TOP500 и HPL: как измеряют производительность суперкомпьютеров

Что такое TOP500

TOP500 - проект по составлению рейтинга и описаний 500 самых мощных общественно известных суперкомпьютеров мира

Основные рейтинги:

- TOP500 - High Performance Linpack (HPL) - https://www.netlib.org/benchmark/hpl/
- HPCG - High-Performance Conjugate Gradient (HPCG) - https://www.hpcg-benchmark.org/
- GREEN500

Частота обновления: 2 раза в год - Июнь и Ноября

Почему TOP500 важен

- престиж страны, лаборатории или компании
- подтверждение технологического уровня
- маркетинг для производителей железа
- аргумент для финансирования
- тренды в вычислительных системах

Метрики: Rpeak, Rmax и Power

- Rpeak - теоретическая пиковая производительность
- Rmax - полученная производительность HPL
  - Rpeak и Rmax изменяются в числе операций с числами с плавающей точкой (FLoating-point OPerations) в секунду (per Second). P - Peta - 10^{15}
- Power - энергопотребление (kW - киловатты)

Последние результаты TOP500

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**LineShine** - LingKun, LX2 304C 1.55GHz, LingQi, Kylin OS, Shenzhen Cloud Computing Center Co., Ltd. National Supercomputing Centre in Shenzhen (NSCS) China	13,789,440	2,198.40	2,735.82	42,220
2	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,340,000	1,809.00	2,821.10	29,685
3	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607

Что измеряет HPL

HPL решает систему линейных уравнений с одной правой частью Ax=b, где A является плотной матрицей размера n на n, на системах с распределённой памятью

Алгоритм: LU разложение с частичными перестановками

Тип данных: FP64 (Double Precision)

