Dense solver for eigenvalue problems on Petascale and towards post-Peta-scale systems

Toshiyuki IMAMURA,
Joint work with
Takeshi Fukaya, Yusuke Hirota
RIKEN Advanced Institute for Computational Science
and Susumu Yamada, Masahiko Machida
Japan Atomic Energy Agency
This work is supported by CREST JST

AICS International Symposium 2016, 22-23, Feb. 2016
at RIKEN AICS, Kobe, Japan

Agenda

1. **Quick Overview of Project**
   - Past and Present of EigenExa
   - Diagonalization of a 1million x 1million matrix on K computer
2. The latest updates
3. Future direction
4. Summary

H4ES (2011-2016)

1. Funded by JST CREST, Post-Petascale system software
2. Eigenvalue problem is of significance in many scientific and engineering fields

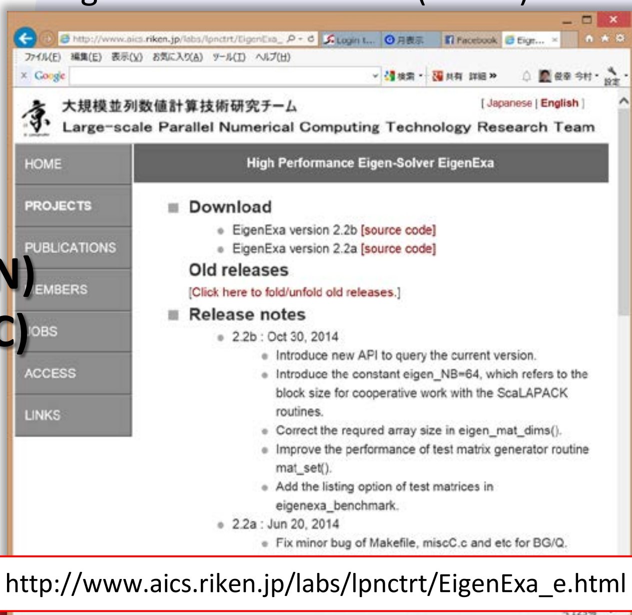
$$Ax = \lambda Bx$$

1. In practical simulation, Sparse and dense solver must be tightly cooperating
2. Project consists Not only theory but computer science and applications

Prof. Sakurai Team 'H4ES'

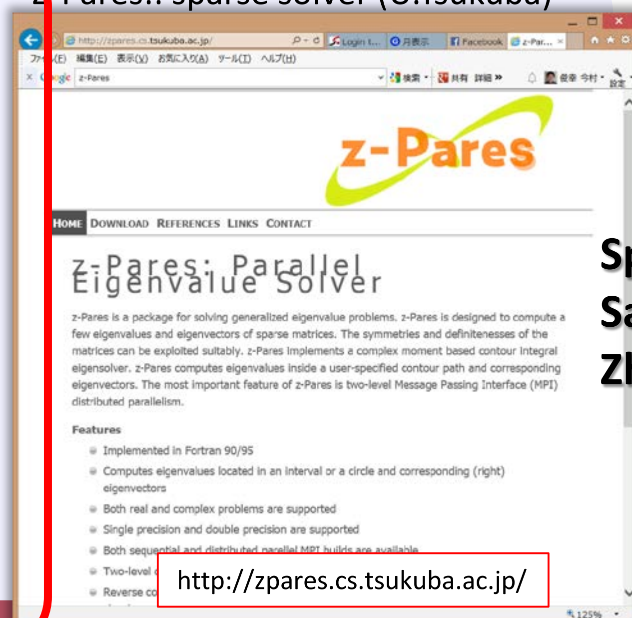
Application:
Hoshi(Tottori)
Kuramashi(Tsukuba)

EigenExa:: dense solver (RIKEN)



http://www.aics.riken.jp/labs/lpnctrt/EigenExa_e.html

Z-Pares:: sparse solver (U.Tsukuba)



