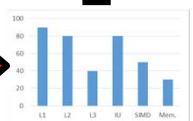
- #SIMD, SIMD length, #core,
- cache (size and bandwidth)
- memory technologies
- specialized hardware
- Interconnect
- I/O network

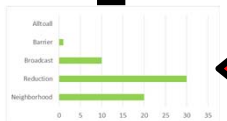
- Mutual understanding both computer architecture/system software and applications
- Looking at performance predictions
- Finding out the best solution with constraints, e.g., power consumption, budget, and space

Prediction of node-level performance

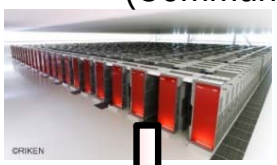
Profiling applications, e.g., cache misses and execution unit usages

Prediction Tool





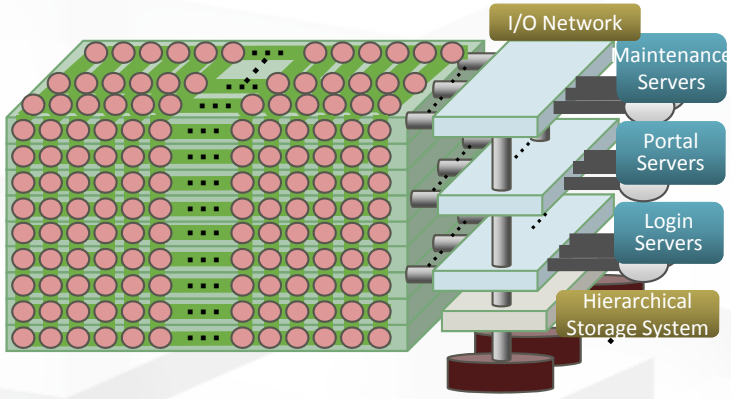
Prediction of scalability (Communication cost)



An Overview of post K

MC-kernel: a lightweight Kernel for manycore

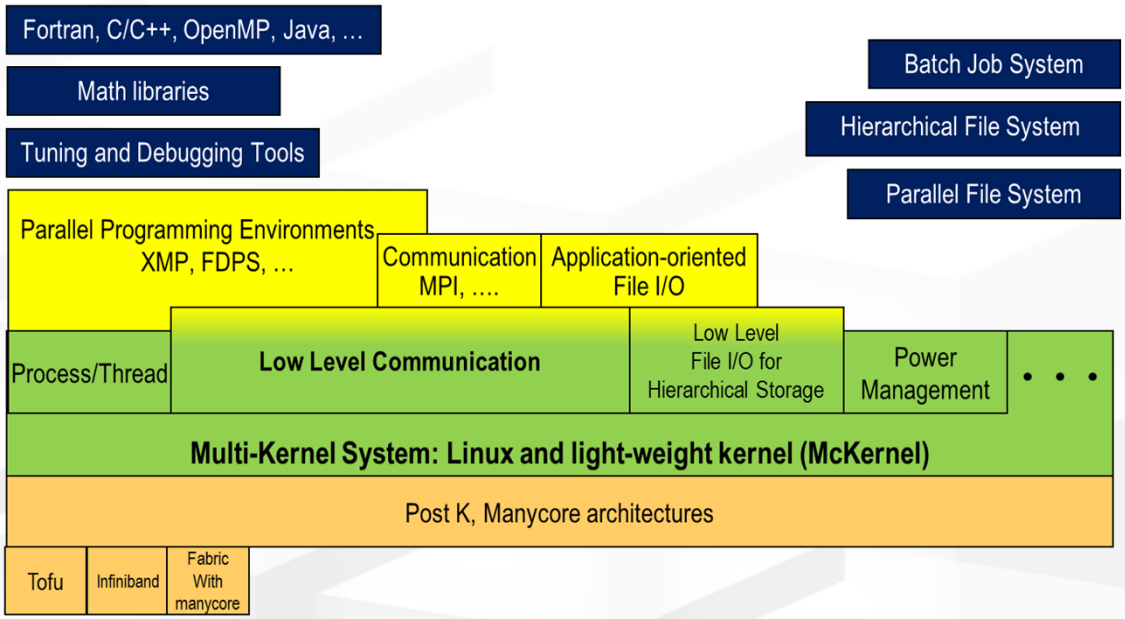
- **Hardware**
  - Manycore architecture
  - 6D mesh/torus Interconnect
  - 3-level hierarchical storage system
    - Silicon Disk
    - Magnetic Disk
    - Storage for archive



- **System Software**
  - Multi-Kernel: Linux with Light-weight Kernel
  - File I/O middleware for 3-level hierarchical storage system and application
  - Application-oriented file I/O middleware
  - MPI+OpenMP programming environment
  - Highly productive programming language and libraries

XcalableMP PGAS language

FPDS DSL

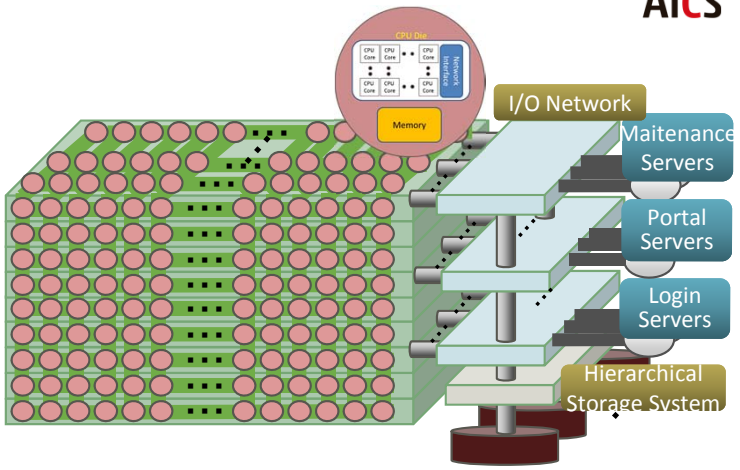


FLAGSHIP2020 Project

AICS

Missions

- Building the Japanese national flagship supercomputer, post K, and
- Developing wide range of HPC applications, running on post K, in order to solve social and science issues in Japan

