Overview of the Post-K processor

ポスト京システムの概要と開発進捗状況

Mitsuhisa Sato Team Leader of Architecture Development Team

Deputy project leader, FLAGSHIP 2020 project
Deputy Director, RIKEN Center for Computational Science (R-CCS)

Professor (Cooperative Graduate School Program),
University of Tsukuba

FLAGSHIP2020 Project

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

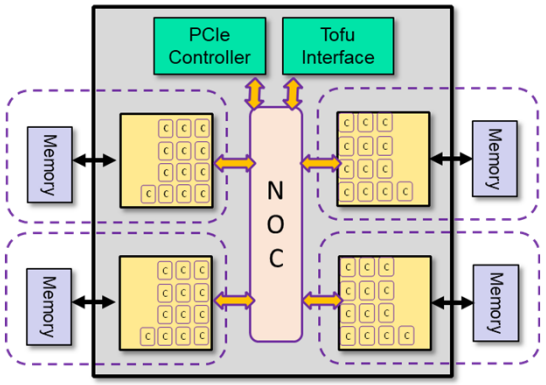
Overview of post-K architecture

Node: Manycore architecture

- Armv8-A + SVE (Scalable Vector Extension)
- SIMD Length: 512 bits
- # of Cores: 48 + (2/4 for OS) (> 2.7 TF / 48 core)
- Co-design with application developers and high memory bandwidth utilizing on-package stacked memory (HBM2) 1 TB/s B/W
- Low power : 15GF/W (dgemm)

Network: TofuD

- Chip-Integrated NIC, 6D mesh/torus Interconnect

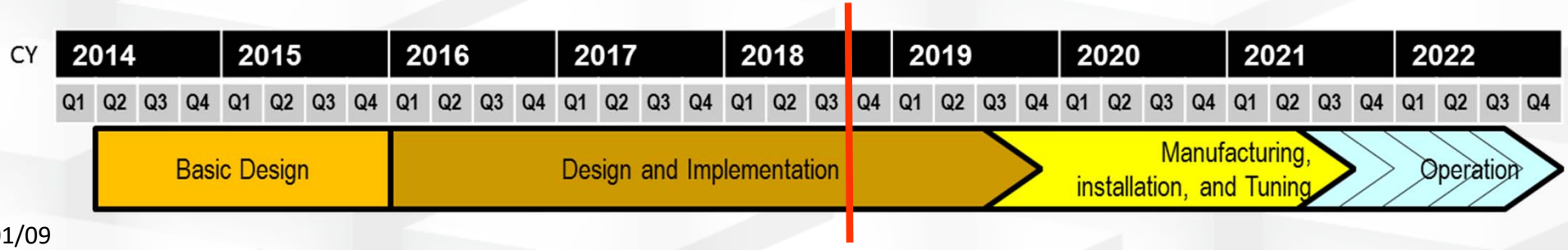


Post-K processor

Prototype board

Status and Update

- Close to end in “Design and Implementation”.
- The prototype CPU powered-on and development is as scheduled
- RIKEN announced the Post-K early access program to begin around Q2/CY2020
- We are working on performance evaluation and tuning by simulators and compilers



KPIs on post-K development in FLAGSHIP 2020 project

3 KPIs (key performance indicator) were defined for post-K development

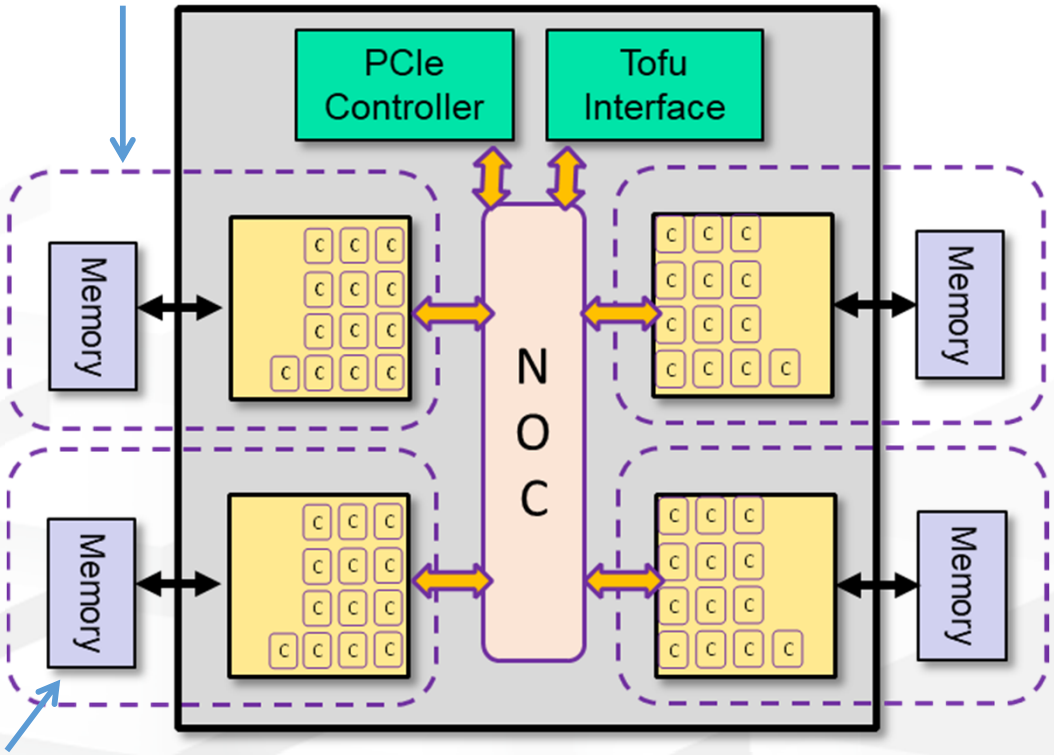
- **1. Extreme Power-Efficient System**
  - 30-40 MW at system level
- **2. Effective performance of target applications**
  - It is expected to exceed 100 times higher than the K computer's performance in some applications
- **3. Easy-of-use system for wide-range of users**