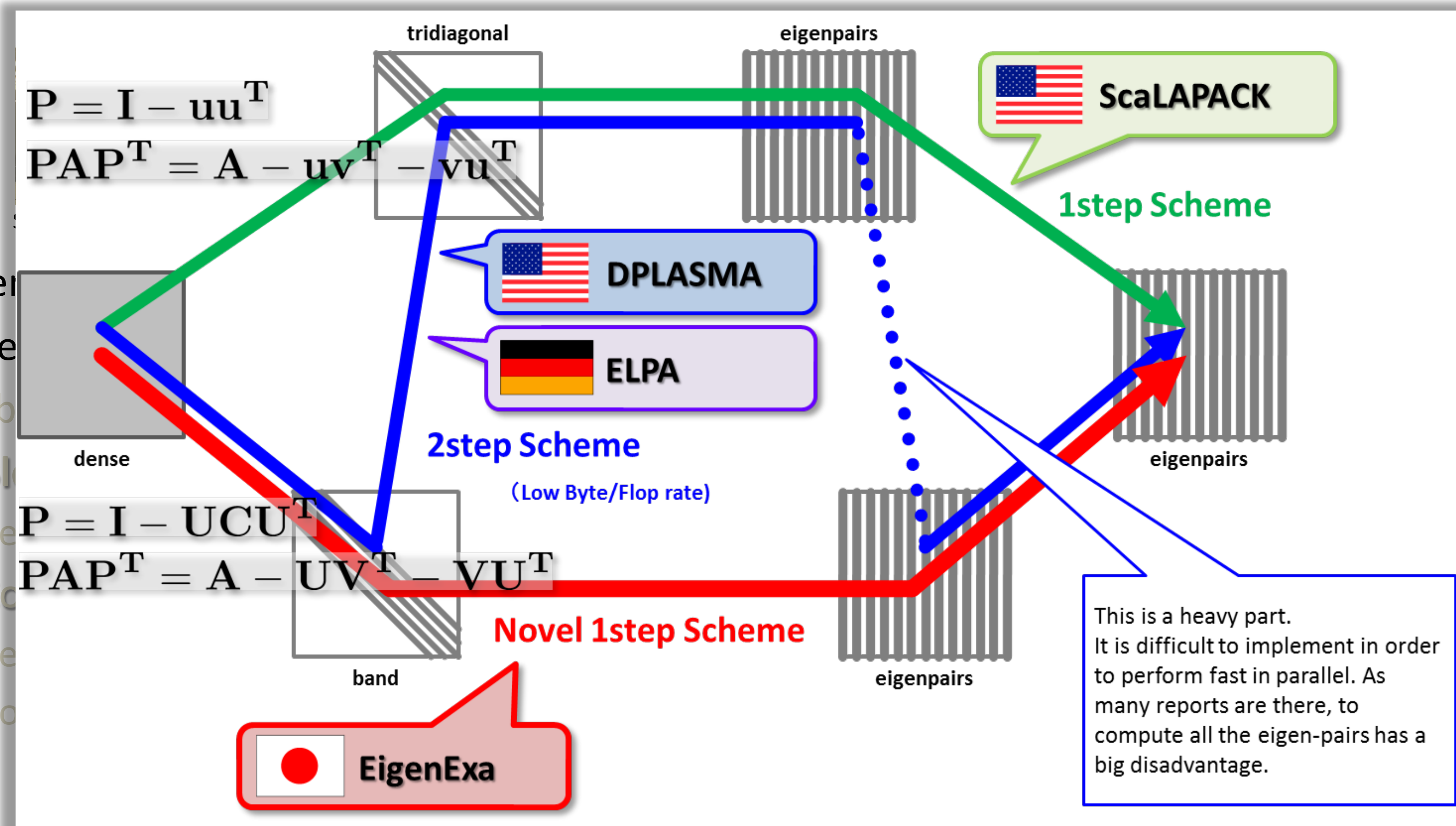
http://zpares.cs.tsukuba.ac.jp/

Dense:
Imamura(RIKEN)
Yamamoto(UEC)

Sparse & LS:
Sakurai(Tsukuba)
Zhang(Nagoya)

EigenExa Project

- Project itself is old…
  - Earth Simulator version was published in SC06. and the speakers continue to update it approximately 10 years. Partly funded by another CREST organized by Prof. Yagawa.