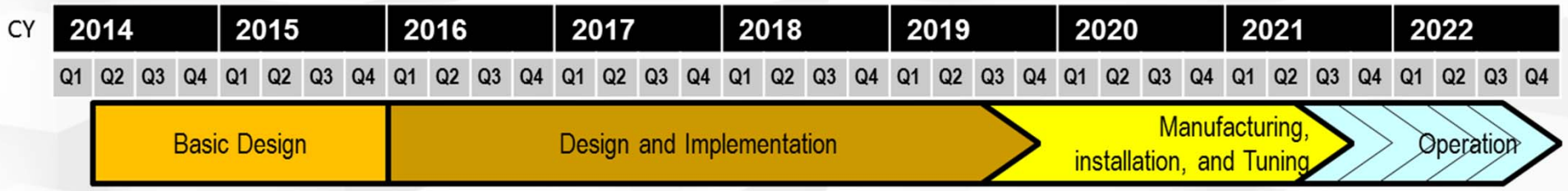


Project organization

- Post K Computer development
  - RIKEN AICS is in charge of development
  - Fujitsu is vendor partner.
  - International collaborations: DOE, JLESC, ..
- Applications
  - The government selected 9 social & scientific priority issues and their R&D organizations.

Status and Update

- "Basic Design" was finalized and now in "Design and Implementation" phase.
- We have decided to choose **ARM v8 with SVE as ISA** for post-K manycore processor.
- Some delay of delivery will be expected.



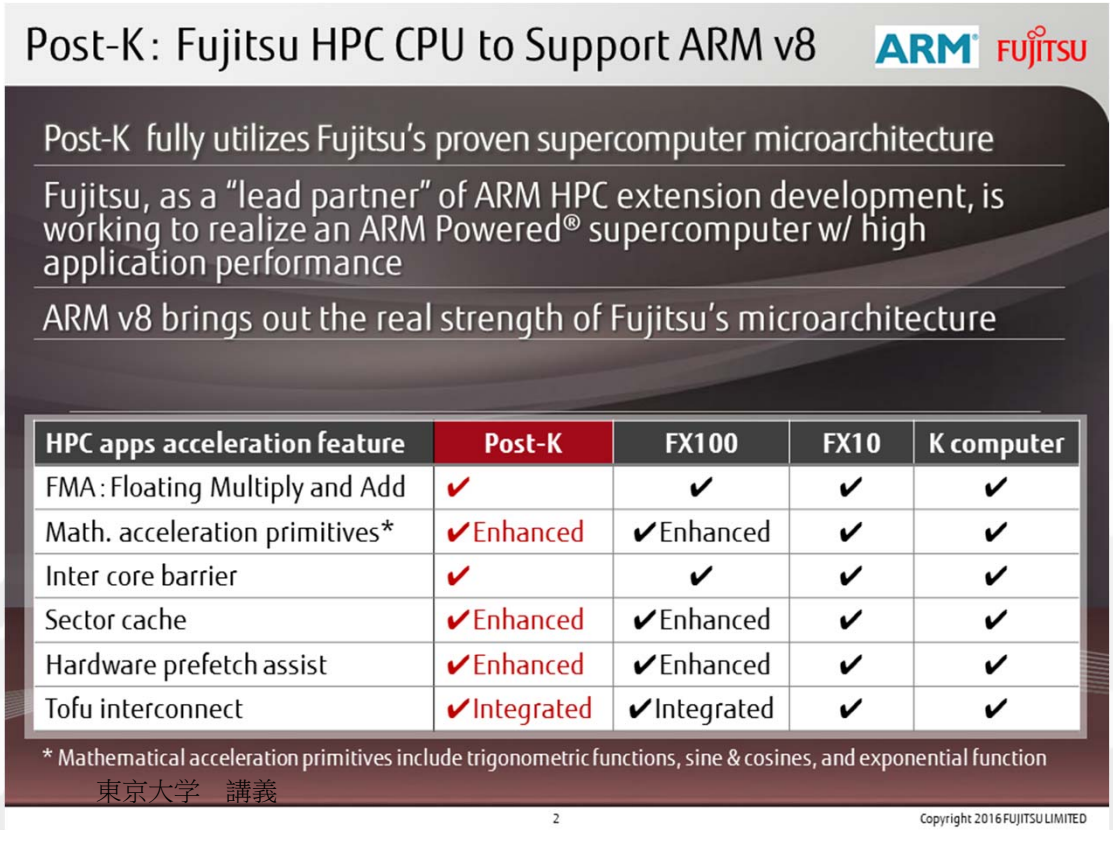
Instruction Set Architecture

- **ARM V8 with HPC Extension SVE**
  - Fujitsu is a lead partner of ARM HPC extension development
  - Detailed features were announced at Hot Chips 28 - 2016

    http://www.hotchips.org/program/
    Mon 8/22 Day1 9:45AM GPUs & HPCs
    “ARMv8-A Next Generation Vector Architecture for HPC”

SVE (Scalable Vector Extension)

- **Fujitsu’s additional support**
  - FMA
  - Math acceleration primitives
  - Inter-core hardware-suppoted barrier
  - Sector cache
  - Hardware prefetch assist



