計算宇宙物理学
～計算科学と高性能計算技術の理想的なランデブー～
（計算機科学屋の視点から）

筑波大学計算科学研究センター・センター長
朴 泰祐
taisuke@ccs.tsukuba.ac.jp

自己紹介（朴泰祐・筑波大学）

- 1992年2月より筑波大学電子・情報工学系講師→（現在）筑波大学計算科学研究センター(CCS: Center for Computational Sciences)教授/センター長（2019～）
- 大学学部時代から一貫して並列処理システムのアーキテクチャ，システムソフトウェア，ミドルウェア，コンパイラ，アプリケーションというほぼ全階層に跨る研究を行ってきた
- 1992年からCCS（当時はCCP: Center for Computational Physics）では文科省概算要求予算により「新プログラム」により，素粒子物理学を中心とした大規模計算物理学研究を推進する超並列計算機開発を実施，その中で特に並列処理ネットワーク等の開発を担当
  →CP-PACS, 1996/11 TOP500世界第一位
- 以後，CCSにおける全てのスーパーコンピュータの研究開発と調達・導入のシステム側責任者を務める
  →PACS-CS, FIRST, HA-PACS, HA-PACS/TCA, COMA, Cygnus (, Pegasus)
- 単にMPPやクラスタを購入するだけでなく，独自の工夫を盛り込むユニークなシステム開発を行ってきた
- CCSでは超並列システムとそれを用いるアプリケーションの開発を一体化して一つのセンター内で30年に渡り進めてきた
  → codesignの走り→ Application-oriented System
- 「京」「富岳」の開発ではアーキテクチャWGの一員として参画
  →「富岳」公開時にはHPCIコンソーシアム理事長として運用に貢献
- 現在，文科省次世代計算機システム（「富岳Next」）のFeasibility Studyの Program Director の一人としてFSプログラムの運営に参加
- 国内スーパーコンピュータの諸々の運営業務：「富岳」成果創出加速課題プログラム領域総括，HPCI計画推進委員会委員
- 国際会議運営：HPC Asia Steering Committee Chair，ICPP SC, Cluster SC等，2023年ACM Gordon Bell Prize Committee Chair

牧野さんとの関係（=計算宇宙物理学との出会い）

- 出会いは33年前（1990年？）
  - 杉本大一郎先生がGRAPE計画の相談に，当時朴が川合敏雄先生の下で助手をやっていた慶應義塾大学理工学部物理学科を訪れた
  - 「ハードウェアのことをほとんど知らないので」，どうやってボードを組むとか部品を選ぶとか，計算機科学としては比較的初歩的な技術の相談
    （ラッピングのやり方とか。。）
  - 一緒に訪問されたのが牧野さん（当時博士課程学生）と伊藤さんで，ハードウェアの組み方について議論
    - GRAPEボードをホストPCから制御するのにGPIBを使う技術
    - GRAPEボードの回路図のチェック
    - データを一時保存するregisterに，FlipFlopではなくShift Registerを使っていたので驚いた（牧野さん曰く「多数のビットを保存するのにこちらの方が合理的」）
    - GRAPE-1お披露目会で完成したボードを見せて頂いた

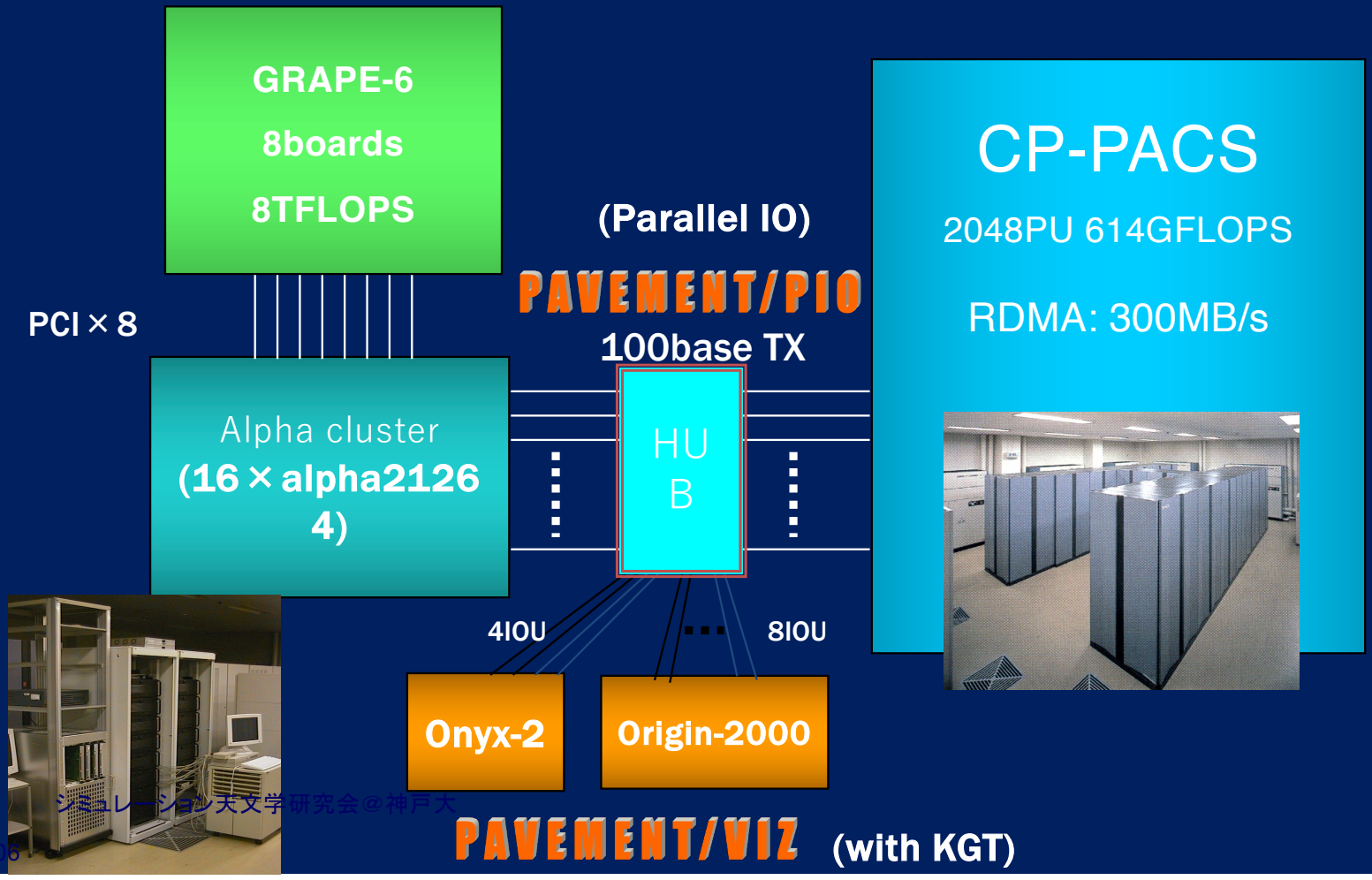
杉本先生の本


BLUE BACKS
手作りスーパーコンピュータ
への挑戦
テラ・フロップス・マシンをめざして
杉本大一郎
Sugimoto Daiichirô
DO IT
YOURSELF
SUPER
COMPUTER

再会

- 2000年代に入り，CCSで文科省概算要求「未来開拓」プロジェクトが採択され，再びApplication-oriented System のアーキテクチャとアプリケーション開発を実施
- HMCS (Heterogeneous Multi-Computer System)を提案
  - 超並列システムに演算加速装置を結合し，CPUでの一般的な処理と演算加速装置の高速性を組み合わせ，複合問題を処理する
  - ターゲットに宇宙物理を選び（アプリケーション側は梅村雅之先生），GRAPE-6ボードを導入して宇宙初期天体シミュレーション，輻射流体問題をターゲットに
  - 32個のFPGA-GRAPEチップを載せたボードを8枚，DEC Alpha CPUによるクラスタを介してCP-PACSと100base Ethernetで結合
  - システム開発過程で牧野さんと何度も相談し，ボードのメンテナンスやライブラリソフトの実装等のノウハウを教わる
- HMCSの実装とアプリケーション実行の論文が2002年情報処理学会論文賞を受賞
  - 「Heterogeneous Multi-Computer Systemにおける重力効果を含む宇宙輻射流体計算」（Vol.43,No.SIG6(HPS5)）、朴　泰祐、牧野　淳一郎、須佐　元、梅村　雅之、福重　俊幸、宇川　彰

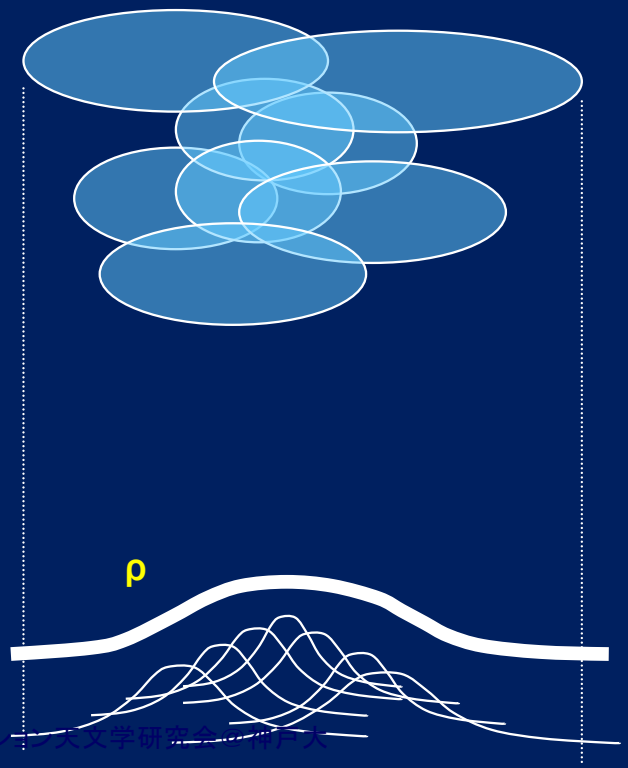
Prototype HMCS SYSTEM



g6cpplib

- `g6cpp_start(myid, nio, mode, error)`
- `g6cpp_unit(n, t_unit, x_unit, eps2, error)`
- `g6cpp_calc(mass, r, f_old, phi_old, error)`
- `g6cpp_wait(acc, pot, error)`
- `g6cpp_end(error)`

SPH
(Smoothed Particle Hydrodynamics)