CPU Architecture: A64FX

- **Armv8.2-A (AArch64 only) + SVE (Scalable Vector Extension)**
  - FP64/FP32/FP16 (https://developer.arm.com/products/architecture/a-profile/docs)
- **SVE 512-bit wide SIMD**
- **# of Cores: 48 + (2/4 for OS)**
- Co-design with application developers and high memory bandwidth utilizing on-package stacked memory: **HBM2(32GiB)**
- Leading-edge Si-technology (7nm FinFET), low power logic design **(approx. 15 GF/W (dgemm))**, and power-controlling knobs
- PCIe Gen3 16 lanes
- Peak performance
  - > 2.7 TFLOPS (>90% @ dgemm)
  - Memory B/W 1024GB/s (>80% stream)
  - Byte per Flops: approx. 0.4

- "Common" programing model will be to run each MPI process on a NUMA node (CMG) with OpenMP-MPI hybrid programming.
- 48 threads OpenMP is also supported.

CMG(Core-Memory-Group): NUMA node
12+1 core



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 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)

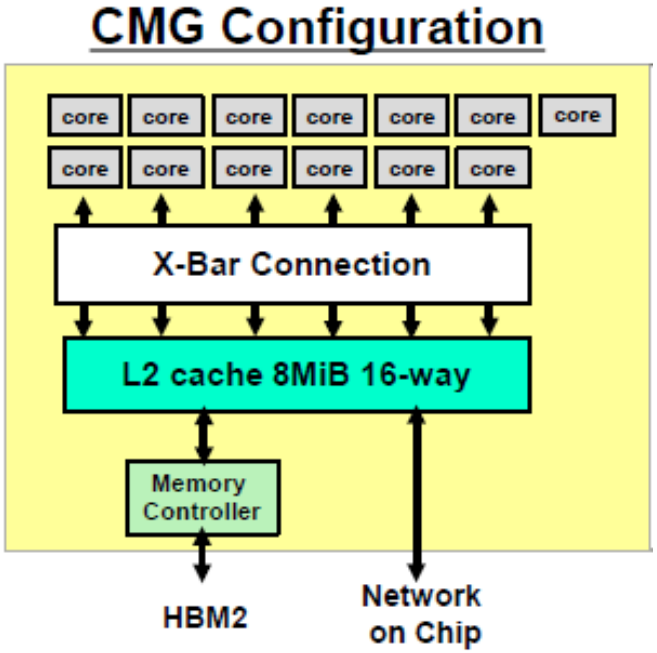
```
// ----------------------------------------
//       subroutine daxpy(x,y,a,n)
//       real*8 x(n),y(n),a
//       do i = 1,n
//          y(i) = a*x(i) + y(i)
//       enddo
// ----------------------------------------
// x0 = &x[0], x1 = &y[0], x2 = &a, x3 = &n
daxpy_:
    ldrsw   x3, [x3]            // 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
```