Количество операций: ~ 2/3 N^3

LU разложение

P A = L U

P — матрица перестановок

L — нижняя треугольная матрица

U — верхняя треугольная матрица

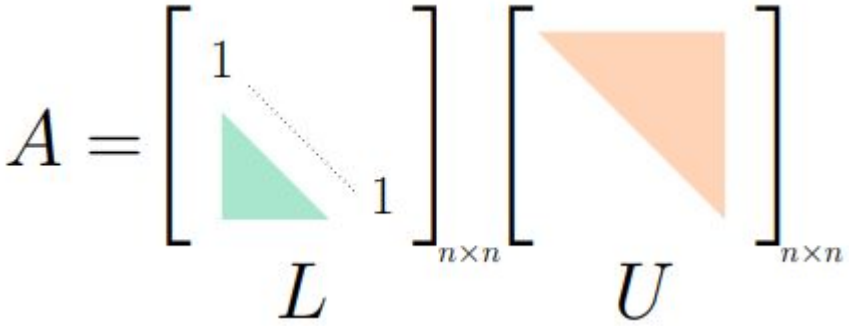
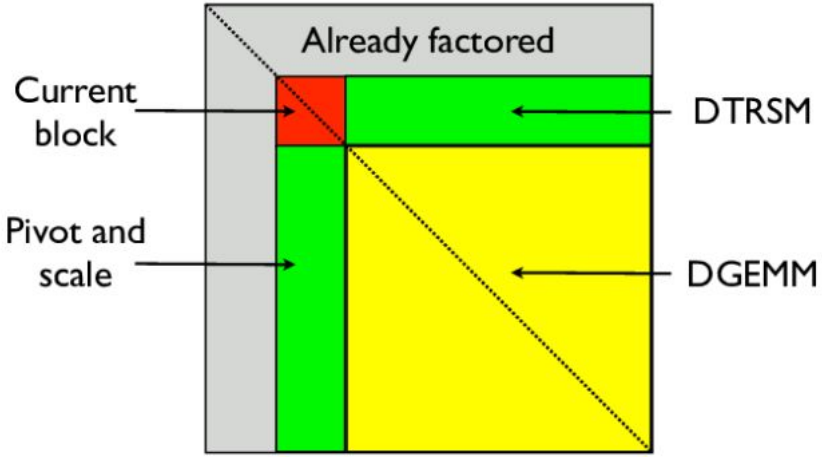
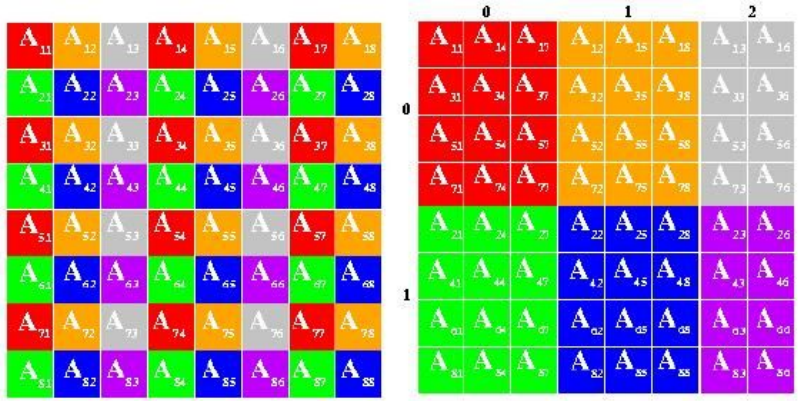
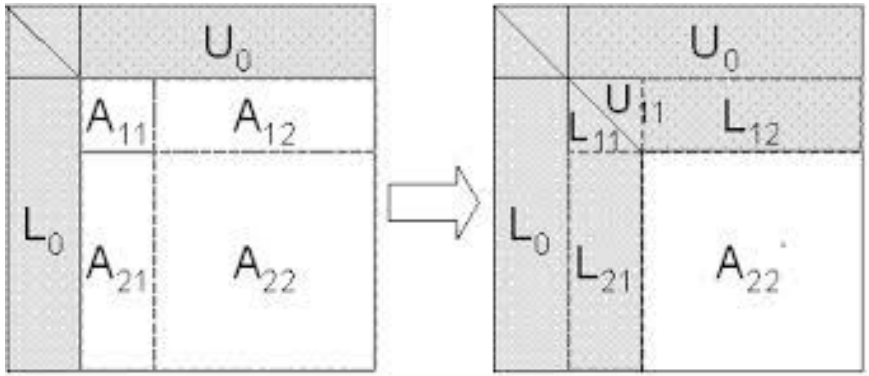


Схема алгоритма HPL

1. Разложение панели
2. Рассылка панели
3. Обработка перестановок строк
4. Обновление оставшейся матрицы
5. Переход к следующей панели







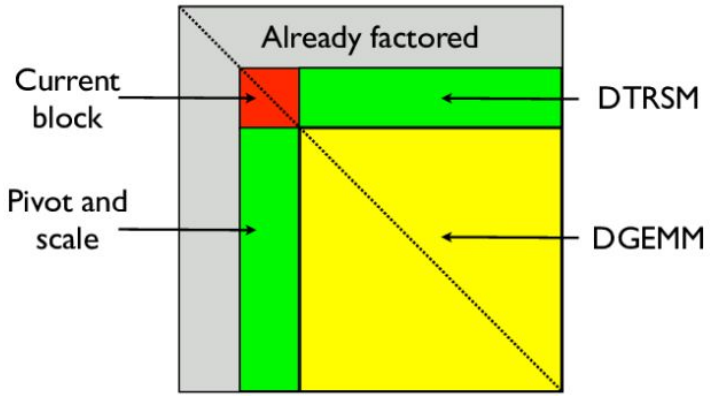
Разложение панели

Панель - узкий блок-столбец (M >>> (N = NB))

Производится последовательность Гауссовых преобразований на панели матрицы A(k) (т. е. A_{11} и A_{21}). После этого матрицы L_{11}, L_{21} и U_{11} известны и мы можем вычислить остальные блоки.