Matter density represented as a collection of particles

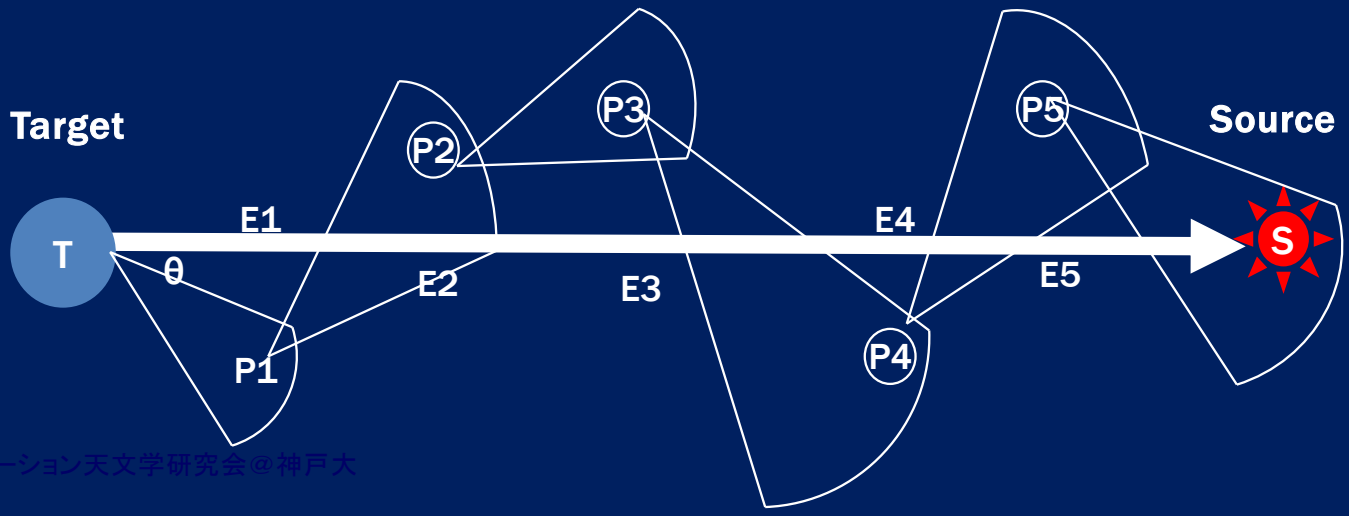
$$\rho(\mathbf{r}_i) = \sum_j \rho_{j0} W(|\mathbf{r}_i - \mathbf{r}_j|)$$

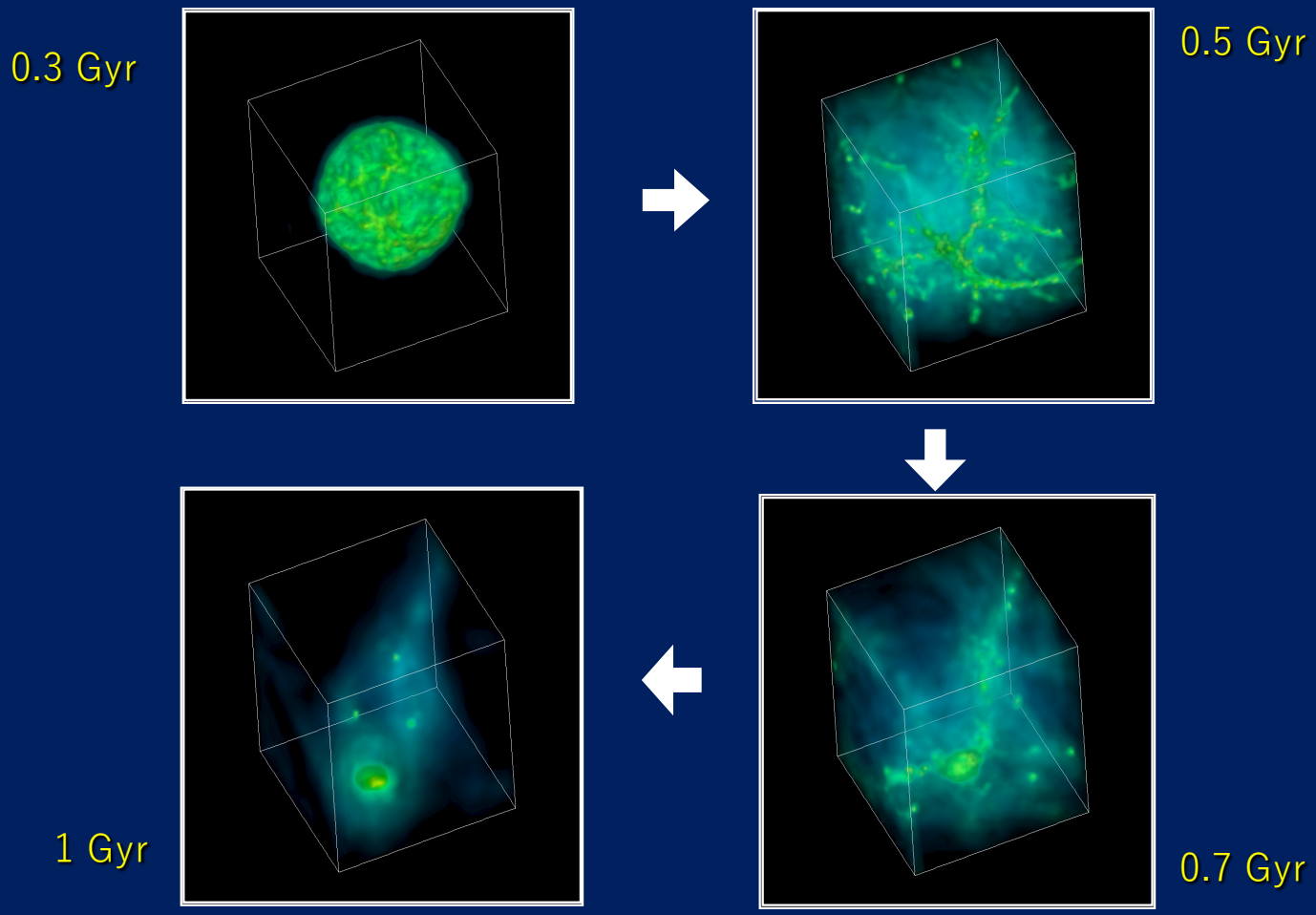
W : kernel function

Radiative Transfer for SPH

- Accurate calculation of optical depth along light paths required.
- Use the method by Kessel-Deynet & Burkert (2000) .

$$\tau_{TS} = \sum_i \frac{\sigma}{2}\left(n_{E_i} + n_{E_{i+1}}\right)\left(s_{E_{i+1}} - s_{E_i}\right)$$



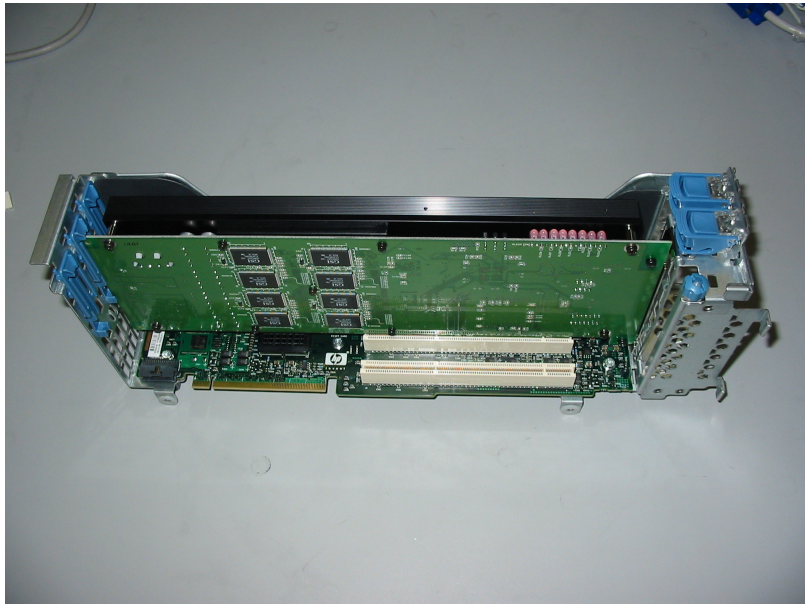


At the initial stage, a density fluctuation is generated to match a cold dark matter spectrum. This fluctuation expands with the cosmic expansion, and simultaneously smaller-scale density fluctuations develop inside to form filamentary structures. Tiny filaments evaporate due to the heating by background UV radiation, whereas larger filaments shrink to coalesce into a condensed rotating cloud. This rotating cloud would evolve into a galaxy. This simulation has revealed that the background UV radiation plays an important role for the final structure of the galaxy.

HMCSからの発展(1)

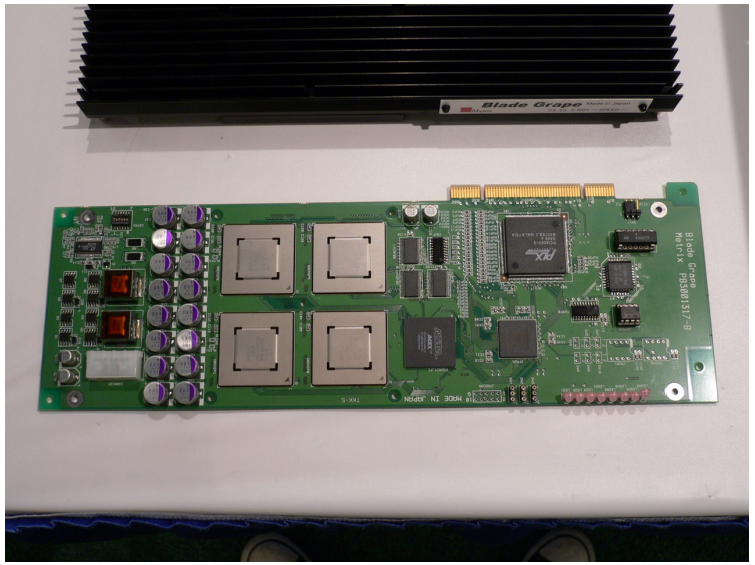
- **HMCS-E (HMCS with Embedded system)**
  - 超並列クラスタの各計算ノードに小型化したGRAPEボードをPCIで搭載
  - GRAPE-6ベースのチップを搭載したBlade-GRAPEを開発しPCI-Xフォームファクタとして標準のラックマウントサーバに挿せるようにした
  - 256ノード分作成→FIRST（2004～2007）
  - 科研特別推進研究で構築（「融合型並列計算機による宇宙第一世代天体の起源の解明」，PI:梅村雅之）
    - Blade-GRAPEによりピーク性能40TFLOPS
      →当時「地球シミュレータの1/100以下の予算で同等のスピードを実現」と宣伝