Make predicate mask

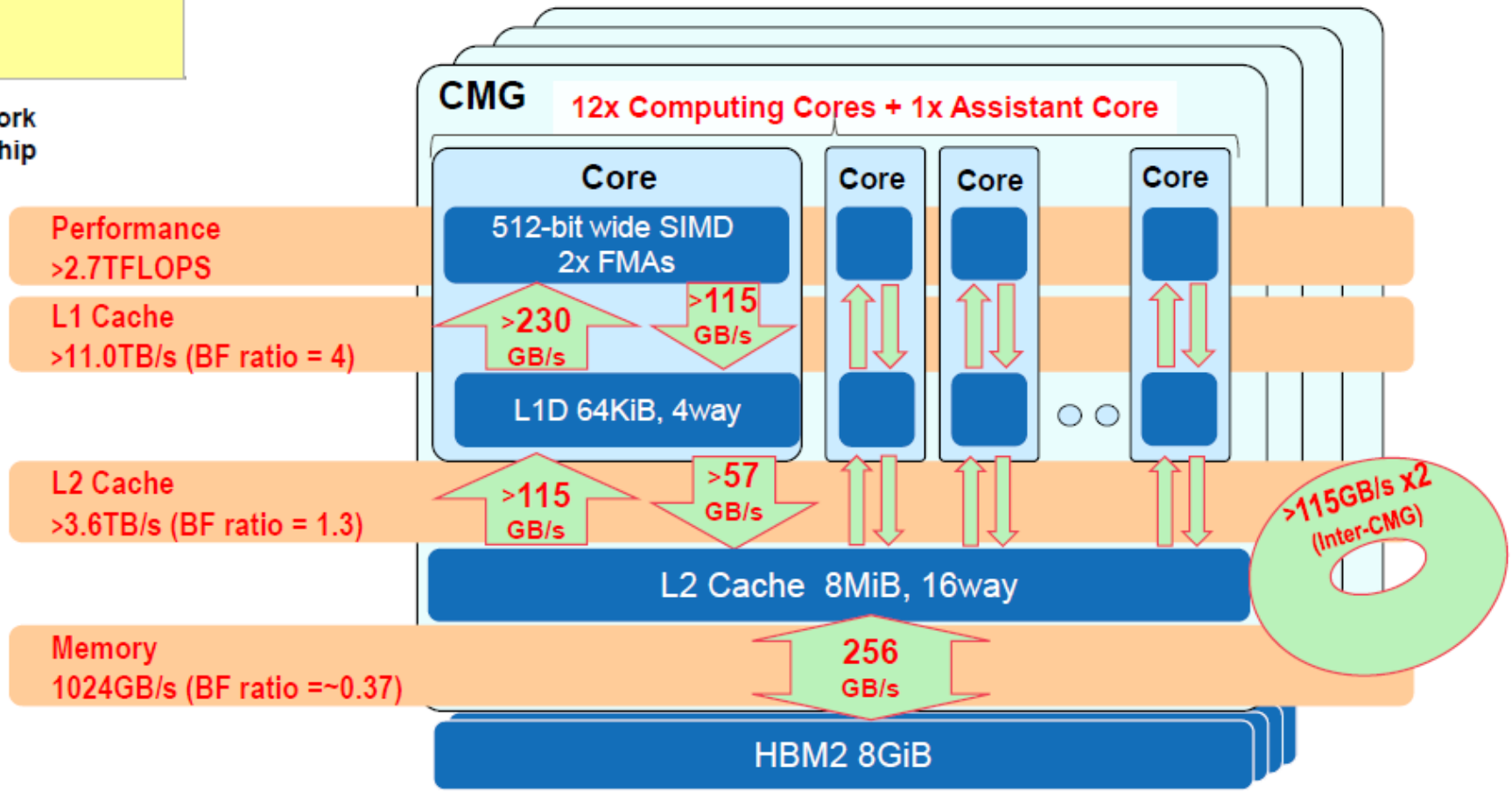
SIMD with mask

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

CMG (Core Memory Group)