Проблемы:

- меньше параллелизма
- высокая интенсивность коммуникаций
- критический путь

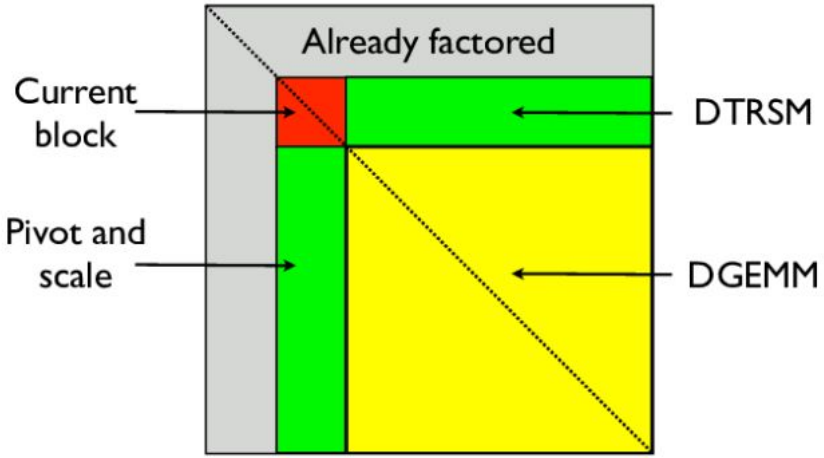


Обновление оставшейся матрицы

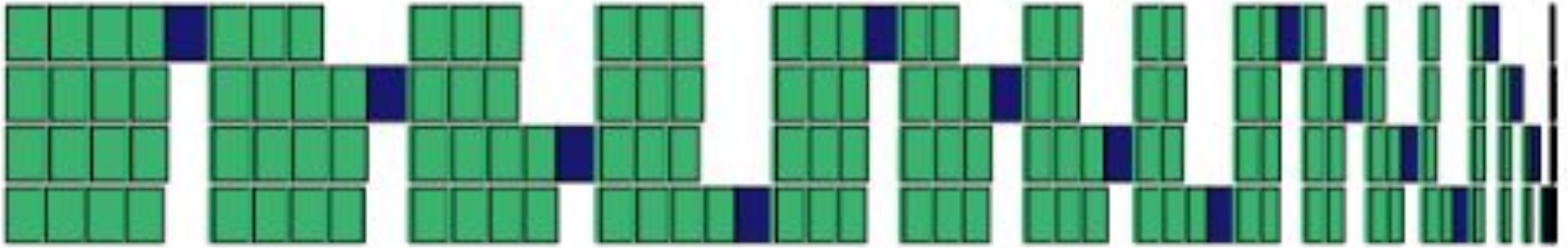
Вычисляется $U_{12} \leftarrow (L_{11})^{-1} A_{12}$, выполняется перестановка строк и дополнение Шура для оставшихся подблоков: $A_{22} = A_{22} - L_{21} U_{12}$.

Основные операции:

- Решение СЛАУ $L_{11}X = A_{12}$ (TRSM)
- Перестановка строк
- Умножение матриц (GEMM)



Выполнение HPL



HPL.dat

Главные параметры:

- N — размер матрицы
- NB — размер блока
- P × Q — двумерная сетка MPI-процессов
- BCAST — способ рассылки панели
- DEPTH — глубина Look-ahead
- SWAP — стратегия перестановок строк
- и пр.

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
4            # of problems sizes (N)
29 30 34 35  Ns
4            # of NBs
1 2 3 4      NBs
0            PMAP process mapping (0=Row-,1=Column-major)
3            # of process grids (P x Q)
2 1 4        Ps
2 4 1        Qs
16.0         threshold
3            # of panel fact
0 1 2        PFACTs (0=left, 1=Crout, 2=Right)
2            # of recursive stopping criterium
2 4          NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
3            # of recursive panel fact.
0 1 2        RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
0            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

Как выглядит результат

```
================================================================================
HPLinpack 2.3  --  High-Performance LINPACK benchmark  --   December 2, 2018
Innovative Computing Laboratory, University of Tennessee
================================================================================

The following parameter values will be used:

N      :   143600
NB     :      384
PMAP   : Row-major process mapping
P      :        4
Q      :        4
PFACT  :    Right
NBMIN  :        2
NDIV   :        2
RFACT  :    Crout
BCAST  :    1ring
DEPTH  :        1
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : non-transposed form
EQUIL  : yes
ALIGN  : 8 double precision words
```