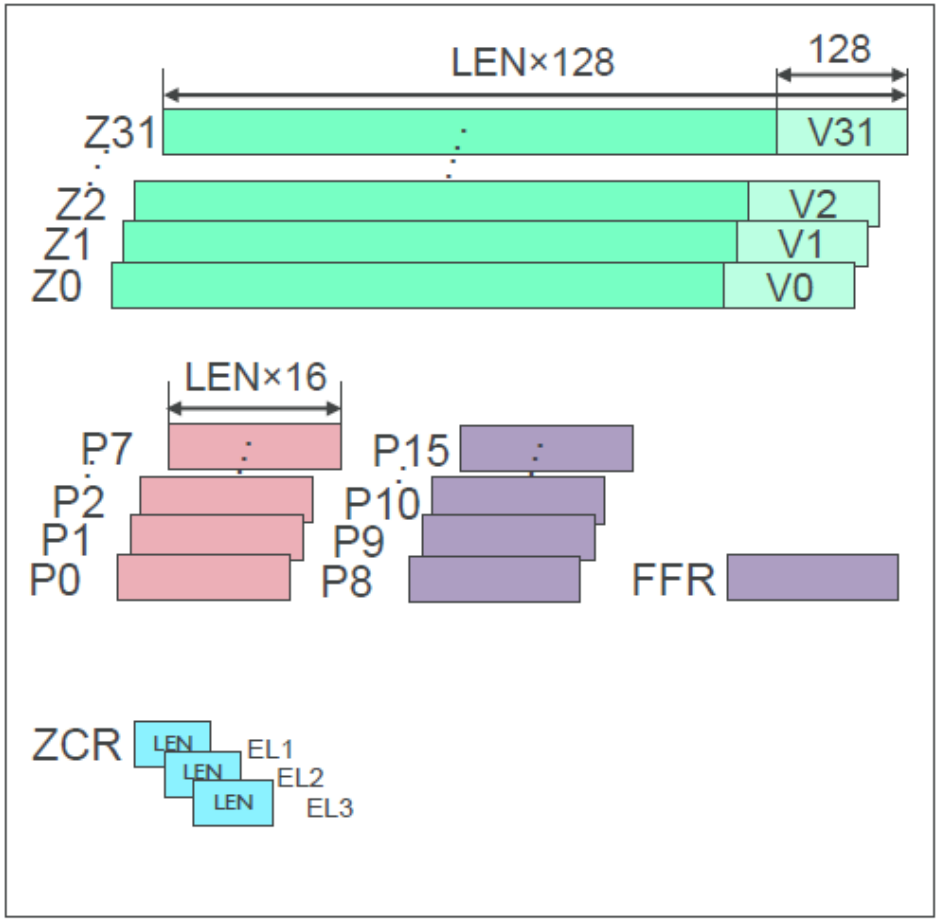
HPC apps acceleration feature	Post-K	FX100	FX10	K computer
FMA: Floating Multiply and Add	✔	✔	✔	✔
Math. acceleration primitives*	✔Enhanced	✔Enhanced	✔	✔
Inter core barrier	✔	✔	✔	✔
Sector cache	✔Enhanced	✔Enhanced	✔	✔
Hardware prefetch assist	✔Enhanced	✔Enhanced	✔	✔
Tofu interconnect	✔Integrated	✔Integrated	✔	✔

ARM v8 Scalable Vector Extension (SVE)

- **SVE is a complementary extension that does not replace NEON, and was developed specifically for vectorization of HPC scientific workloads.**
- **The new features and the benefits of SVE comparing to NEON**
  - **Scalable vector length (VL)** : Increased parallelism while allowing implementation choice of VL
  - **VL agnostic (VLA) programming**: Supports a programming paradigm of write-once, run-anywhere scalable vector code
  - **Gather-load & Scatter-store**: Enables vectorization of complex data structures with non-linear access patterns
  - **Per-lane predication**: Enables vectorization of complex, nested control code containing side effects and avoidance of loop heads and tails (particularly for VLA)
  - Predicate-driven loop control and management: Reduces vectorization overhead relative to scalar code
  - Vector partitioning and SW managed speculation: Permits vectorization of uncounted loops with data-dependent exits
  - Extended integer and floating-point horizontal reductions: Allows vectorization of more types of reducible loop-carried dependencies
  - Scalarized intra-vector sub-loops: Supports vectorization of loops containing complex loop-carried dependencies

SVE architectural state

- Scalable vector registers
  - Z0-Z31 extending NEON's V0-V31
    - DP & SP floating-point
    - 64, 32, 16 & 8-bit integer
- Scalable predicate registers
  - P0-P7 lane masks for ld/st/arith
  - P8-P15 for predicate manipulation
  - FFR *first fault register*
- Scalable vector control registers
  - ZCR_ELx vector length (LEN=1..16)
  - Exception / privilege level EL1 to EL3



SVE example

DAXPY (scalar)

```
// ----------------------------------------
//       subroutine daxpy(x,y,a,n)
//       real*8 x(n),y(n),a
//       do i = 1,n
//          y(i) = a*x(i) + y(i)
//       enddo
// ----------------------------------------
// x0 = &x[0], x1 = &y[0], x2 = &a, x3 = &n
daxpy_:
    ldrsw   x3, [x3]            // x3=*n
    mov     x4, #0              // x4=i=0
    ldr     d0, [x2]            // d0=*a
    b       .latch
.loop:
    ldr     d1, [x0,x4,lsl 3]   // d1=x[i]
    ldr     d2, [x1,x4,lsl 3]   // d2=y[i]
    fmadd   d2, d1, d0, d2      // d2+=x[i]*a
    str     d2, [x1,x4,lsl 3]   // y[i]=d2
    add     x4, x4, #1          // i+=1
.latch:
    cmp     x4, x3              // i < n
    b.lt    .loop               // more to do?
    ret
```

DAXPY (SVE)