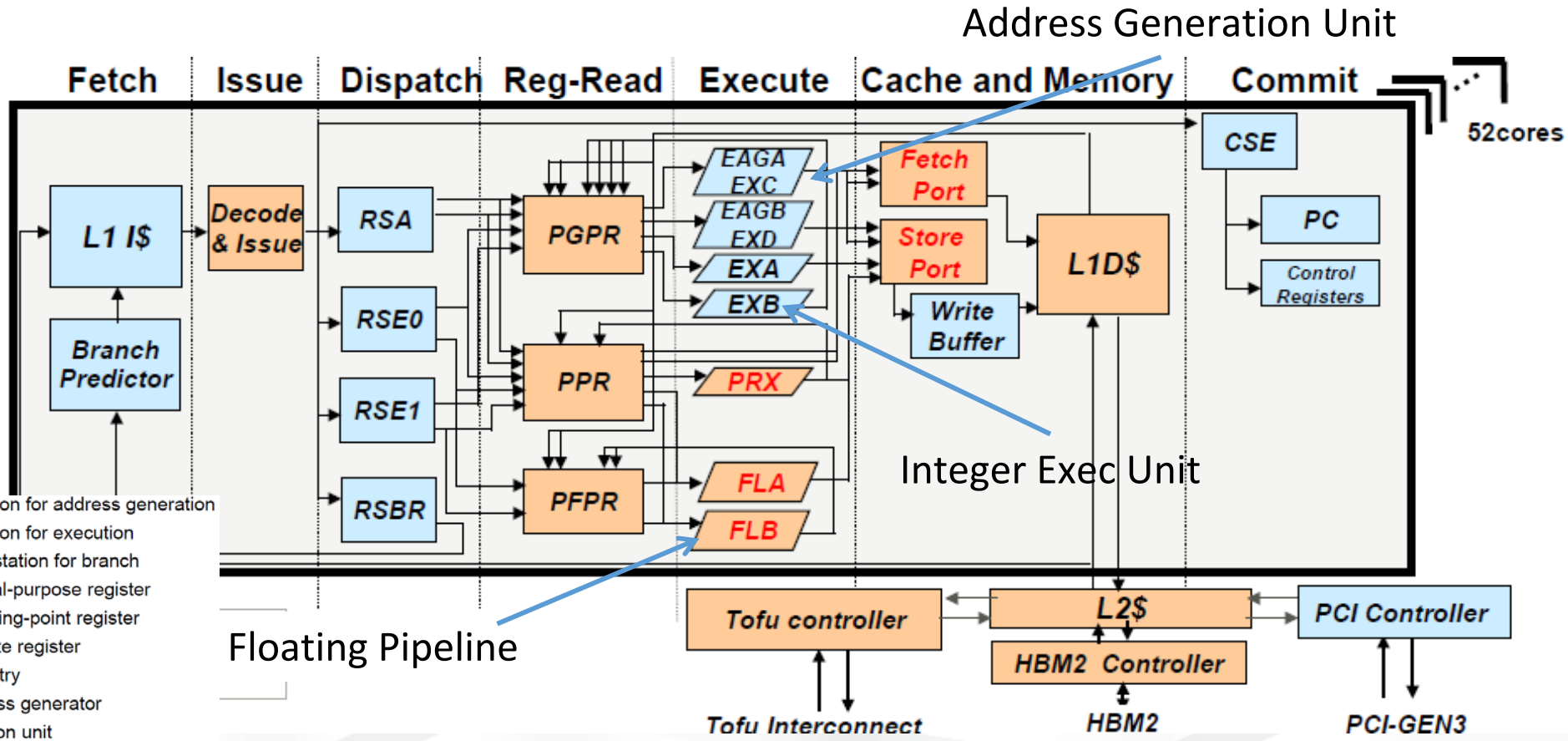
- CMG: 13 cores (12+1) and L2 cache (8MiB 16way) and memory controller for HBM2 (8GiB)
- X-bar connection in a CMG maximize efficiency for throughput of L2 (>115 GB/s for R, >57 GB/s for W)
- Assistant core is dedicated to run OS demon, I/O, etc
- 4 CMGs support cache coherency by ccNUMA with on-chip directory (> 115GB/s x 2 for inter-CMGs)



Figures from the slide presented in Hotchips 30 by Fujitsu

FX64A Core Pipeline

- **Superscalar Arch with out-of-order, branch prediction, inherited from Fujitsu SPARC**
- **L1D cache: 64 KiB, 4 ways, “Combined Gather” mechanism on L1**
- **SIMD and predicate operations**
  - 2x 512-bit wide SIMD FMA + Predicate Operation + 4x ALU (shared w/ 2x AGEN)
  - 2x 512-bit wide SIMD load or 512-bit wide SIMD store



Figures from the slide presented in Hotchips 30 by Fujitsu

- RSA: Reservation station for address generation
- RSE: Reservation station for execution
- RSBR: Reservation station for branch
- PGPR: Physical general-purpose register
- PFPR: Physical floating-point register
- PPR: Physical predicate register
- CSE: Commit stack entry
- EAG: Effective address generator
- EX : Integer execution unit
- FL : Floating-point execution unit
- PRX : Predicate execution unit

Tofu interconnect D

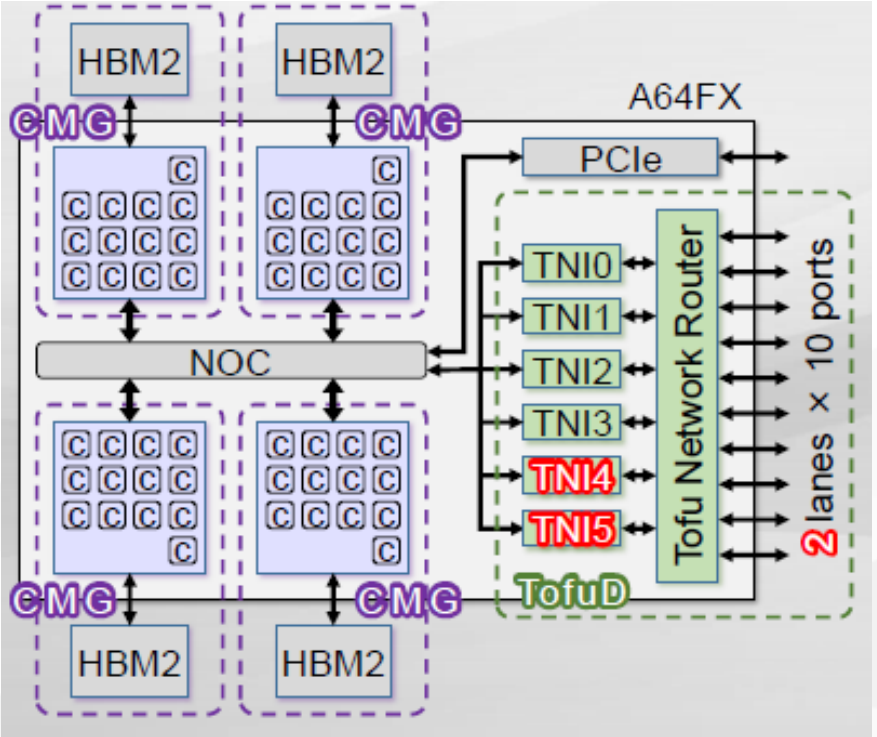
Presented in IEEE Cluster 2018
By Fujitsu