Как выглядит результат

```
================================================================================
T/V                N      NB     P     Q               Time          Gflops
--------------------------------------------------------------------------------
WR11C2R4      143600     384     4     4            1245.51        1.584e+03
--------------------------------------------------------------------------------
||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo ) =        0.0006241       PASSED
================================================================================
Finished 1 tests with the following results:
1 tests completed and passed residual checks.
0 tests completed and failed residual checks.
0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------
End of Tests.
================================================================================
```

Почему HPL хорошо масштабируется

Плотная матрица

-> блочные операции

-> DGEMM

-> высокая загрузка CPU/GPU

Почему HPL не равен “реальная производительность”

HPL:

- линейная алгебра на плотной матрице
- FP64
- частые обращения в одну и ту же память
- регулярные коммуникации по одинаковым паттернам

Реальные задачи:

- разреженные матрицы
- графы
- нерегулярные обмены
- ввод-вывод
- AI-обучение и инференс

Как собирают кластер

- проектирование архитектуры
- установка стоек
- питание и охлаждение
- развёртывание сети
- установка ОС
- драйверы
- MPI / BLAS / компиляторы
- планировщик и мониторинг
- настройка HPL
- проверка и отправка результата

Железо кластера

Вычислительный узел:

- CPU / GPU / HBM / DDR / сетевой адаптер / PCIe / NVLink

Кластер:

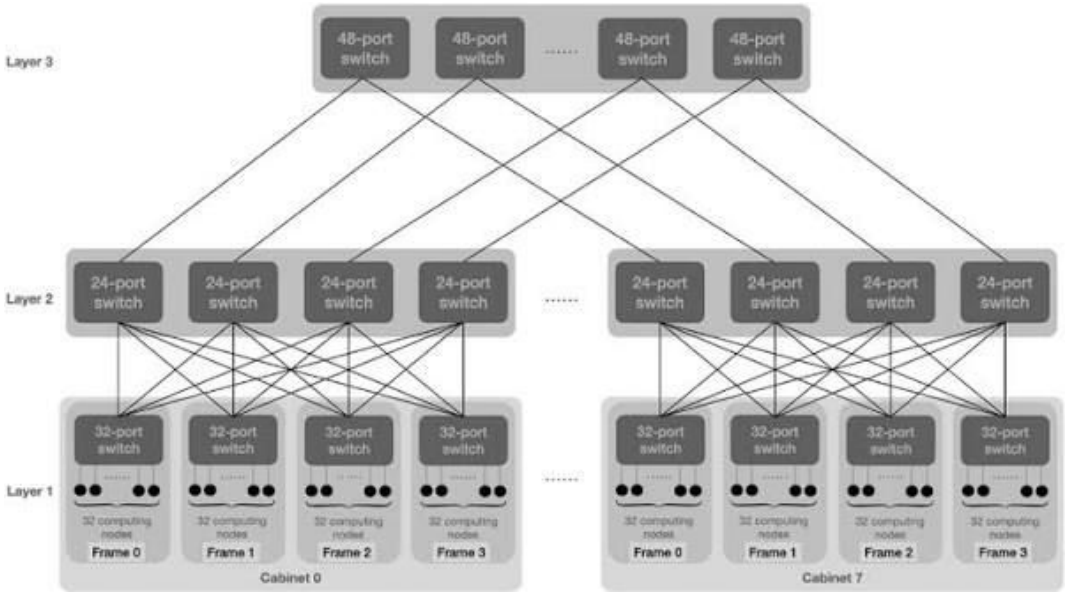
- стойки / сеть / хранилище / узлы входа
- управляющие узлы / сервисные узлы / охлаждение / питание

Сеть

HPL использует сеть для:

- рассылки панелей
- передачи информации о перестановках
- обмена строками

Зачастую реализация MPI полагается на архитектуру сети кластера



Программный стек

- операционная система
- драйверы
- компиляторы
- MPI
- BLAS
- бинарник HPL
- планировщик
- мониторинг
- профилировщики