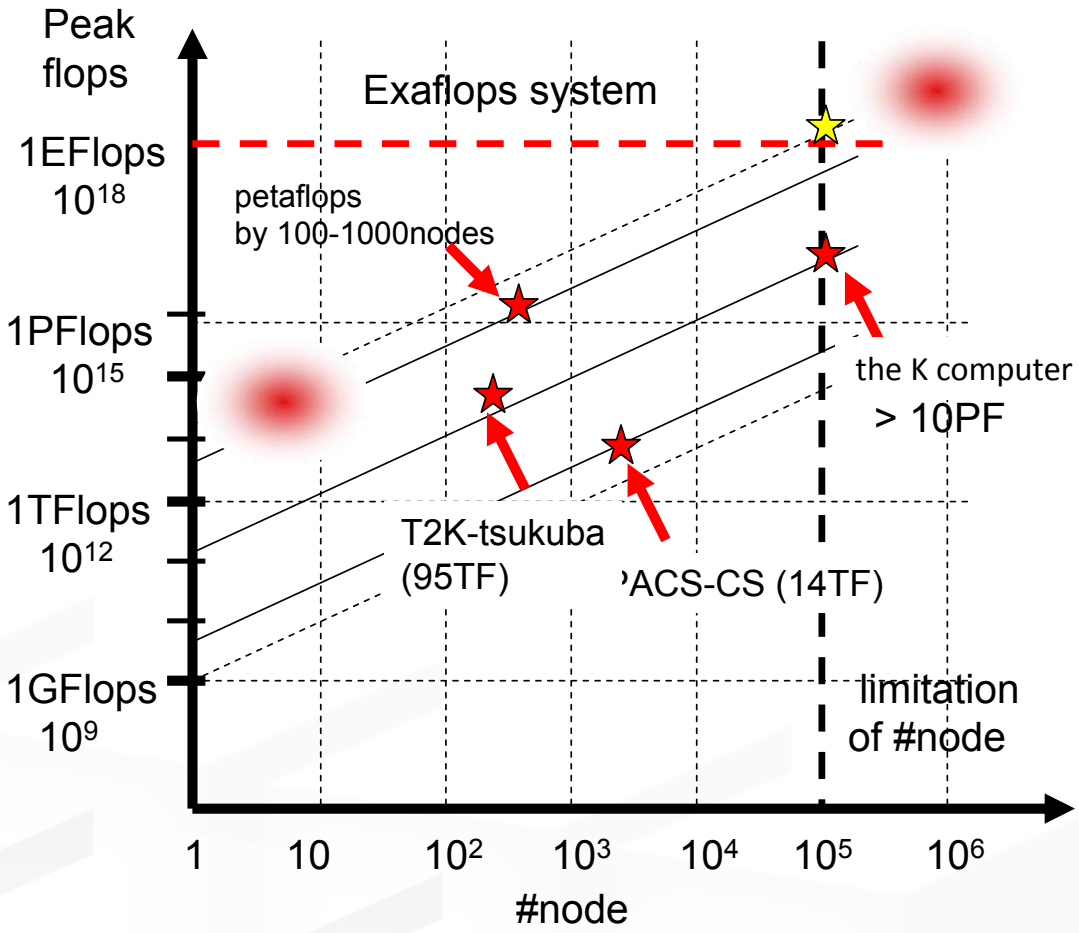
```
// ----------------------------------------
//       subroutine daxpy(x,y,a,n)
//       real*8 x(n),y(n),a
//       do i = 1,n
//          y(i) = a*x(i) + y(i)
//       enddo
// ----------------------------------------
// x0 = &x[0], x1 = &y[0], x2 = &a, x3 = &n
daxpy_:
    ldrsw   x3, [x3]            // x3=*n
    mov     x4, #0              // x4=i=0
    whilelt p0.d, x4, x3        // p0=while(i++<n)
    ld1rd   z0.d, p0/z, [x2]    // p0:z0=bcast(*a)
.loop:
    ld1d    z1.d, p0/z, [x0,x4,lsl 3] // p0:z1=x[i]
    ld1d    z2.d, p0/z, [x1,x4,lsl 3] // p0:z2=y[i]
    fmla    z2.d, p0/m, z1.d, z0.d    // p0?z2+=x[i]*a
    st1d    z2.d, p0, [x1,x4,lsl 3]   // p0?y[i]=z2
    incd    x4                        // i+=(VL/64)
.latch:
    whilelt p0.d, x4, x3        // p0=while(i++<n)
    b.first .loop               // more to do?
    ret
```

- **Compact code for SVE as scalar loop**
- **OpenMP SIMD directive is expected to help the SVE programming**

エクサスケール・コンピューティングの課題

Issues for exascale computing

- **Important aspects of post-petascale computing**
  - Large-scale system
    - < 10^6 nodes, for FT
  - Strong-scaling
    - > 10TFlops/node
    - accelerator, many-cores
  - Power limitation
    - < 20-30 MW



Simple relationship between #nodes and node performance to achieve exascale

A projection: Pre-exa, exa, post-exa

	Pre-exa	exascale	Post-exa
System performance (PF)	50～500	500～5,000	1,000～10,000
node performance (TF)	1～10	5～50	10～100
#number of node (K)	5～500	10～1,000	10～1,000
Performance/ power(GF/W)	2～20	20～200?	400?
Memory bandwidth and technology	0.5～1TB/s (HBM) 150GB/s (DDR4)	1～4TB/s (HBM)	???

- Node performance must increase! Because the system scale is limited by space and power.
- Memory performance will be limited. So, the cap between B/F will be getting worse.
- Improvement of performance/power will be difficult and limited.

Top500の動向　（１）

- **Top500から、近年のスパコンの進歩の停滞が指摘されている。**
  - 性能の伸びが、これまでの年率1.9%から1.2倍に。
  - 2014年11月(SC14)のリストでは、1位から9位までは変化がなかった。Top500に新規に入るシステムの数が激減（これまでの200～150システムから80程度に）
  - 2010年頃までは上位50～70システムの性能合計が全システムの性能合計の半分を占めるという状況であった。しかし、このところ上位10～30システムで半分を占めるという状況に。

Top500: 全世界のスパコンの性能をLinpackと呼ばれるプログラムでランキングしたもの。例年6月と１１月に更新される。http://www.top500.org

参考：

http://news.mynavi.jp/articles/2014/12/10/sc14_top500bof/　http://www.cnet.com/news/top500-supercomputer-race-hits-a-slow-patch/

top500の動向　（２）

- **システムの性能の伸びに比べ、プロセッサの性能は伸びていない。**
  - 性能は、プロセッサの個数の増加（大規模化）、アクセラレータ（メニ・コアを含む）による。
  - プロセッサ自体の性能は伸びが鈍っている。Intel メニーコア、NVIDA GPU等の最新プロセッサのデリバリの遅れ。ムーアーの法則のスローダウン
- **大規模化により、電力性能の重要性が顕著になっている**
  - Top10システムでは2000GFlops/kW程度になっているのに対して、Top50システムでは1500GFlops/kW、Top500システム全体では1000GFlops/kWとTop10に比べて半分の効率でしかない。

- **米国、中国、日本で、2015年から2017年頃に、数１０～数100PFlops級のスパコンの設置計画あり、現在のスローダウンは一時的なものであるという見方もある。**
- **国別のシステム数では、米国がほぼ半分の46%を占め、中国が12%。日本、英国、フランス、ドイツの各国が5～6%。**
- **Top500は主にＣＰＵ性能のみで、ワークロードを反映していないという意見から、HPCGやgraph500での評価にも興味が集まっている。**