- **Direct network, 6-D Mesh/Torus**
- **28Gbps x 2 lanes x 10 ports (6.8GB/s / link)**
- **Network Interface on Chip**
  - 6 TNIs: Increased TNIs (Tofu Network Interface) achieves higher injection BW & flexible comm. Patterns
  - Memory bypassing achieves low latency

	TofuD spec
Data rate	28.05 Gbps
Link bandwidth	6.8 GB/s
Injection bandwidth	40.8 GB/s

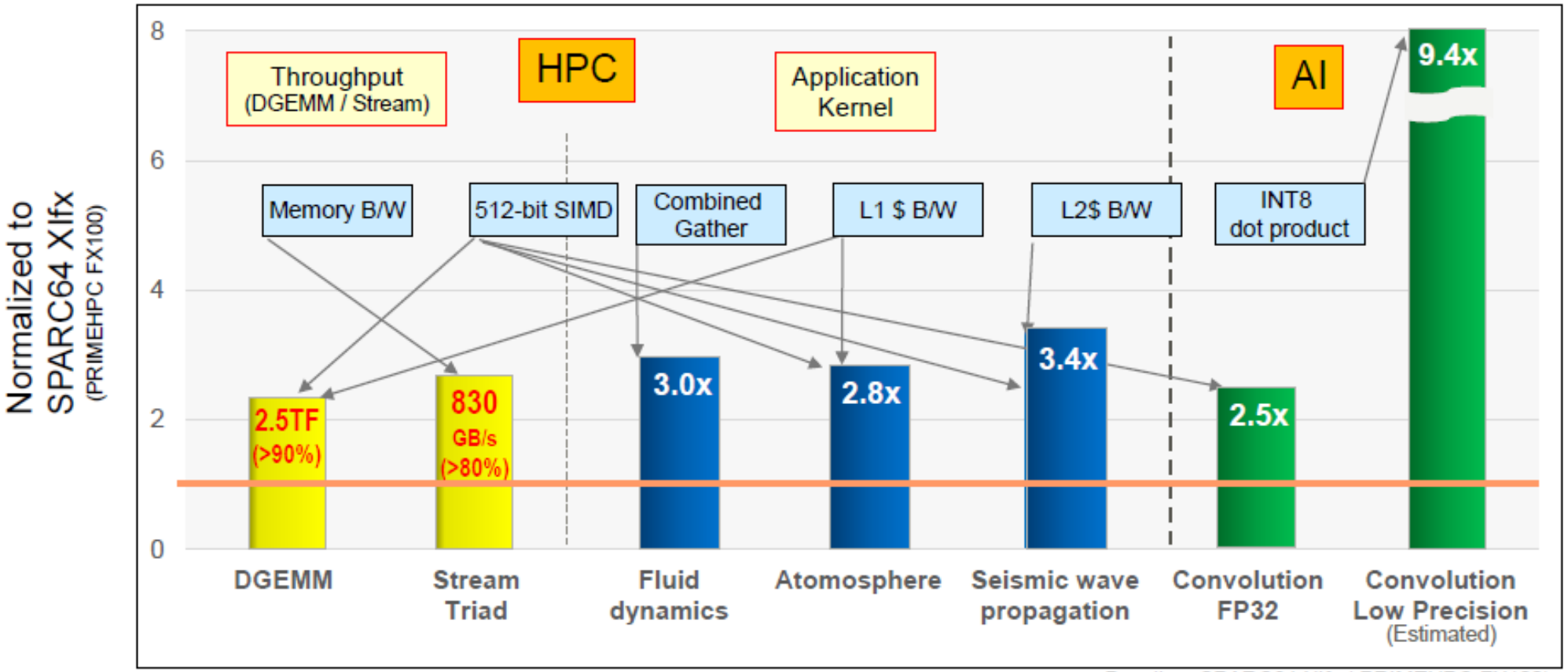
Ref) K computer: Link BW=5.0GB/s, #TNI=4

	Measured
Put throughput	6.35 GB/s
PingPong latency	0.49~0.54 μs



Preliminary Performance by “real silicon”

- The prototype CPU has been powered-on and preliminary performance evaluation by the prototype CPU has been done.
- Improvement by micro architectural enhancements, 512-bit wide SIMD, HBM2 and process technology
- The results are based on the Fujitsu compiler optimized for our microarchitecture and SVE
- AI apps will be supported by SVE FP16 instructions.