BladeGRAPE: Compact engine with GRAPE-6

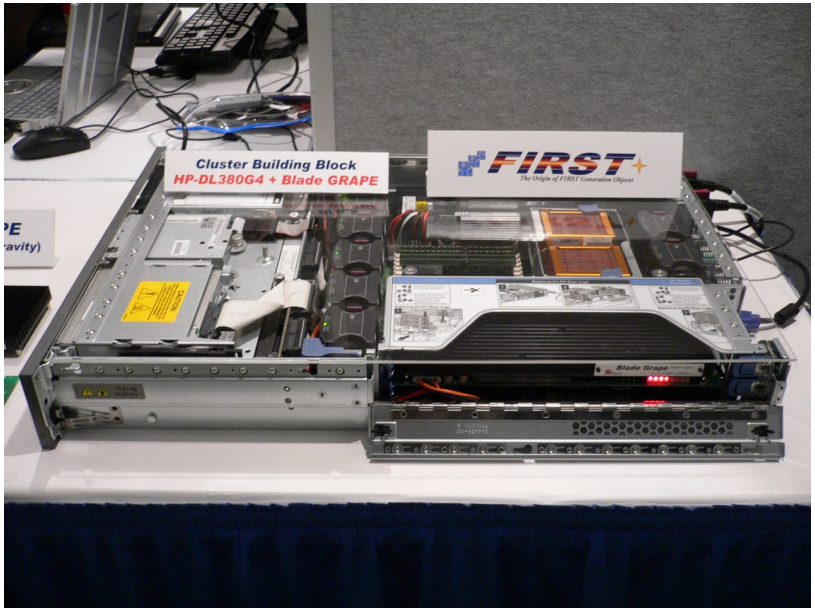


BladeGRAPE on PCI rack
ready to built on 2U PC server

4 GRAPE-6 core chips on PCI-X board



FIRST full system



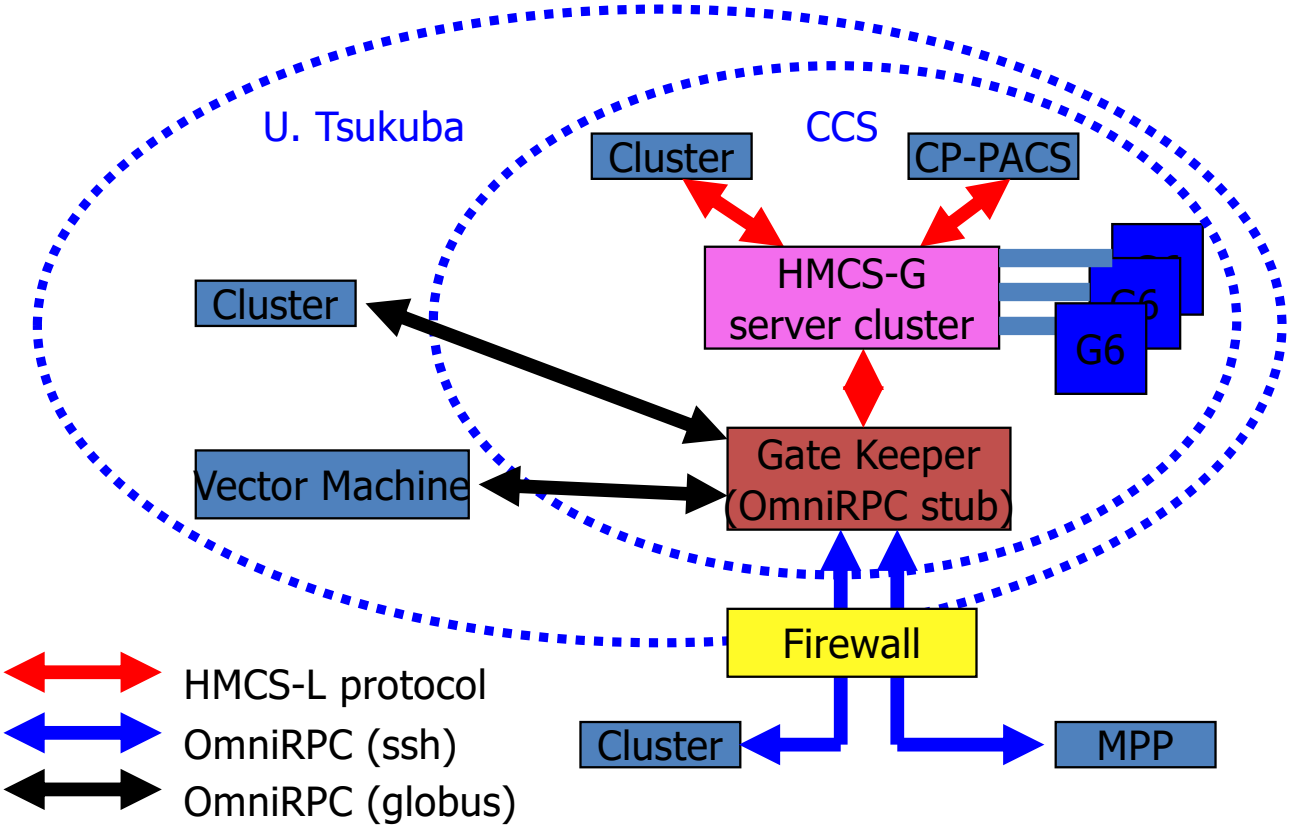
A node with 2U high-end PC server with BladeGRAPE
（SCで実物を出展，ホットデモ）

256 node system installed at CCS, U. Tsukuba

HMCSからの発展(2)

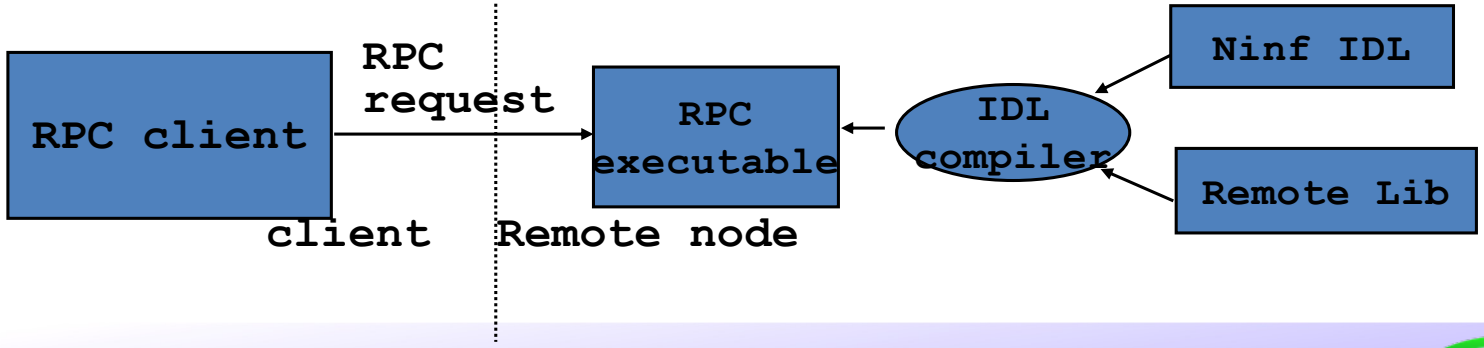
- HMCS-EはHMCSで超並列システムとGRAPEをより近接に「組み込み」型で実装した
- 逆に，リモートのGRAPEサーバを世界中から使えるようしてリソースシェアしてはどうか？
  → RPC (Remote Procedure Call)でGRAPEサーバへのアクセスを実現
  → HMCS-G (Grid)
- HMCSプロトタイプで使ったAlpha Cluster上にweb serviceとRPCを実装し，インターネット経由で使えるようにした
  →プロトタイプ開発のみ

HMCS-G block diagram



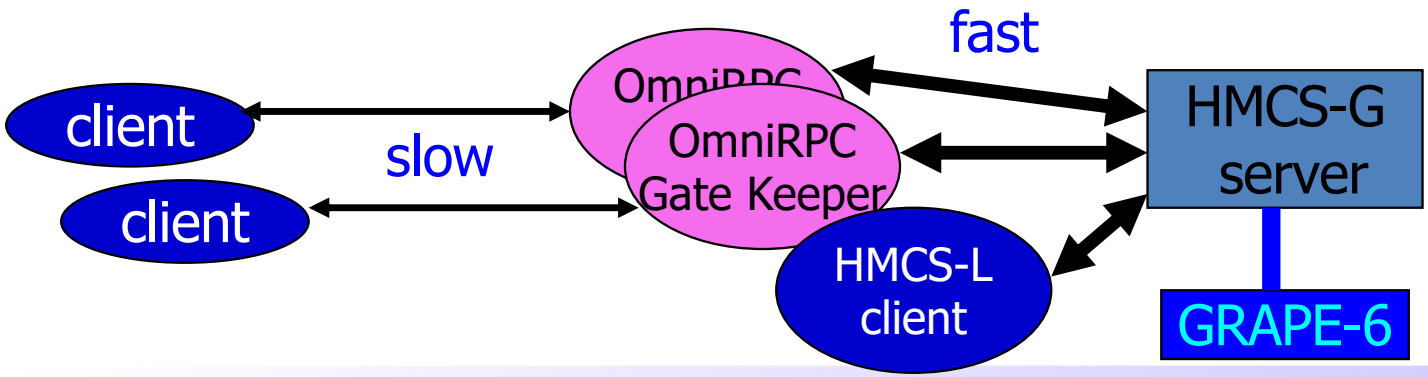
OmniRPC （RWCP & 筑波大）

- Ninf grid RPCに基づくスレッドセーフなRPC
  - **RPC実行可能プログラム (library programs)**
    - RPCのためのstubを被せたライブラリプログラム
    - Ninf IDLより生成
    - RPCリクエストにより起動
  - **プログラミングインタフェース**
    - 言語非依存な単純なインタフェース： `OmniRpcCall`
    - C, Fortranのような既存言語から簡単に利用可能



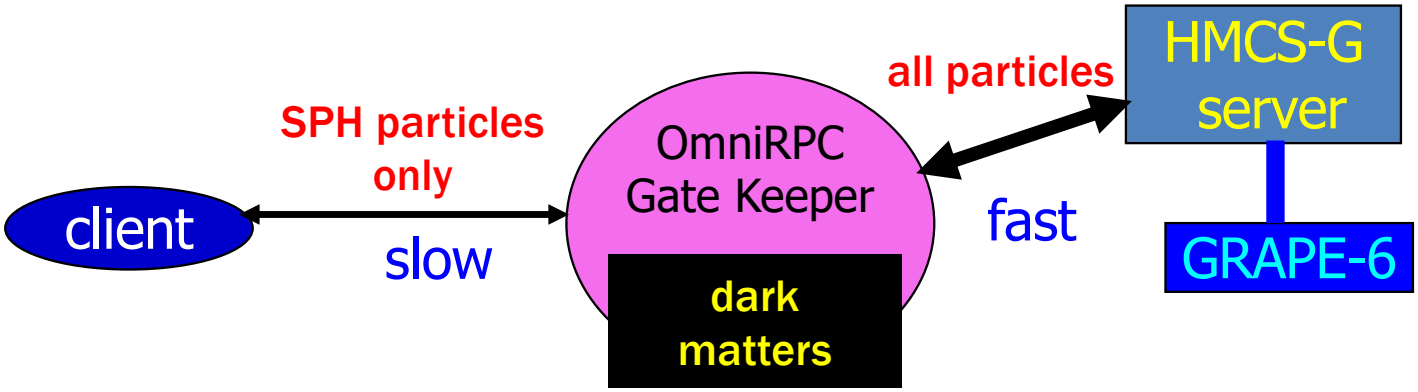
Gate Keeper (OmniRPC stub)

- ユーザ認証とアカウンティング
- アプリケーションユーザフレンドリー
  - 簡単な API
  - globus オプション（Grid環境標準）
  - ssh オプション（インストールが簡単）
- データバッファリングによる通信スピードギャップの緩衝
- 元からあるHMCS-Lクライアントとの共存