Software	
Operating System:	Kylin OS
Compiler:	Lclang 1.0.0
Math Library:	OpenBLAS-local
MPI:	OpenMPI with customized LingQi interconnect

Software	
Operating System:	TOSS
Compiler:	g++ 8.5.0 and hipcc 6.4.2
Math Library:	AMD rocBLAS 6.4.2
MPI:	HPE Cray MPI

Оптимизация HPL на практике

- Приложение
  - Подбор параметров N, NB, P, Q
  - Выбор алгоритмов
  - Вычислительные оптимизации (BLAS и HPL)
- Сеть
  - Иерархия сети
  - Топология MPI-процессов
  - Алгоритмы коллективных коммуникаций (MPI)
- ОС
  - Биндинг CPU(NUMA)/GPU на процессы
  - Минимизация потребления ресурсов (CPU/RAM)

TOP500 — это не только FLOPS

Это:

архитектура

+ инфраструктура

+ математика

+ программный стек

+ настройка

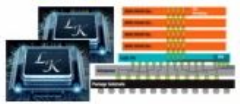
+ люди

Design Principles

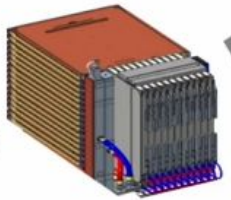
- Application-driven
- Balanced architecture
- Co-design full stack