Figures from the slide presented in Hotchips 30 by Fujitsu

Low-power Design & Power Management

- **Leading-edge Si-technology (7nm FinFET)**
- **Low power logic design (15 GF/W @ dgemm)**

- **A64FX provides power management function called "Power Knob"**
  - FL pipeline usage: FLA only, EX pipeline usage : EXA only, Frequency reduction …
  - User program can change "Power Knob" for power optimization
  - "Energy monitor" facility enables chip-level power monitoring and detailed power analysis of applications

- **"Eco-mode" : FLA only with lower "stand-by" power for ALUs**
  - Reduce the power-consumption for memory intensive apps.
- **Retention mode: power state for de-activation of CPU with keeping network alive**
  - Large reduction of system power-consumption at idle time

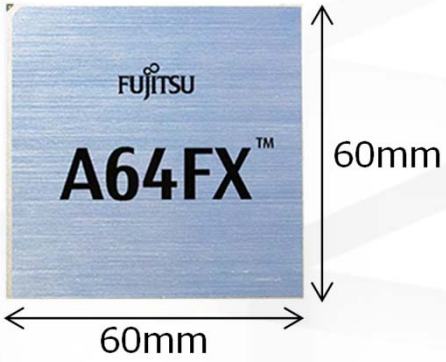
KPIs on post-K development in FLAGSHIP 2020 project

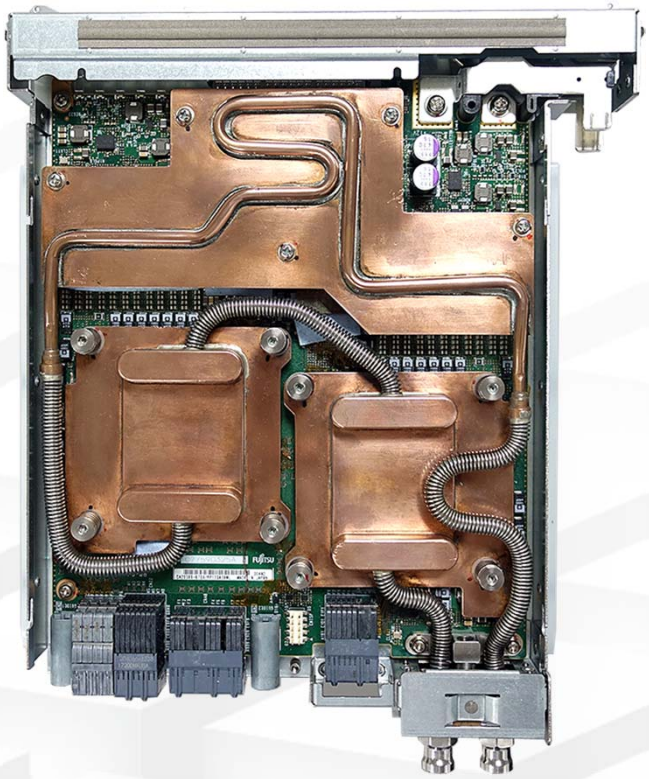
3 KPIs (key performance indicator) were defined for post-K development

- **1. Extreme Power-Efficient System**
  - Approx. 15 GF/W (dgemm) confirmed by the prototype CPU
  - Power consumption of 30 - 40MW (for system) is expected to be achieved
- **2. Effective performance of target applications**
  - It is expected to exceed 100 times higher than the K computer's performance in some applications
  - 106 times faster in GENESIS (MD application), 153 times faster in NICAM+LETKF (climate simulation and data assimilation) were estimated
- **3. Easy-of-use system for wide-range of users**
  - Shared memory system with high-bandwidth on-package memory must make existing OpenMP-MPI program ported easily.
  - No programming effort for accelerators such as GPUs is required.
  - Co-design with application developers

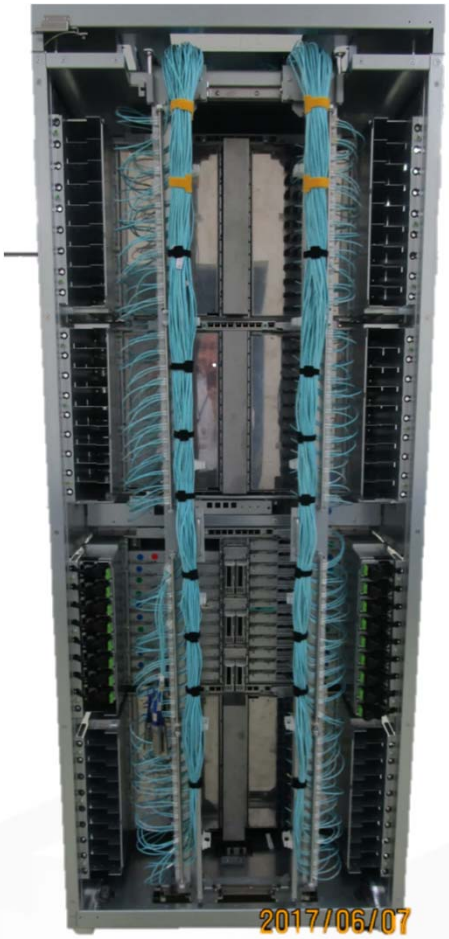
Post-K prototype board and rack

- **"Fujitsu Completes Post-K Supercomputer CPU Prototype, Begins Functionality Trials", HPCwire June 21, 2018**
  - "Fujitsu has now completed the prototype CPU chip that will serve as the core of post-K, commencing functionality field trials."