ダークマターの局所化

- RT-SPH シミュレーションでは半分あるいはそれ以上の粒子がダークマター
- ダークマターは汎用計算機側でSPH粒子と共に処理される必要がない（通常粒子との反応がなく、自己重力にのみ影響する）
- Gate Keeperはダークマターのデータを保存しておき、汎用計算機側とのやりとりを行わない
  ⇒ OmniRPCが提供するデータ永続性機能を利用
  ⇒ データ通信の量が大幅に削減される



計算宇宙物理学との出会いを通じて

- **Application-oriented System**の有効性の確認
- 課題
  - どんなアプリケーションでも専用システムがうまく作れるわけではない
  - 計算の骨格となる共通するアルゴリズム（ex: 重力計算）が存在し，かつその処理が汎用**CPU**では極めて重いケースでないとバランスしない
  - 演算・メモリ・通信のバランスがうまく取れないと解がない
- 宇宙物理は専用性の高い高性能計算システムとの親和性が非常に高いので**GRAPE**は成功した（と思っている）
- その後，**QCD**や気象等の分野とも**codesign**を進めたが，宇宙物理をターゲットにしたものが最も対費用効果が高く現実的なものとなった
  → 筑波大**CCS**では**FIRST**が成功例

より汎用なプラットフォームを目指して

- GRAPEは用途とアルゴリズムが極めて明確で，かつ宇宙物理での汎用性に長けている
- 他の分野の計算をHMCS的に行うには？→演算加速装置が鍵
  - GPUはHPCの花形演算加速装置
  - しかし万能ではない
    - 大規模かつ均質な空間並列性（超SIMD, 超STMD）が必要
    - 条件分岐に弱い
    - 通信はホスト任せなので頻繁なノード間通信があるとkernelとホストを行ったり来たりしないといけない→通信オーバヘッド大
- より自律的な演算加速装置は？
  - FPGA (Field Programmable Gate Array)

FPGA in HPC

- Goodness of recent FPGA for HPC
  - True codesigning with applications (essential)
  - Programmability improvement: OpenCL, other high level languages
  - High performance interconnect: 100Gbps x 4
  - Precision control is possible
  - Relatively low power
- Problems
  - Programmability: OpenCL is not enough, not efficient
  - Low standard FLOPS: still cannot catch up to GPU
    -> “never try what GPU works well on”
  - Memory bandwidth: 1-gen older than high-end CPU/GPU
    -> be improved by HBM, but still difficult to use

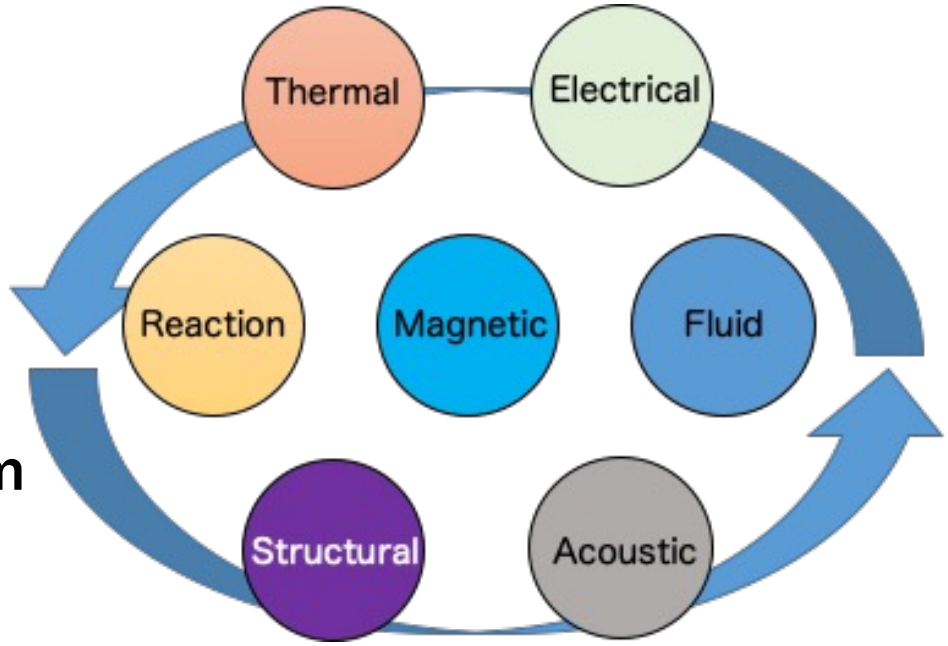
BittWare 520N with Intel Stratix10 FPGA equipped with 4x 100Gbps optical interconnection interfaces

GPU vs FPGA as HPC solutions

device	GPU	FPGA
parallelization	SIMD (x multi-group)	pipeline (x multi-group)
standard FLOPS	😀😀 (>1000 cores)	😀 (~100 pipeline)
conditional branch	😢 (warp divergence)	😀 (both direction)
memory	😀😀 (HBM2e)	😢 (DDR)⟶😀 (HBM2)
interconnect	😢 (via host facility)	😀😀 (own optical links)
programming	😀 (CUDA, OpenACC, OpenMP)	😢 (HDL)⟶😏 (HLS)
self-controllability	😢 (slave device of host CPU)	😀 (autonomous)
HPC applications	😀 (various fields)	😢 (limited)

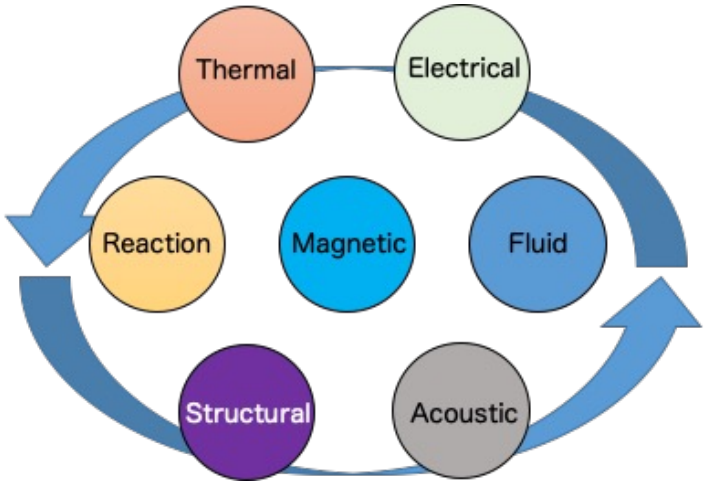
Coupling GPU and FPGA, why ?

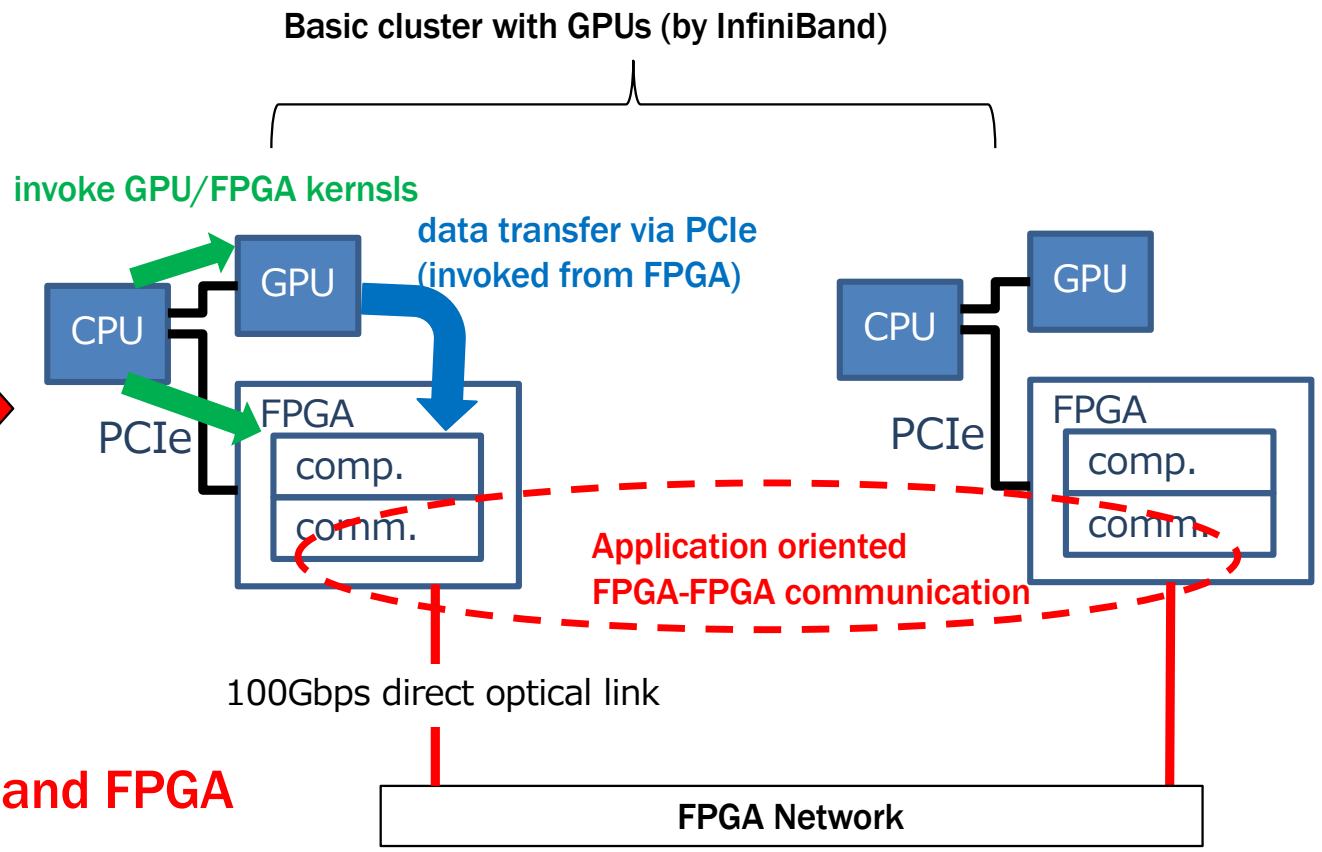
- Many multi-physical simulation to combine several sorts of different phenomena on a system is required in advanced physics
  - space – particle reaction
  - fluid dynamics with chemical reaction
  - macroscopic/microscopic hybrid simulation molecular simulation
- Characteristics and dynamism of parallelism drastically changes during the simulation
  - SIMD friendly or pipeline friendly
  - small fraction of low degree parallelism in a code makes “last one mile problem” under Amdahl‘s Law