Compute Node
2 CPUs per node
120.6TFlops

CPU HBM

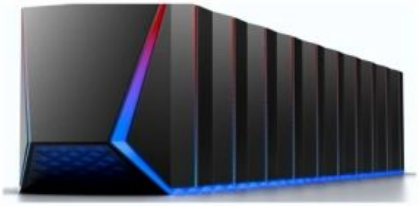


Compute Blade
8 nodes
16 CPUs



Compute Frame
16 Blades

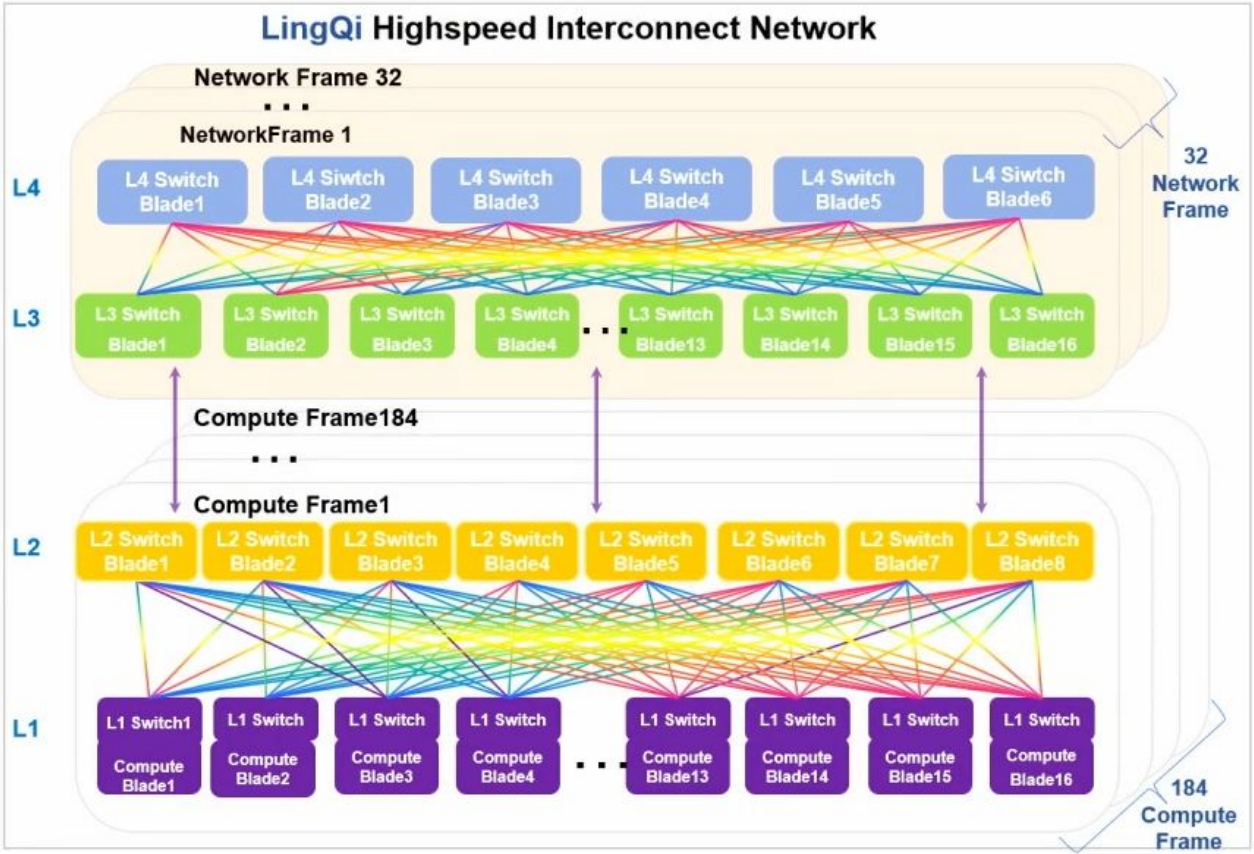
Compute Cabinet
2 Frames
30.87PFlops



Supercomputer
~ 90 Compute Cabinets
22,000+ nodes
13 million+ cores
Sustained: 2.19EF@FP64
Power: 42.22MW
Power Eff: 51.8GF/W

Online Acceleration CPU-Only Architecture

Interconnect Network



Architecture Capabilities

High Performance & Scalability

- 4-layer fat-tree topology
- Large-scale network: 100K nodes, 2M ports
- Bisection bandwidth: ≥3.5 Pbps
- Minimum latency: 1.07 μs

High Reliability

- Dual-plane networking with multi-rail communication
- Credit-based flow control for lossless communication
- Four-layer network with only one optical layer
- Link-, chip-, and cabinet-level redundancy

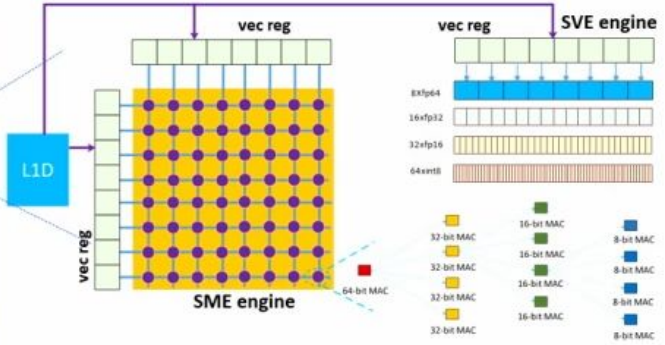
High Maintainability

- Hardware-supported telemetry with second-level data collection and proactive push

High-Performance CPU with Multi-Precision Online Acceleration

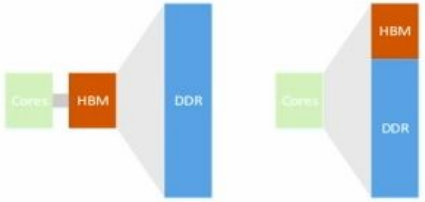
LX2 CPU

- Matrix Online Acceleration
- FP64/32/16, INT8
- Many core, 304 cores, 8 Numa domains
- SVE2, SME, PAC…
- ARM ISA Compatibility
- 60.3TFlops@FP64