2 CPU / CMU



Shelf: 48 CPUs (24 CMU)
Rack: 8 shelves = 384 CPUs (8x48)

Advances from K computer

	K computer	Post-K	ratio	
# core	8	48		⇦ Si Tech
Si tech. (nm)	45	7		
Core perf. (GFLOPS)	16	56~	3.5	⇦ SVE
Chip(node) perf. (TFLOPS)	0.128	2.7~	21	⇦ CMG&Si Tech
Memory BW (GB/s)	64	1024		⇦ HBM
B/F (Bytes/FLOP)	0.5	0.4		
#node / rack	96	384	4	
Rack perf. (TFLOPS)	12.3	1036.8	84	
#node/system	82,944	???		
System perf.(PFLOPS)	10.6	???		

- SVE increases core performance
- Silicon tech. and scalable architecture (CMG) to increase node performance
- HBM enables high bandwidth

Global Competitiveness

- **Post-K has good power-performance as a “general-purpose” processor.**
- **In term with arithmetic performance and memory bandwidth, interconnect bandwidth, the post-K system is expected to be competitive to other world-class HPC systems.**

	Peak Flops (double precision) TFlops	Memory bandwidth (STREAM triad) GB/sec	fficiency in Linpack	Power-Performance GFlops/Watt	Interconnect Performance GB/sec
Post-K／A64fx	> 2.7	840	> 85 %	15.0	40.8
Oakforest-PACS／Xeon Phi KNL	3.0464	490	54.4 %	4.9	12.5 ※3
Niagara／Xeon Skylake※1	1.536	104.5	66.7 %	4.5	6.3 ※3
Summit／GPU Volta GV100※2	7.8	855	65.2 %	13.8	4.2 ※3
DGX-1 SaturnV Volta／GPU Tesla V100※2	7.8	855	58.8 %	15.1	6.3 ※3

※1 one socket performance estimated by open information on two-socket performance of Skylake（Xeon Gold 6148 20C 2.4GHz)
※2 Peak performance of one socket connected with NVLINK. Memory bandwidth by one socket GPU.
※3 Network controller is not integrated on chip. Attached Infiniband network of 100Gbps（12.5GB/sec)
For Niagara, one 100Gbps Infiniband for two sockets. For Summit, two 100Gbps Infiniband for 6 sockets.
For DGX-1 SaturnV Volta, four 100Gbps Infiniband for 8 sockets GPU. For all systems, network performance indicated for one socket..