CHARM: Cooperative Heterogeneous Acceleration with Reconfigurable Multi-devices

multi-physics/multi-scale complicated problem

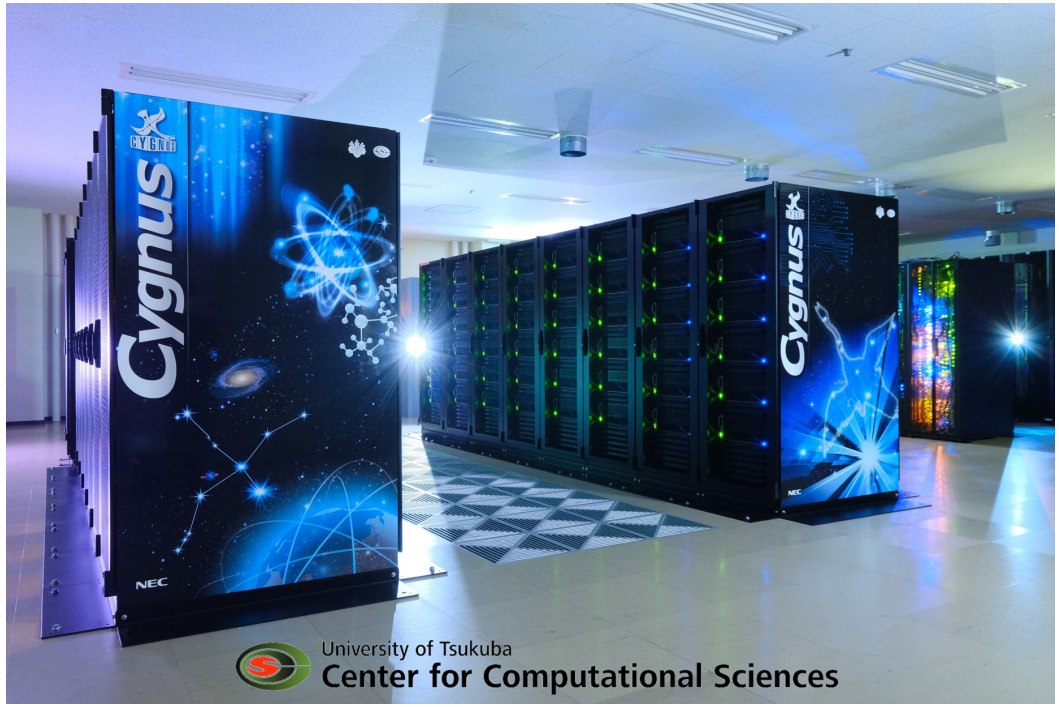




Cooperative computing with GPU and FPGA

Cygnus: world first multi-hybrid cluster with GPU+FPGA

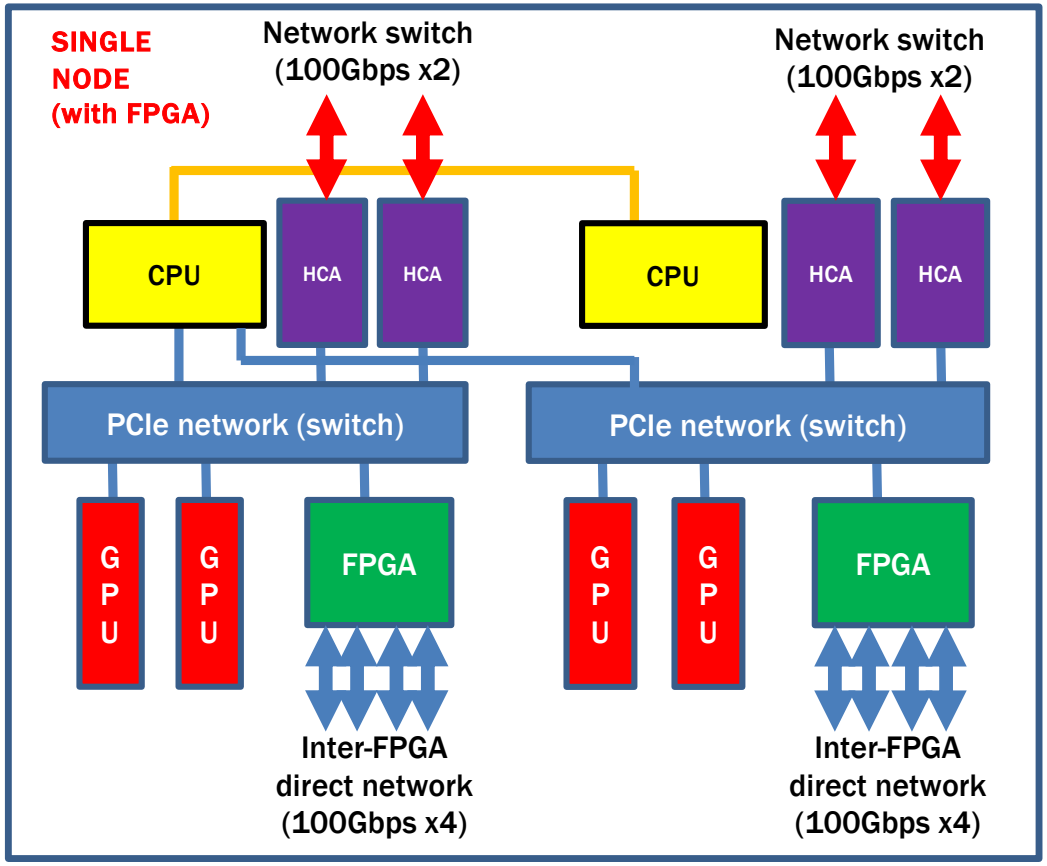
@ CCS, Univ. of Tsukuba (deployed by NEC)



Cygnus supercomputer at Center for Computational Sciences, Univ. of Tsukuba (Apr. 2019~)
85 nodes in total including 32 “Albireo” nodes with GPU+FPGA (other “Deneb” nodes have GPU only)

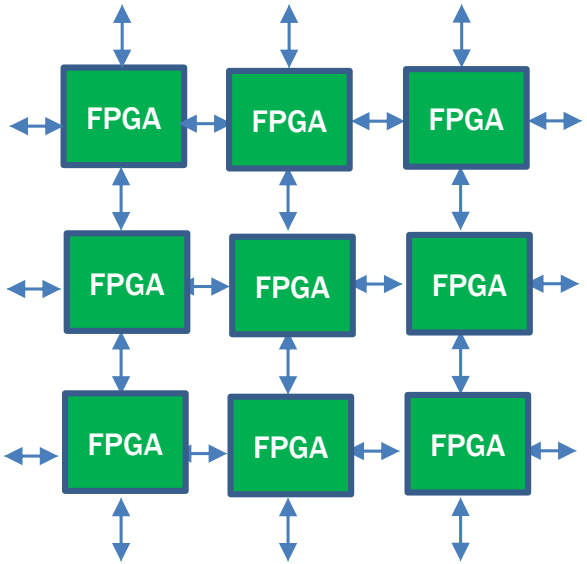
Single node configuration (Albireo)

- Each node is equipped with both IB EDR and FPGA-direct network
- Some nodes are equipped with both FPGAs and GPUs, and other nodes are with GPUs only



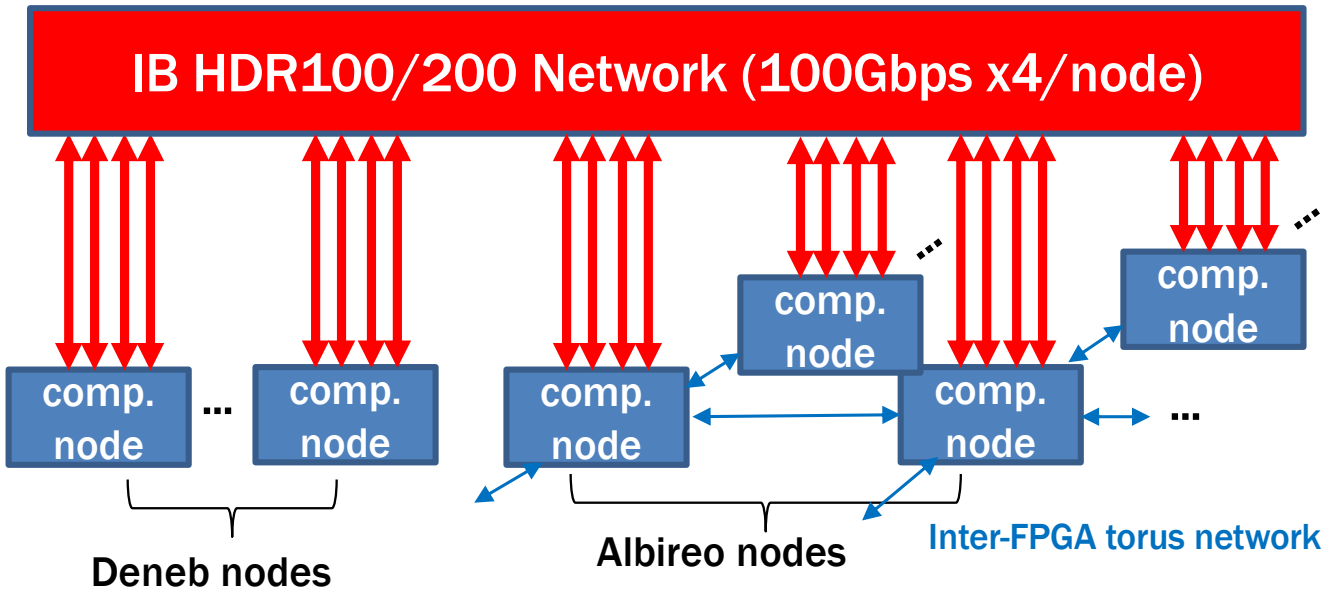
Two types of interconnection network

Inter-FPGA direct network (only for Albireo nodes)