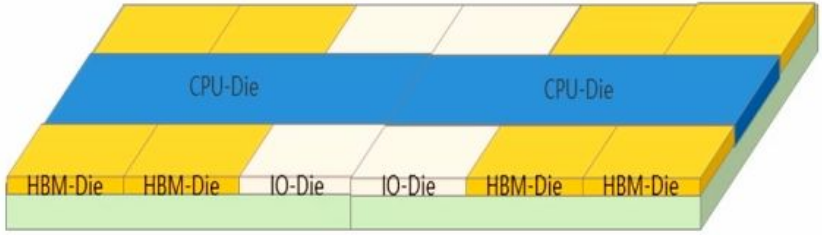
Memory

- High Bandwidth Memory 4 TB/s
- Supports Cache mode and Flat mode

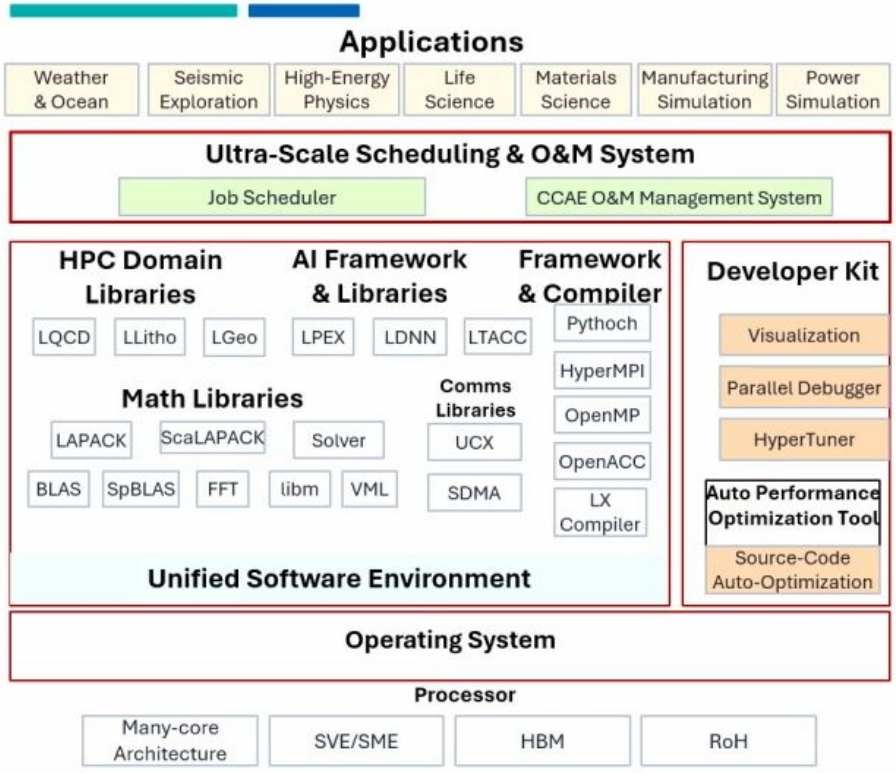


Chiplet Packaging

- On-chip integrated NIC 800Gbps



Software Stack



Enable efficient SME matrixization

SME: Handles unified matmul across HPC and AI workloads
SVE: Complements SME with row-wise and irregular operations

Memory-aware data placement

HBM/DDR: Proactive buffer management
Cache: Maximizes data reuse

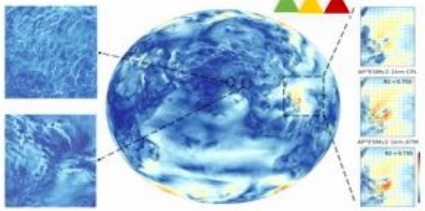
Many cores/Numa aware
OS，compiler and math library