電力効率の現状

- **Green 500: HPL (Linpack)の実行時の電力性能をランキング**

2016/Nov

Green500 Rank	MFLOPS/W	Site	System	Total Power(kW)
1	9462.1	NVIDIA Corporation	NVIDIA DGX-1, Xeon E5-2698v4 20C 2.2GHz, Infiniband EDR, NVIDIA Tesla P100	349.5
2	7453.5	Swiss National Supercomputing Centre (CSCS)	Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100	1312
3	6673.8	Advanced Center for Computing and Communication, RIKEN	ZettaScaler-1.6, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SCnp	150.0
4	6051.3	National Supercomputing Center in Wuxi	Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway	15371
5	5806.3	Fujitsu Technology Solutions GmbH	PRIMERGY CX1640 M1, Intel Xeon Phi 7210 64C 1.3GHz, Intel Omni-Path	77
6	4985.7	Joint Center for Advanced High Performance Computing	PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, Intel Omni-Path	2718.7
7	4688.0	DOE/SC/Argonne National Laboratory	Cray XC40, Intel Xeon Phi 7230 64C 1.3GHz, Aries interconnect	1087

エクサスケールを20MWで達成するためには、50GF/Wが必要

Multi-core processor：
Solution of Low power by parallel processing

CPU power dissipation

$$P = N \times \alpha \times C \times V^2 \times f$$

# CPU	Active rate of processors	Capacitance of circiuit	Voltage	Clock Freq

Apporach for Low power by parallel processing

increase N、⇧ decrease V and f, ⇩ improve perf. N × f ⇧

- **Decreasing V and F, makes heat dissipation and power lower within a chip**
- Progress in silicon technology 130nm ⇒ 90nm⇒65nm,22nm （Decrease **C** and **V)**
- Use a silicon process for low power (embedded processor)（Small **α**）

- Perfrmance improvement by Multi-core（N=2～16)
  - Number of transistors are increasing by “Moore’s Law”
- Parallel processing by low power processor

「演算加速機構を持つ
将来のHPCIシステムに関する調査研究」
最終報告

主管事業実施機関： 筑波大学
計算科学研究センター

共同事業参画機関：東京工業大学、理化学研究所、
会津大学、日立製作所
協力機関：東京大学、広島大学、
高エネルギー加速器研究機構

「演算加速機構を持つ将来のHPCIシステムに関する調査研究」

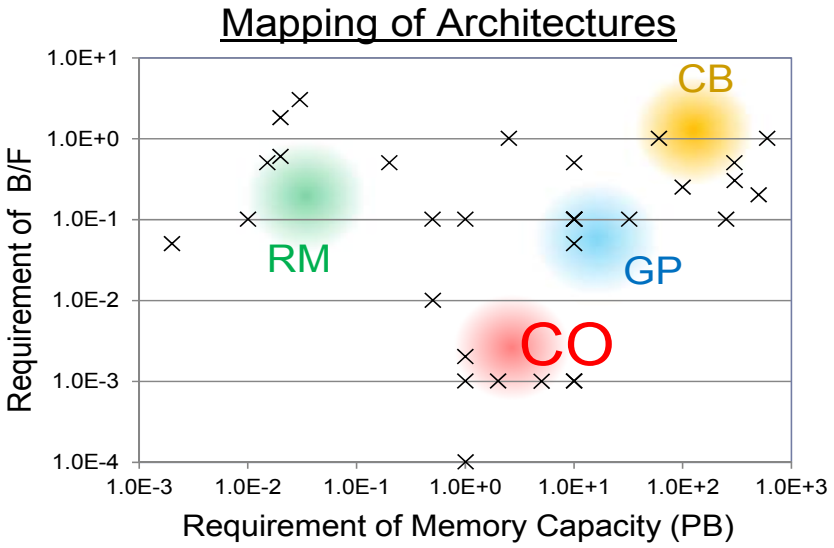
- ナノテクやライフサイエンスの進歩、気候気象予測や地震・防災への対処には計算科学は不可欠かつ有効な手段
  - そのためにはさらなる計算能力が要請されている。
  - 設置面積、消費電力等の制限からノード数の増加による並列システムの性能向上には限界
- ライフサイエンスの分子シミュレーション等、多様な分野で比較的小さい一定サイズの問題の高速化が望まれている(強スケーリング)
  - 対応した研究開発の例: ANTON, MDGRAPE-4

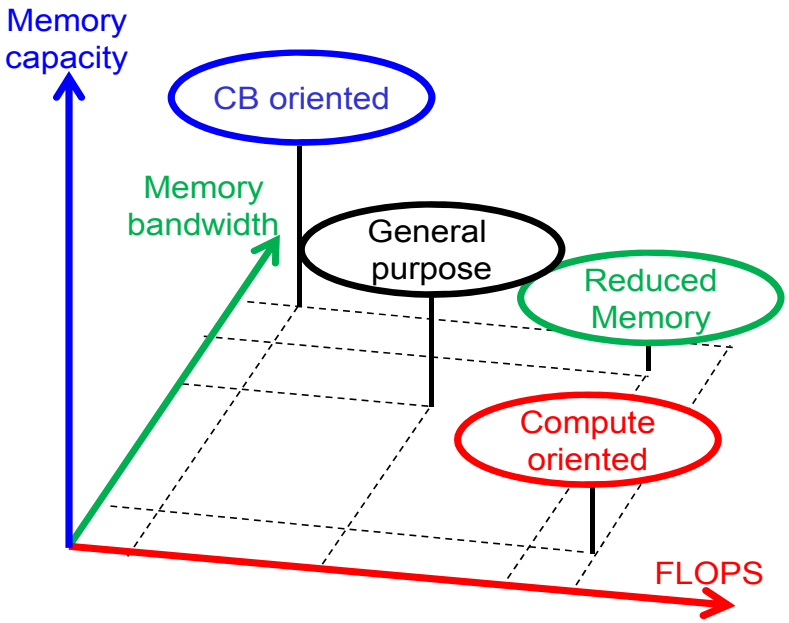
電力効率の大幅な効率化と強スケーリング問題の高速化による新たな計算科学の展開を目指して、演算加速機構による並列大規模システムについて調査研究を行う。