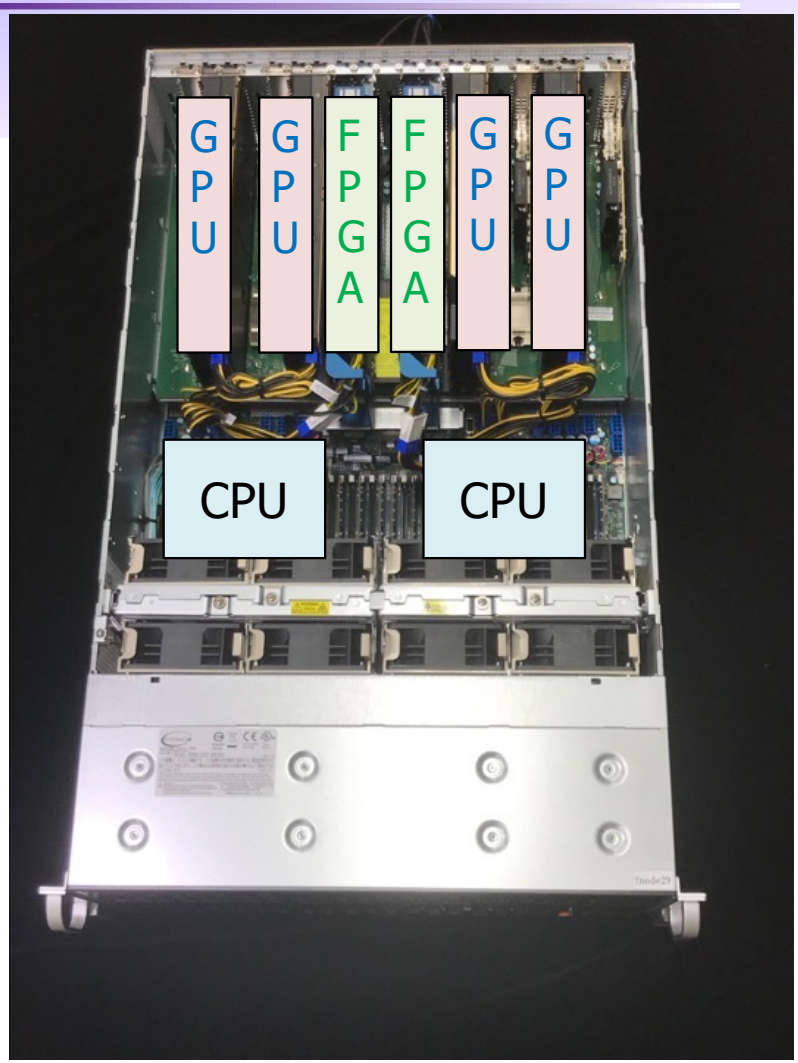


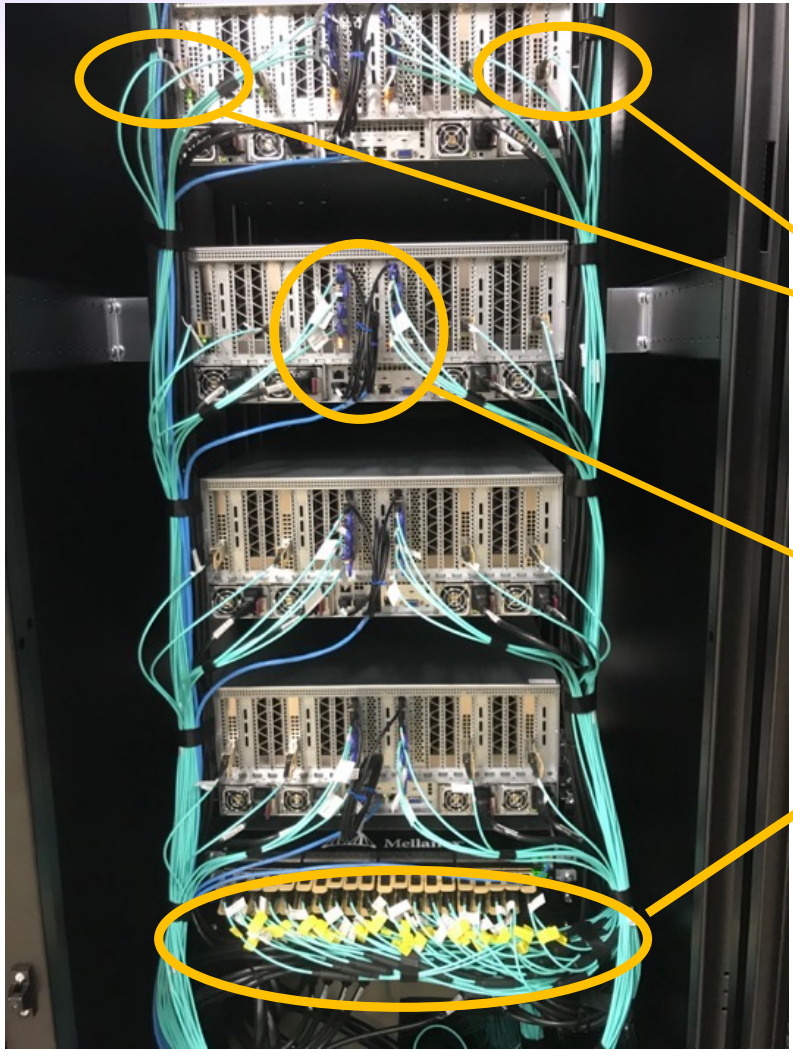
64 of FPGAs on Albireo nodes (2FPGAS/node) are connected by 8x8 2D torus network without switch

InfiniBand HDR100/200 network for parallel processing communication and shared file system access from all nodes



For all computation nodes (Albireo and Deneb) are connected by full-bisection Fat Tree network with 4 channels of InfiniBand HDR100 (combined to HDR200 switch) for parallel processing communication such as MPI, and also used to access to Lustre shared file system.





IB HDR100 x4
⇨ HDR200 x2

100Gbps x4
FPGA optical
network x2

IB HDR200
switch (for
full-bisection
Fat-Tree)

1.2Tbps/node

Albireo node

Application Example – ARGOT code

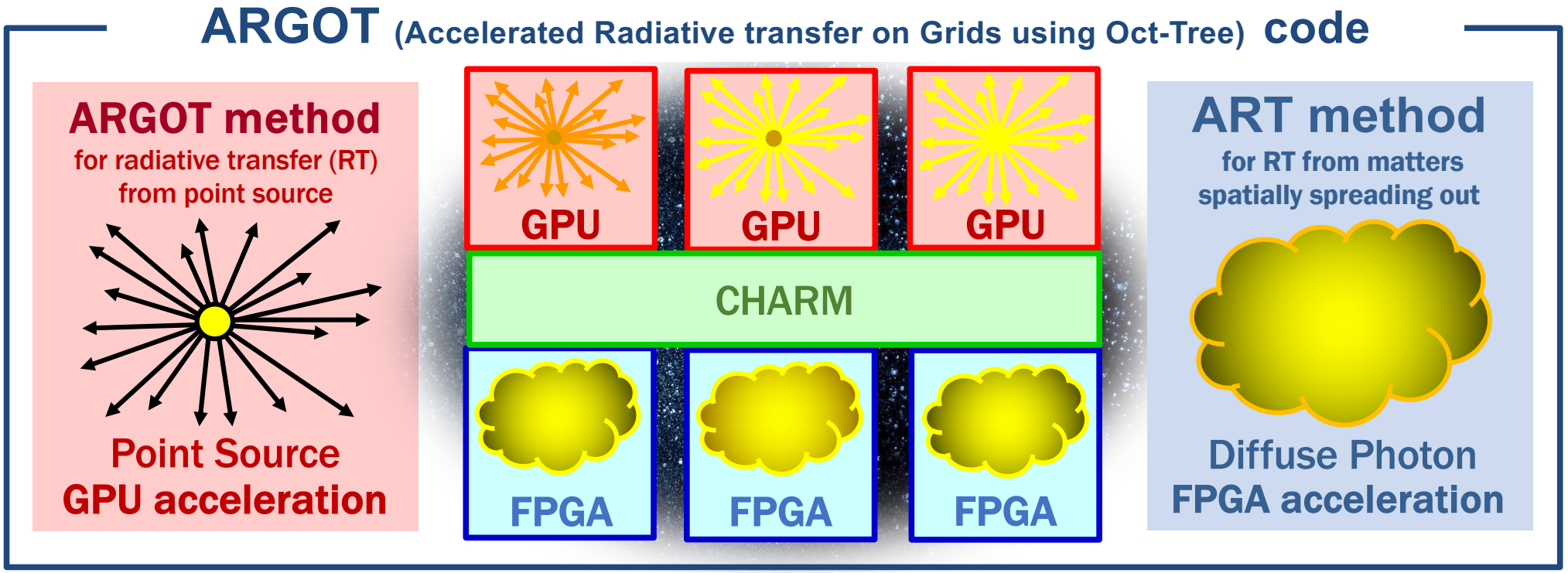
- ARGOT (Accelerated Radiative transfer on Grids using Oct-Tree)
  - Simulator for early stage universe where the first stars and galaxies were born
  - Radiative transfer code developed in Center for Computational Sciences (CCS), University of Tsukuba
  - CPU (OpenMP) and GPU (CUDA) implementations are available
  - Inter-node parallelisms is also supported using MPI
- ART (Authentic Radiation Transfer) method
  - It solves radiative transfer from light source spreading out in the space
  - Dominant computation part (90%~) of the ARGOT code

Two computation elements in ARGOT code: ARGOT method and ART method

- ARGOT method: Point Source processing
- ART method (Authentic Radiation Transfer): Diffused Photon processing

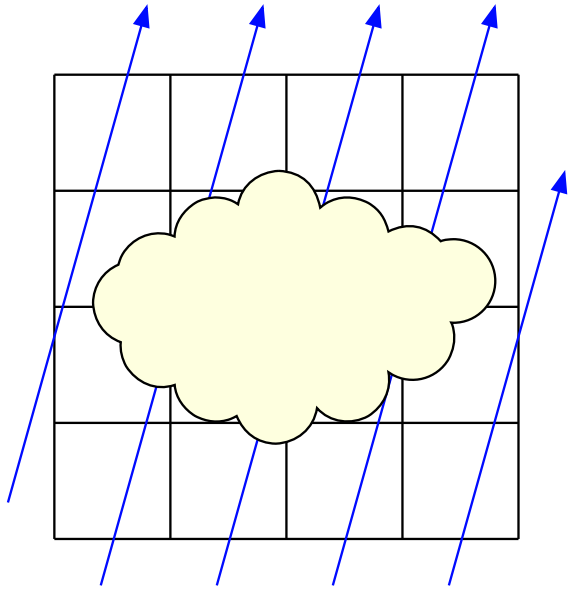
Two computation elements in ARGOT code: ARGOT method and ART method

- ARGOT method: Point Source processing
- ART method (Authentic Radiation Transfer): Diffused Photon processing