Multi-rail MPI
communication optimization

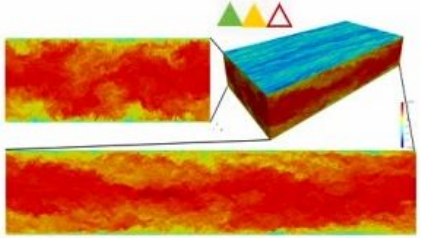
Auto tuning
source-level optimization

Topology aware
scheduler and operations

Scalability of Multi-Domain HPC-AI Converged Applications

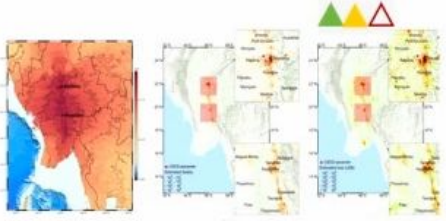


1-km Global Earth-System Simulation

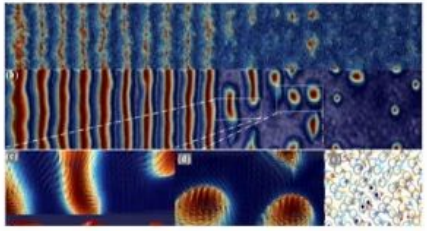


Turbulent channel flow DNS Reτ 2.1/4.3 x 10^4

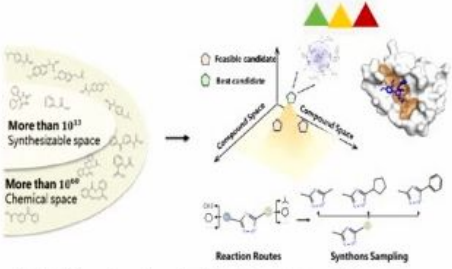
Full-Aircraft WRLES for n Aerodynamic Design



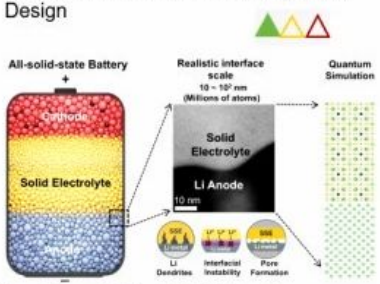
Trillion-grid real-earthquake simulation



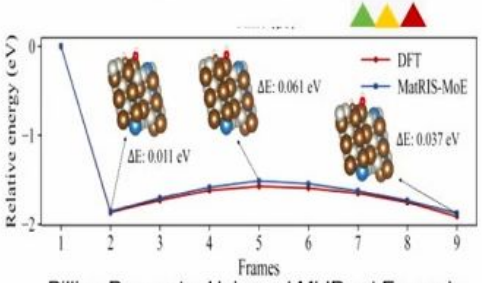
Spin–Lattice Simulation for Skyrmion Formation



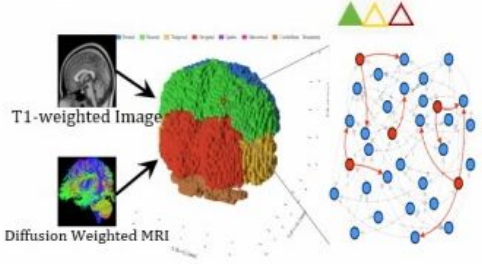
10 Trillion Synthesizable Drug Screening



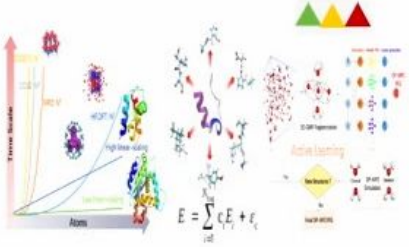
Kohn-Sham DFT at 100 Million Atoms



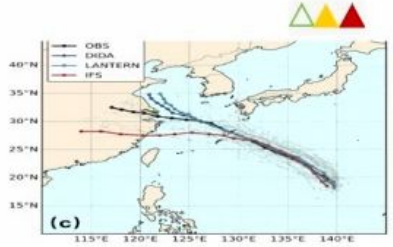
Billion-Parameter Universal MLIPs at Exascale



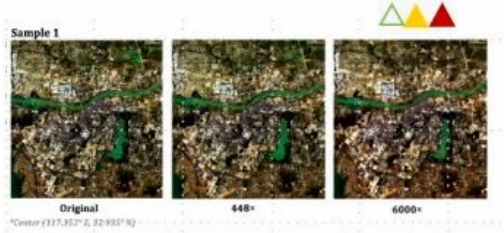
Whole-Brain Simulation at 100-Trillion-Synapse Scale



pMHC-TCR Prediction for Personalized Cancer Vaccine Development



10k-member Global Weather Ensemble Prediction with Minute-scale Cycling



Generative Compression for Earth Observation Data

Simulation Mixed-precision AI