- 計算科学に対する社会的・科学的課題の達成のために必要なアプリケーションのうち、本調査研究で対象とするメモリ削減型(RM)および演算重視型(CO)で、ある程度の実行効率が期待できるもの
  - 生命科学、物性科学における分子動力学計算、生命科学、物性科学、ものづくり分野における第一原理計算、素粒子物理における格子QCD、原子核物理における様々な手法、宇宙物理における粒子シミュレーション、流体計算等(合同作業部会報告より)

- 強スケーリングによる分子動力学アプリケーションの実時間の大幅な高速化
- 電力効率の大幅な効率化による格子QCD等のメモリ削減型アプリケーションの大規模・効率的実行

（合同作業部会報告より抜粋）



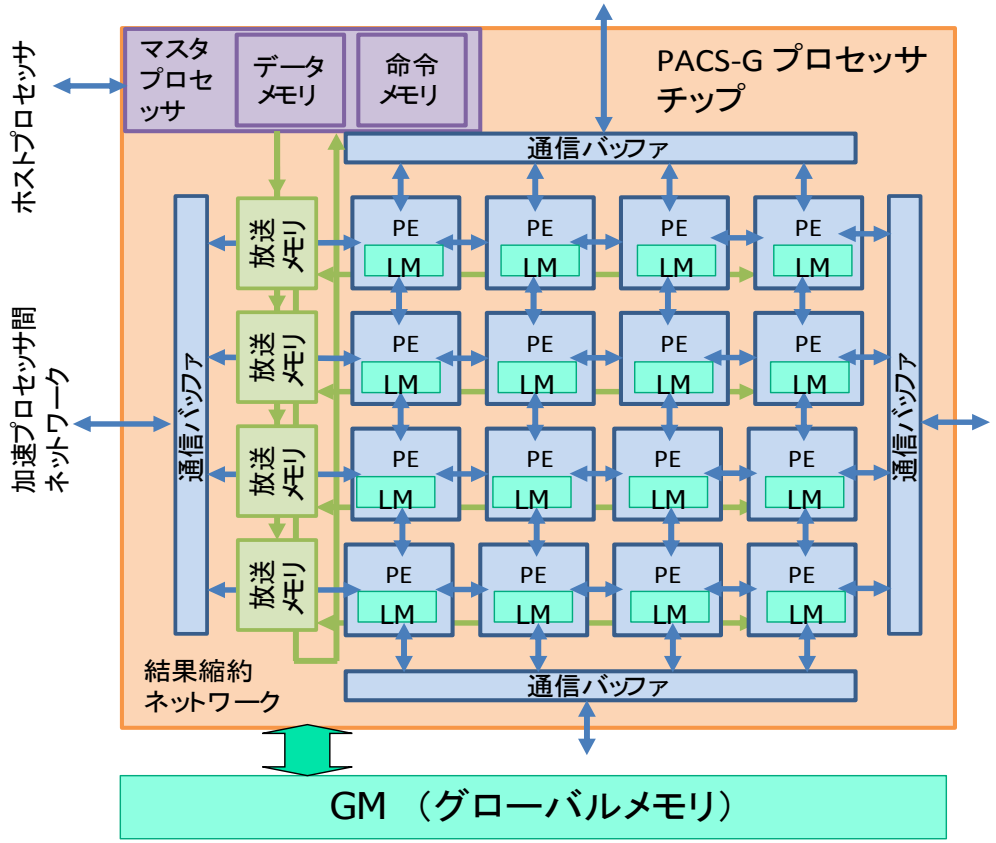


PACS-G アーキテクチャの概要

提案アーキテクチャ　PACS-Gの特徴：

- 演算集約型とメモリ削減型のステンシル計算を両立させるアーキテクチャ（プロセッサ、ネットワーク）をターゲットに設定
- 加速プロセッサは、多数のPE（コア）を内蔵し、SIMD方式で制御。これにより、多数のコアによる演算性能の向上、並列制御の簡略化と、大幅な電力の大幅な削減を実現。
- PEは、演算ユニットとオンチップのローカルメモリ(LM)からなり、ローカルメモリ上のデータを処理。
- 加速プロセッサチップ間は専用ネットワークを持つことにより、低レーテンシ通信を可能とし、アプリケーションの効率実行、強スケーリング化を可能とする。
- 2018～2020年のLSIテクノロジーとして、10nm (FinFET) を想定。チップサイズを20mm x 20 mm程度を想定。

- マスタプロセッサは、通常のレーテンシコアを想定。PEは、マスタプロセッサからSIMD命令として制御される。（GPUとは大きく異なる）
- 1プロセッサチップあたりのPE数は、2048～4096程度
- プロセッサチップには、外付けのグローバルメモリ（GM）をつけることを想定。
  - TSV 2.5次元実装によるHBM、もしくはHMCを検討
  - 外付けメモリとして、DDR/DIMは想定しない。
  - PEからはブロック転送のみ、ランダムアクセスはなし
- チップ内のネットワークは、4次元の隣接通信を可能とするネットワークを検討（図は、2次元メッシュの例）
- PE内の縮約操作、ブロードキャストのためのネットワーク・メモリを想定
- 電力は50GF/W以上を目標



Challenges of Programming Languages/models for exascale computing

- Scalability, Locality and scalable Algorithms in system-wide
- Strong Scaling in node
- Workflow and Fault-Resilience
- (Power-aware)

Programming model : “MPI+X” for exascale?

- **X is OpenMP!**
- **“MPI+Open” is now a standard programming for high-end systems.**
  - I’d like to celebrate that OpenMP became “standard” in HPC programming
- **Questions:**
  - “MPI+OpenMP” is still a main programming model for exa-scale?