ART Method

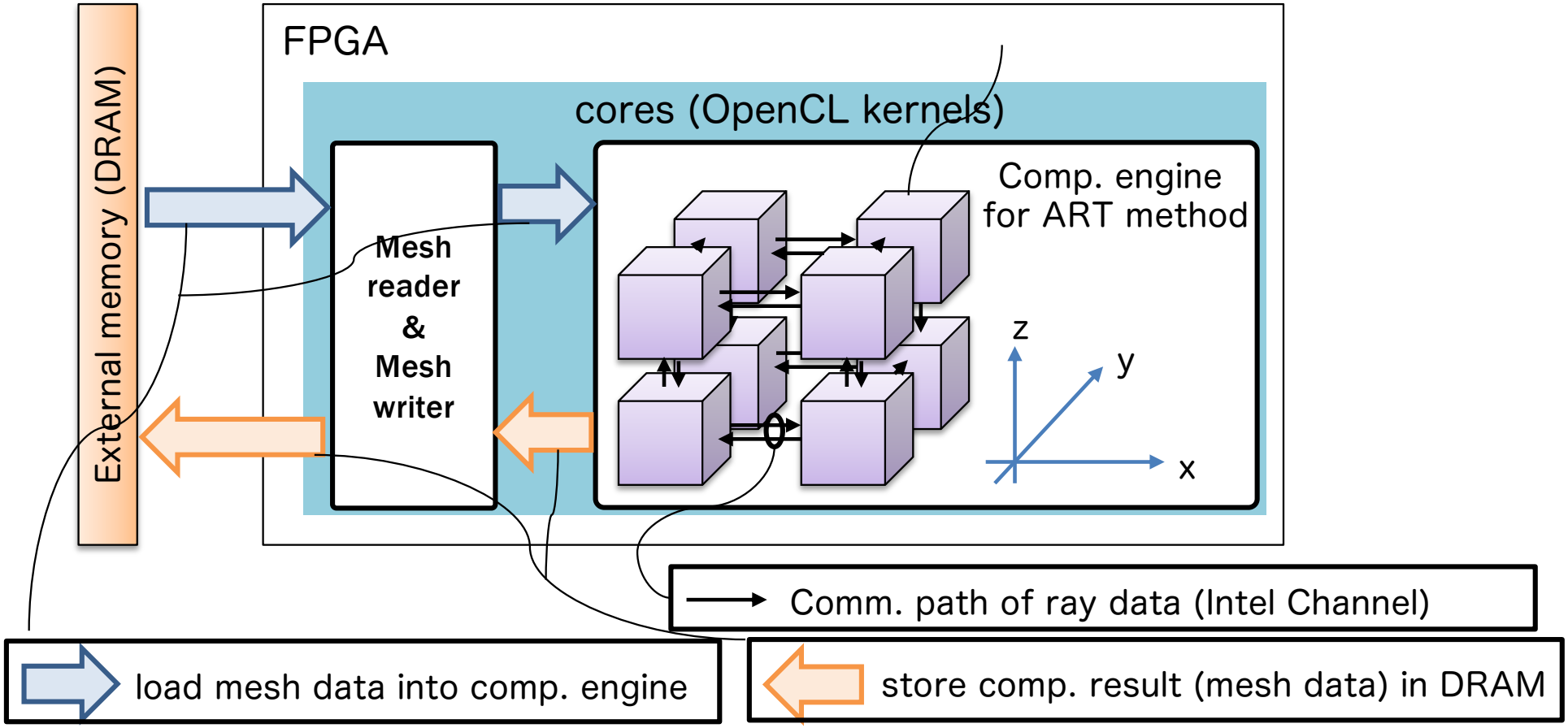
- ART method is based on ray-tracing method
  - 3D target space split into 3D meshes
  - Rays come from boundaries and move in straight in parallel with each other
  - Directions (angles) are given by HEALPix algorithm
- ART method computes radiative intensity on each mesh as shows as formula (1)
  - Bottleneck of this kernel is the exponential function (expf)
  - There is one expf call per frequency (v). Number of frequency is from 1 to 6 at maximum, depending on the target problem
  - All computation uses single precision computations
- Memory access pattern for mesh data is varies depending on ray's direction
  - Not suitable for SIMD style architecture
  - FPGAs can optimize it using BRAM (Block RAM = SRAM) and DRAM



$$I_\nu^{out}(\hat{\boldsymbol{n}}) = I_\nu^{in}(\hat{\boldsymbol{n}})e^{-\Delta\tau_\nu} + S_\nu(1 - e^{-\Delta\tau_\nu}) \qquad (1)$$

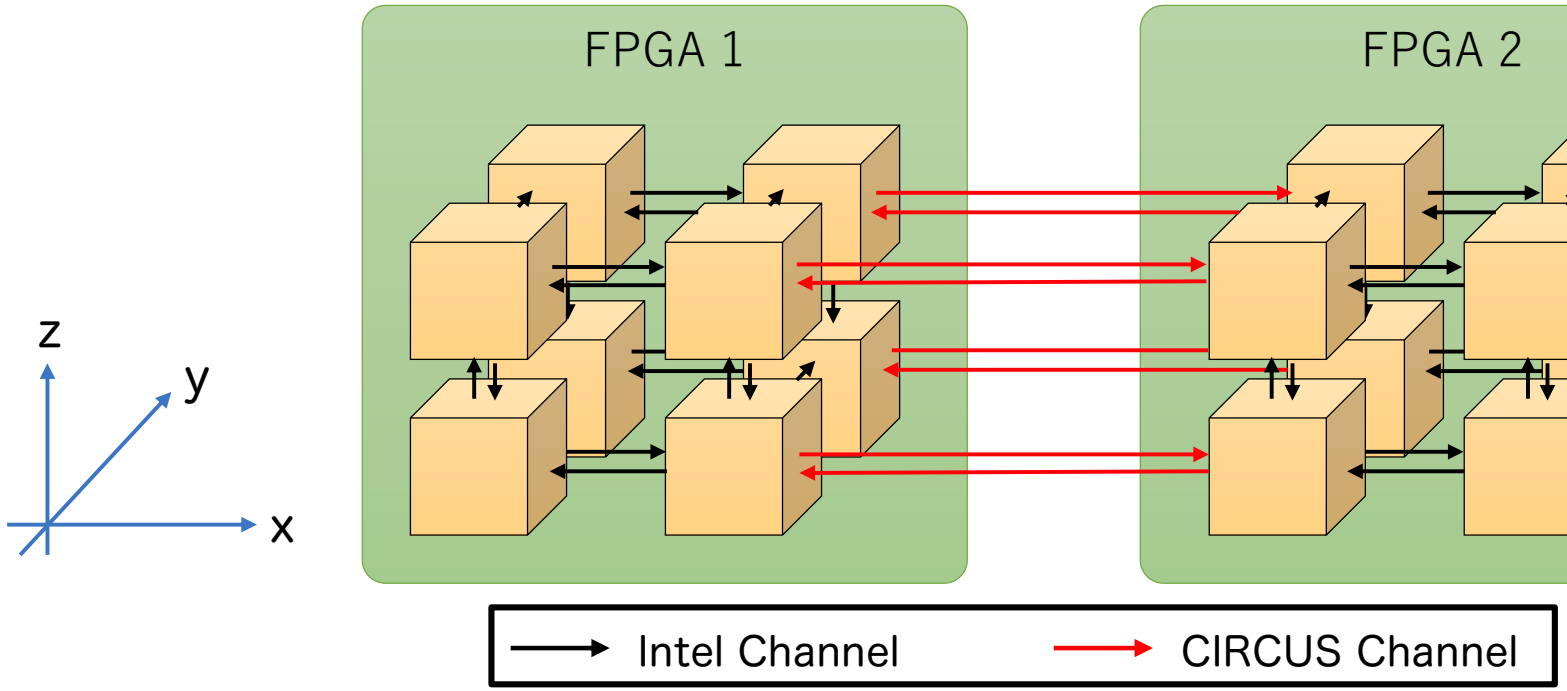
FPGA implementation of ART method

- **Domain Decomposition + Intel Channel for multi-FPGA expansion**

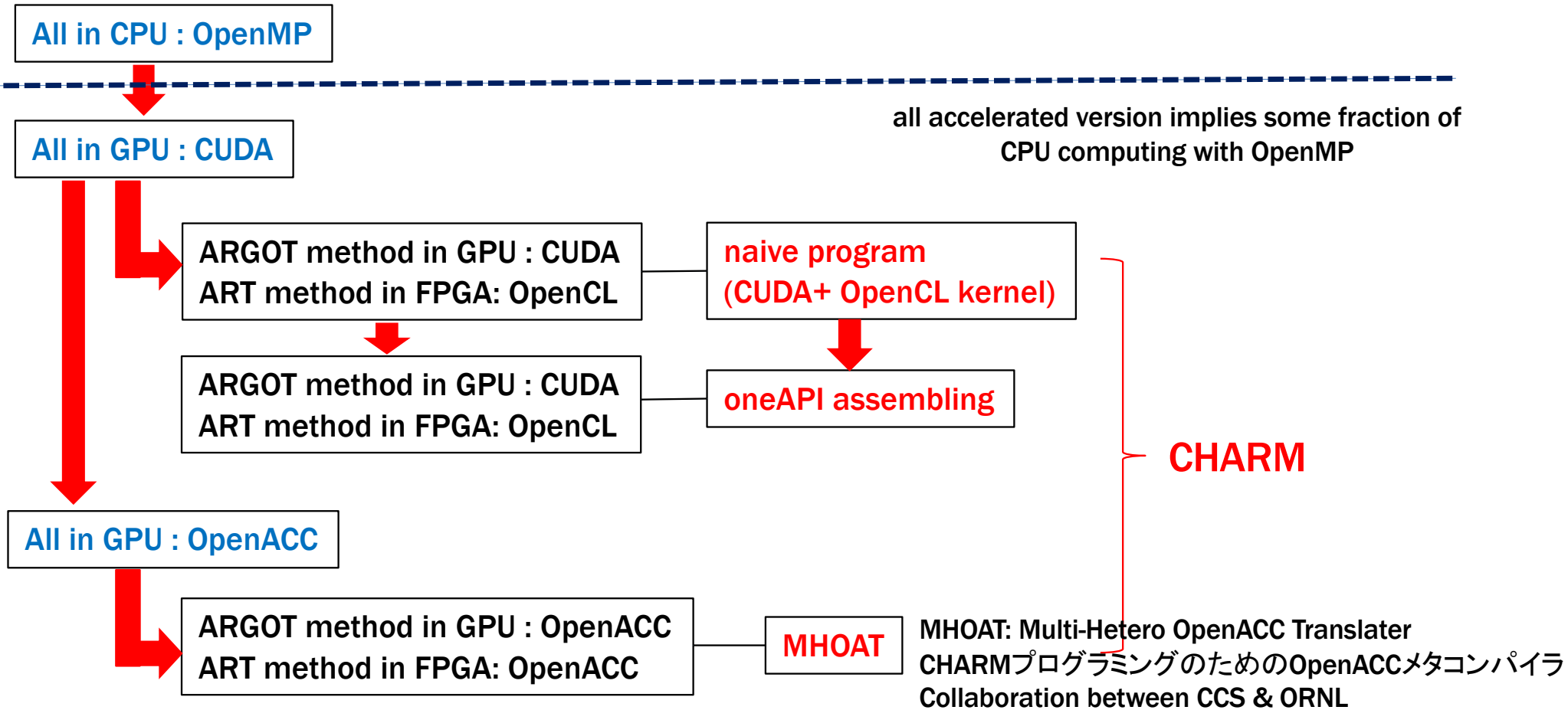


Parallel ART with multiple FPGAs

- replacing Intel Channel at FPGA border with CIRCUS
  - making cluster of ART execution kernels with Intel Channel or CIRCUS