Data taken from the report presented at CSTI, 2018

“PostK” performance evaluation environment

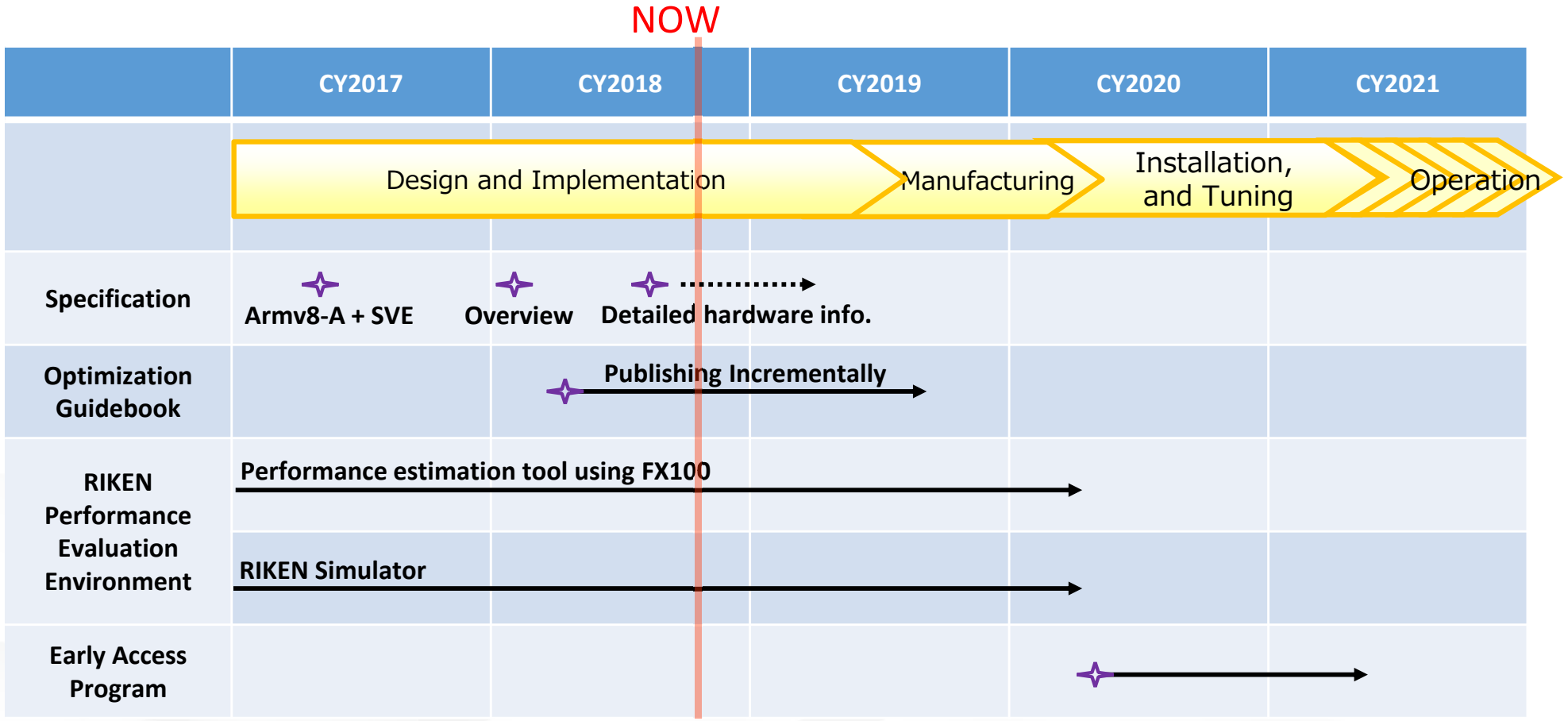
- **RIKEN is constructing “PostK” performance evaluation environment for application programmers to evaluate and estimate the performance of their applications on “PostK” and for performance turning for “postK”.**
- **The “PostK” performance evaluation environment is available on the servers installed in RIKEN. The environment includes the following tools and servers:**
  - A small-scale FX100 system and “postK” performance estimation tool:

    The estimation tool gives the performance estimation of multithreaded programs on “postK” from the profile data taken on FX100.
  - “PostK” processor simulator based on GEM-5:

    “PostK” processor simulator will give a detail performance results including estimated executing time, cache-miss, the number of instruction executed in O3. The user can understand how the compiled code for SVE is executed on “postK” processor for optimization. (Arm released GEM-5 beta0 of SVE)

    FP16 SVE will be available soon.
  - Compilers for “PostK” processor
    - Fujitsu Compilers : Fortran, C, C++. Fully-tuning for “postK” architecture.
    - Arm Compiler : LLVM-based compiler to generate code forArmv8-A + SV. C,C++ by Clang, Fortran by Flang
  - SVE emulator on Arm server, developed by Arm for fast SVE code execution.
  - Arm Severs (Planned 4Q/2018)

Schedule on Development and Porting Support



- CY2018. Q2, Optimization guidebook is incrementally published
- CY2020. Q2, Early access program starts
- CY2021. Q1/Q2, General operation starts

Note: Fujitsu will reveal features of Post-K CPU at Hot Chips 2018.

- Takeo Yoshida, "Fujitsu's HPC processor for the Post-K computer," IEEE Hot Chips: A Symposium on High Performance Chips, San Jose, August 21, 2018.

Post-K CPU New Innovations: Summary

1. **Ultra high bandwidth using on-package memory & matching CPU core**
   - Recent studies show that majority of apps are memory bound, some compute bound but can use lower precision e.g. FP16
   - Comparison w/mainstream CPU: much faster FPU, almost order magnitude faster memory BW, and ultra high performance accordingly
   - Memory controller to sustain massive on package memory (OPM) BW: difficult for coherent memory CPU, first CPU in the world to support OPM
2. **Very Green e.g. extreme power efficiency**
   - Power optimized design, clock gating & power knob, efficient cooling
   - Power efficiency much better than CPUs, comparable to GPU systems
3. **Arm Global Ecosystem & SVE contribution**
   - Annual processor production: x86 3-400mil, ARM 21bil, (2~3 bil high end)
   - Rapid upbringing HPC&IDC Ecosystem (e.g. Cavium, HPE, Sandia, Bristol,…)
   - SVE(Scalable Vector Extension) -> Arm-Fujitsu co-design, future global std.
4. **High Performance on Society5.0 apps including AI**
   - Next gen AI/ML requires massive speedup => high perf chips + HPC massive scalability across chips
   - Post-K processor: support for AI/ML acceleration e.g. Int8/FP16+fast memory for GPU-class convolution, fast interconnect for massive scaling
   - Top performance in AI as well as other Society 5.0 apps