Question

- **What happens when executing code using all cores in manycore processors like this ?**

```
 MPI_recv …
#pragma omp parallel for
for ( … ; … ; … ) {
      … computations …
}
 MPI_send …
```

Data comes into “main shared memory”

Cost for “fork” become large

data must be taken from Main memory

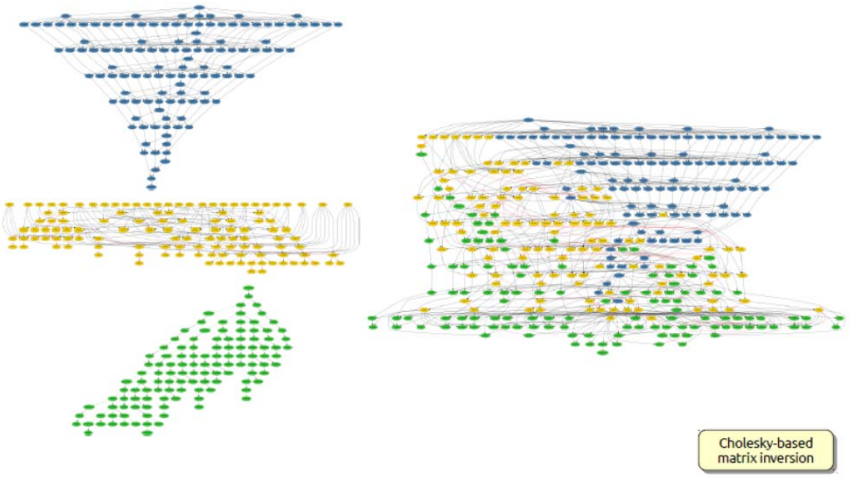
Cost for “barrier” become large

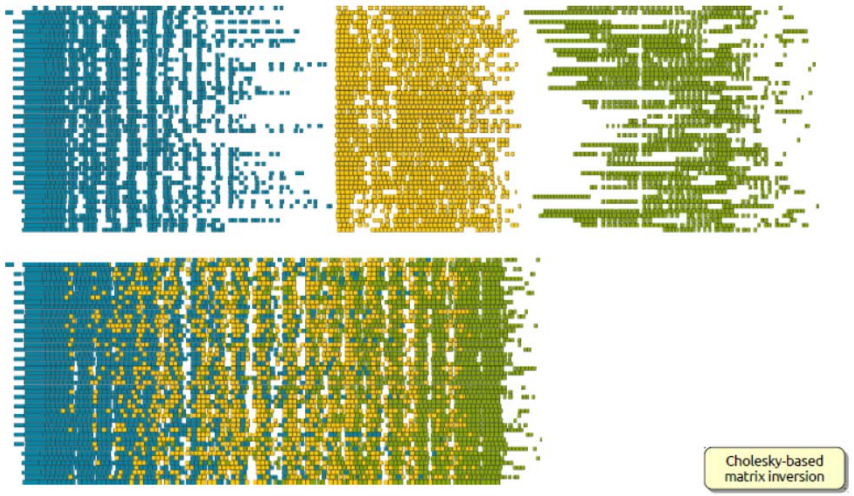
MPI must collect data from each core to send

- **What are solutions?**
  - MPI+OpenMP runs on divided small “NUMA domains” rather than all cores?

Multitasking model

- **Multitasking/Multithreaded execution: many “tasks” are generated/executed and communicates with each others by data dependency.**
  - OpenMP task directive, OmpSS, PLASMA/QUARK, StarPU, ..
  - Thread-to-thread synchronization /communications rather than barrier
- **Advantages**
  - Remove barrier which is costly in large scale manycore system.
  - Overlap of computations and computation is done naturally.
  - New communication fabric such as Intel OPA (OmniPath Architecture) may support core-to-core communication that allows data to come to core directly.
- **New algorithms must be designed to use multitasking**

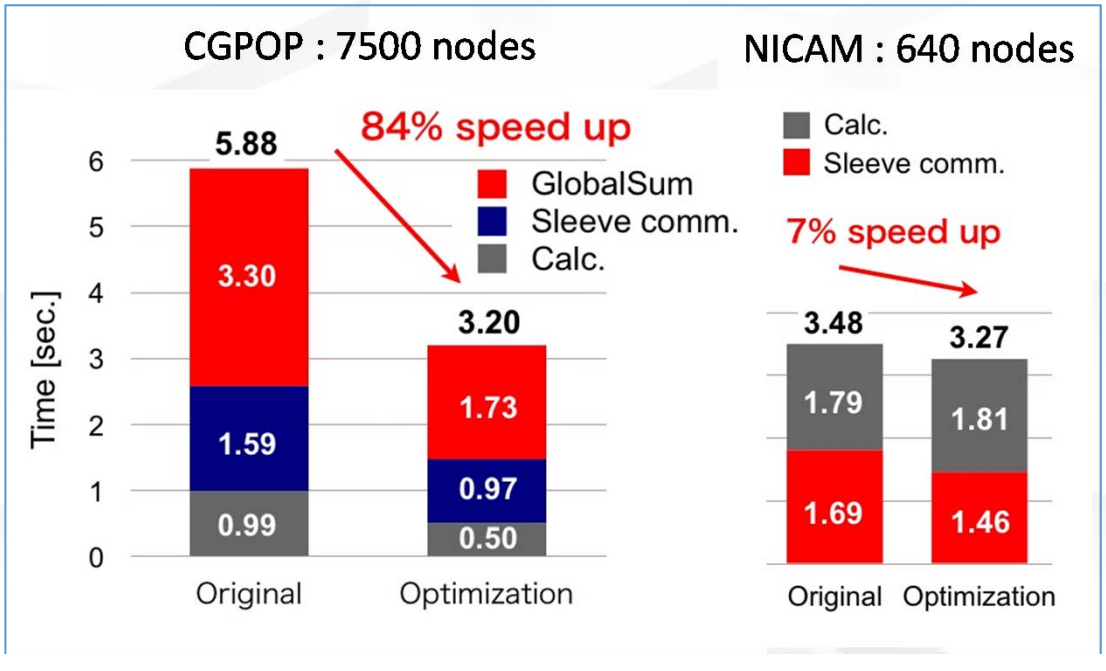




From PLASMA/QUARK slides by ICL, U. Teneessee

PGAS (Partitioned Global Address Space) models

- Light-weight one-sided communication and low overhead synchronization semantics.
- PAGS concept is adopted in Coarray Fortran, UPC, X10, XMP.
  - XMP adopts notion Coarray not only Fortran but also “C”, as “local view” as well as “global view” of data parallelism.
- Advantages and comments
  - Easy and intuitive to describe, not noly one side-comm, but also strided comm.
  - Recent networks such as Cray and Fujitsu Tofu support remote DMA operation which strongly support efficient one-sided communication.
  - Other collective communication library (can be MPI) are required.