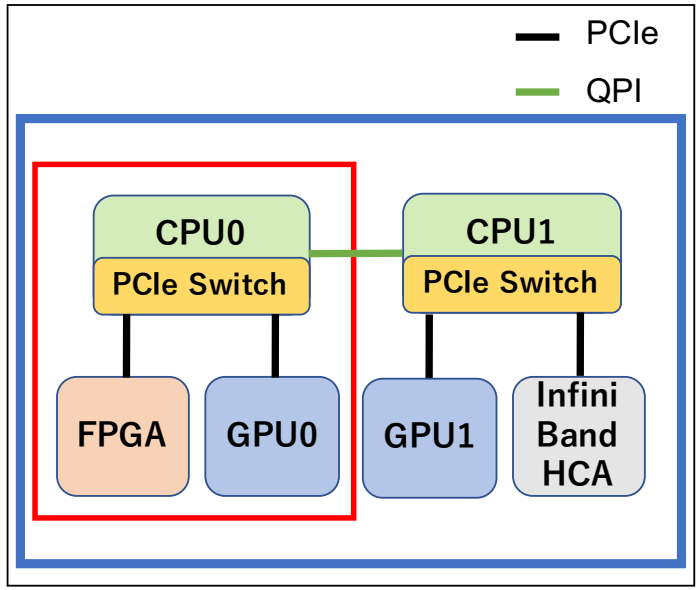
Version history of ARGOT code



Evaluation Environment

- **PPX (Pre PACS-X) mini cluster which is the prototype of Cygnus**

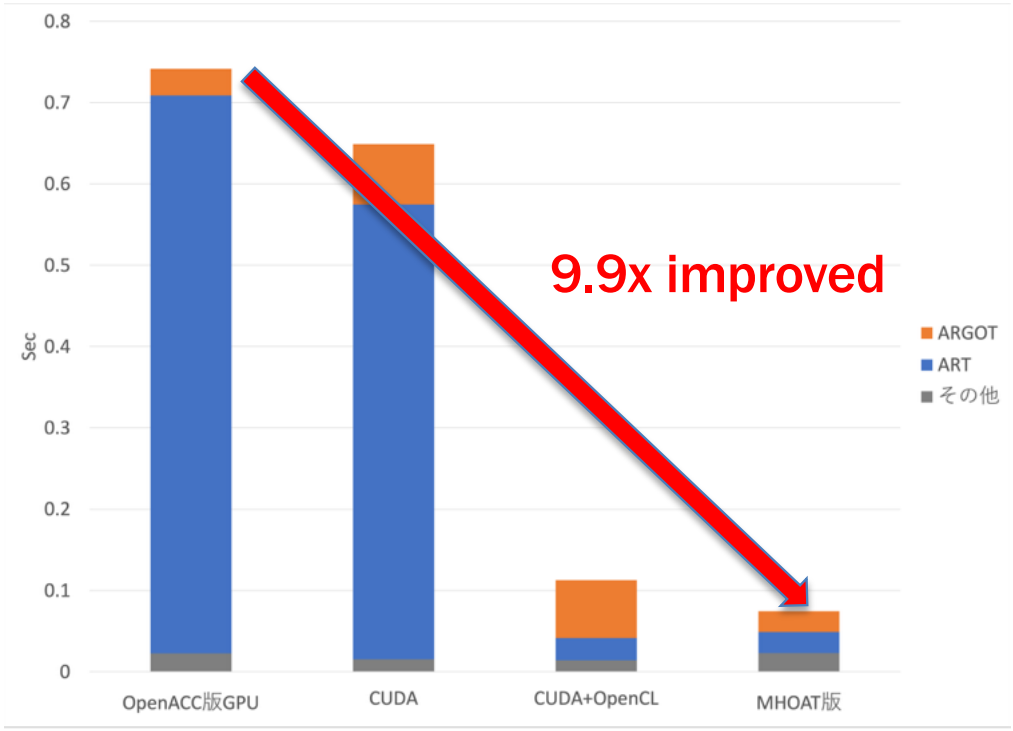
Hardware specification	
CPU	Intel Xeon E5-2690 v4 x2
Host Memory	DDR4-2400 8GB x8
GPU	NVIDIA Tesla P100 (PCIe Gen3 x16)
GPU Memory	16 GiB CoWoS HBM2 @ 720 GB/s with ECC
FPGA	BittWare 520N (1SG280HN2F43E2VG) equipped with 100Gbps x 4 chan.
FPGA Memory	DDR4 2400MHz 32 GB (8GB × 4)
InfiniBand	Mellanox ConnectX-4 Singleport EDR MCX455A-ECAT
Software specification	
OS	CentOS 7.9
Host Compiler	gcc 4.8.5
GPU Compiler	CUDA 11.2.152
Open MPI	3.1.0
OpenCL SDK	Intel FPGA SDK for OpenCL 19.4.0 Build 64 Pro Edition



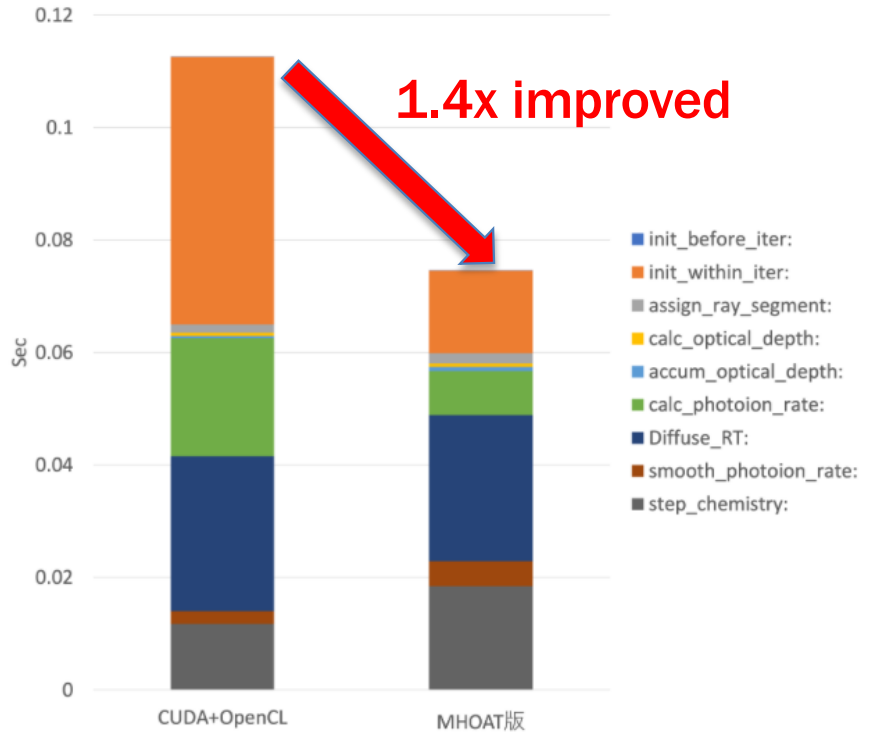
A computation node of PPX

PCIe for FPGA is just gen3 x 8 lanes, but does not affect on performance because the communication overhead is ~1%

Performance ：MHOAT vs CUDA/OpenCL vs GPU-only (PPX single node: GPU: V100 + FPGA: Stratix10　size=32x32x32)



GPU (OpenACC or CUDA) vs CHARM (CUDA+OpenCL or MHOAT)
⇨ MHOAT is the best with 10x speed of OpenACC GPU



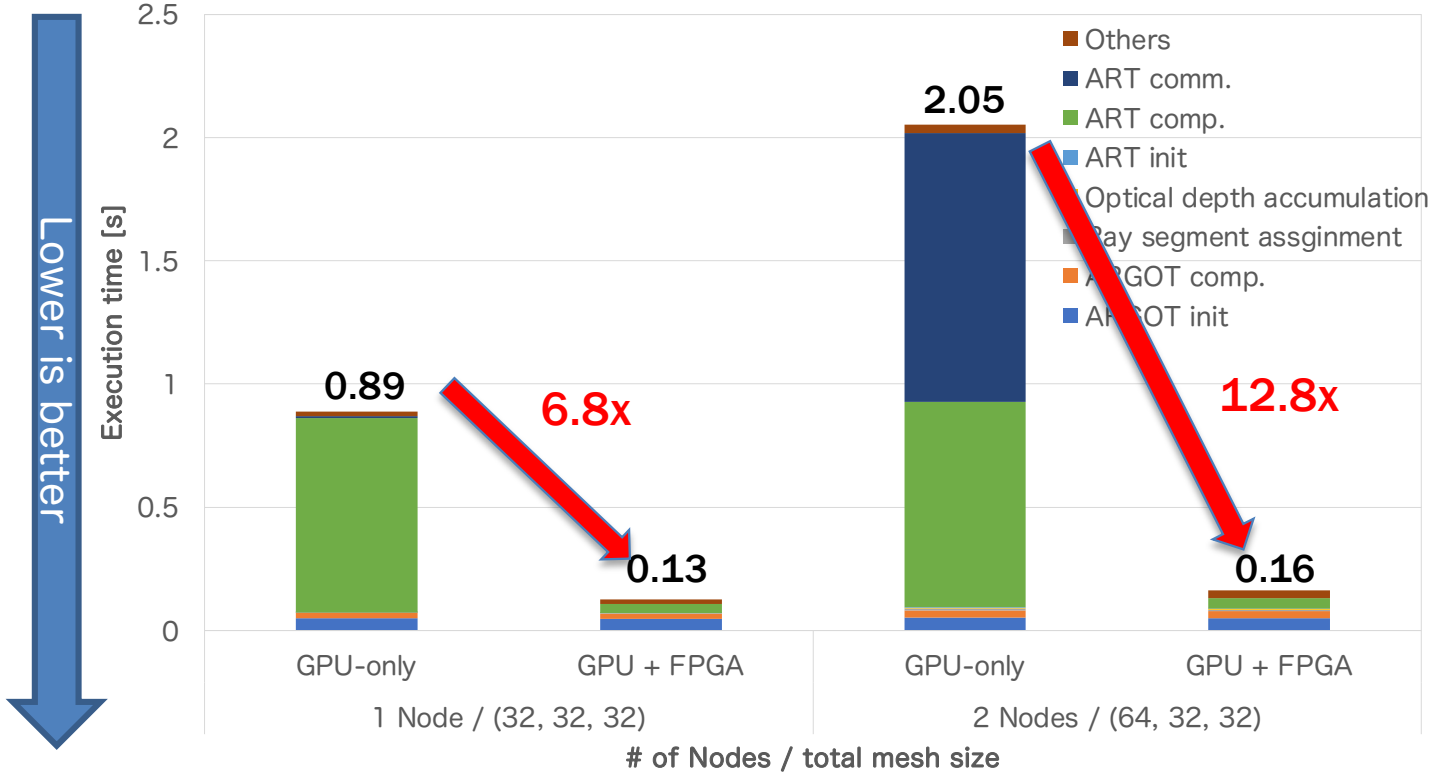
CHARM (CUDA+OpenCL or MHOAT)
⇨ equal with OpenCL on ART method, but ARGOT method is better with OpenACC than CUDA

Parallel ARGOT/CHARM with GPU+FPGA in oneAPI (CUDA+OpenCL)

- **Weak scaling with 2 nodes (2 GPUs + 2 FPGAs) (preliminary evaluation)**
  - **largest mesh size on single FPGA is limited to 32^3**
    - **1 node = 32x32x32　　2 nodes = 64x32x32　　GPU:CUDA　FPGA:OpenCL**
    - **FPGA-FPGA communication with CIRCUS (original FPGA communication system)**

・1 node
  ・GPU+FPGA achieves 6.8x faster than GPU-only

・2 nodes
  ・GPU+FPGA achieves12.8x faster than GPU-only
  → combining computation and communication in pipeline



まとめ

- 牧野さんとの共同研究がきっかけで計算宇宙物理学とHPCの親和性を知ることができた。
- CCSでは梅村雅之先生が身近にいたことで，様々な宇宙物理学問題への取り組みをHPCシステム側から支援し，codesignの重要性を実感した。
- 様々な計算科学分野の中で宇宙物理学は最も成功していると思う。
- GRAPE-1の開発に少しでも関係することができたのは幸せだったと思う。
- 牧野先生，還暦おめでとうございます！