Case study of XMP on K computer
CGPOP, NICAM: Climate code

5-7 % speed up is obtained by replacing MPI with Coarray

Strong Scaling in node

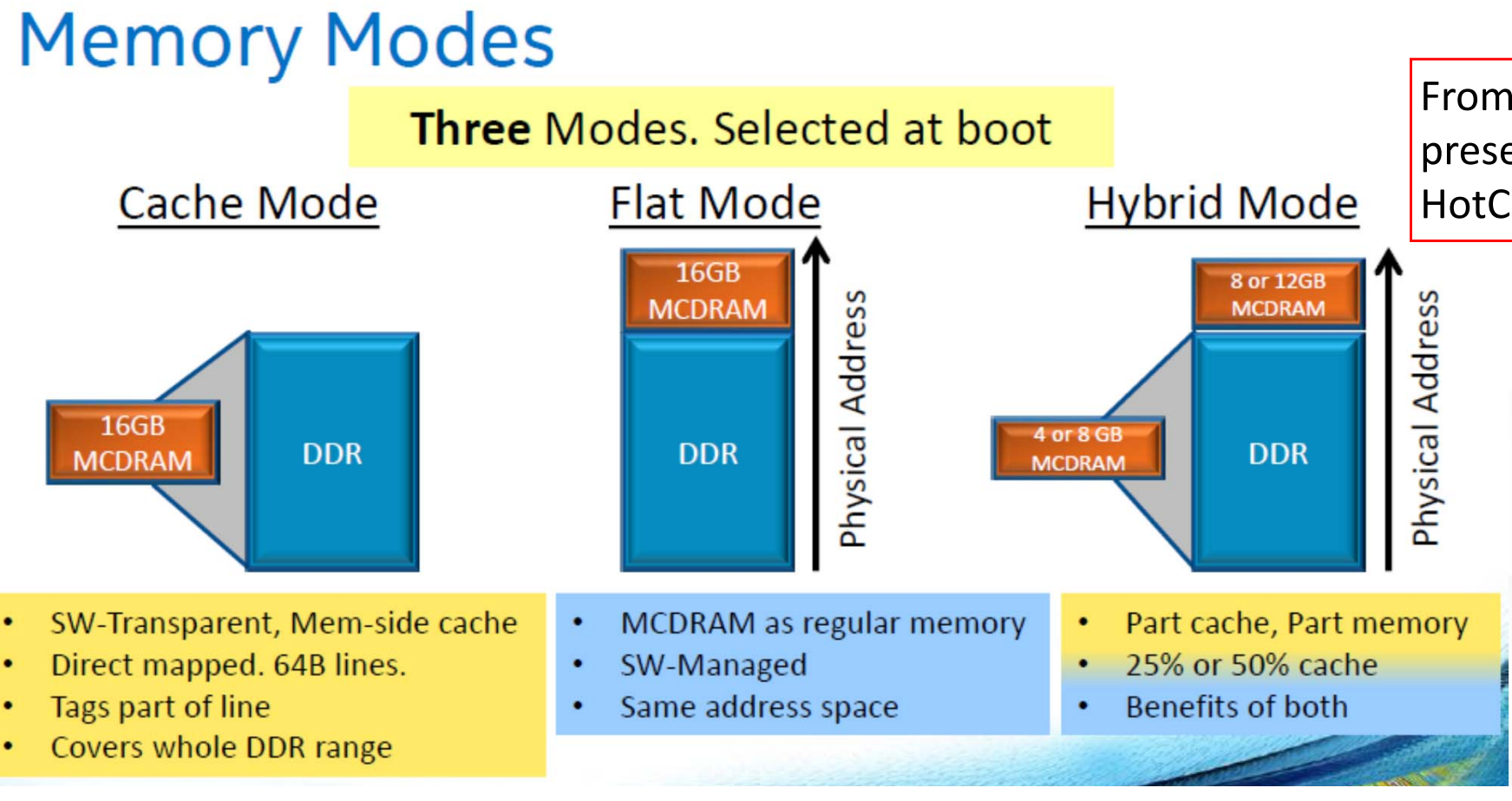
- **Two approaches:**
  - SIMD for core in manycore processors
  - Accelerator such as GPUs

- **Programming for SIMD**
  - Vectorization by directives or automatic compiler technology
  - Limited bandwidth of memory and NoC
  - Complex memory system: Fast-memory (MD-DRAM, HBM, HMC) and DDR , VMRAM

- **Programming for GPUs**
  - Parallelization by OpenACC/OpenMP 4.0. Still immature but getting matured soon
  - Fast memory (HMB) and fast link (NV-Link): similar problem of complex memory system in manycore.
  - Programming model to be shared by manycore and accelerator for high productivity.

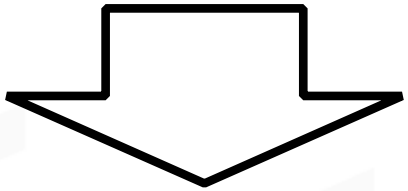
How to use MC-DRAM in KNL?

- **New Xeon Phi (KNL) has fast memory called MC-DRAM.**
  - KNL performance: < 5 TF (Theoretical Peak)
  - DDR4: 100～200 GB/s, MC-DRAM: 0.5 TB/s
  - How to use?



Prog. Models for Workflow and data managements

- Petascale system was targeting some of “capability” computing.
- In exascale system, it become important to execute huge number of medium-grain jobs for parameter-search type applications.



Workflow to control and collect/process data is important, also for “big-data” apps.

Concluding Remarks

- **FLAGSHIP 2020 project**
  - To develop the next Japanese flagship computer system, post-K
  - The basic architecture design was finalized and now in detail design and implementation phase.
  - “Co-design” effort will be continued (application design for architecture)
  - We expect that ARM SVE will deliver high-performance and flexible SIMD-vectorization to our “post-K” manycore processor.

- **エクサスケールに向けて**
  - さらに電力性能を高める必要がある。
  - そのためには、アーキテクチャはメニーコアや演算加速機構的な形態、また、メモリバンド幅が小さくなり、そのためのプログラミング、アルゴリズムの研究が必要