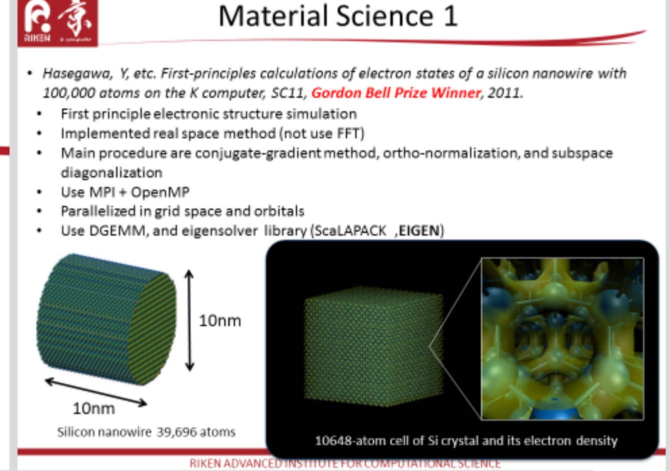


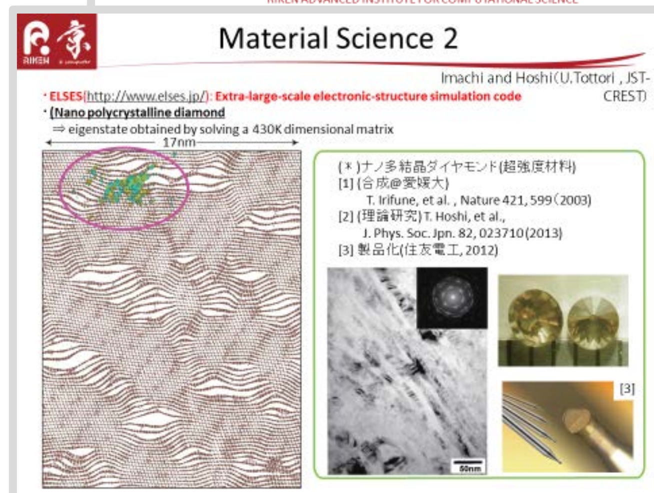
le
??

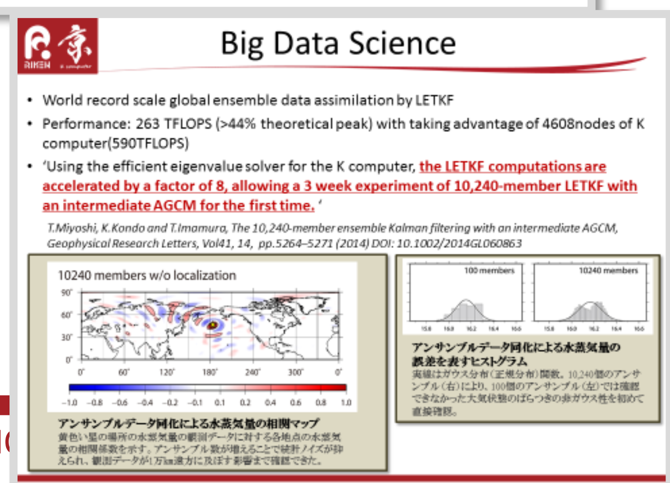
- Curre
  - Eige
- Two b
  1. Bl
     - Re
  2. Co
     - Re
     - Co

Simulation codes

- 8 Applications
  - **Platypus QM/MM:** gives the precise analysis for a biological polymer such as a kinase reaction mechanism by introducing the electron state effect to the molecular mechanics (MM) approach
  - **RSDFT:** is an ab-initio program with the real-space difference method and a pseudo-potential method
  - **PHASE:** is a Computer Software for Band Calculations based on First-principles Pseudo-potential Method
  - **ELSES:** is large-scale atomistic simulation with quantum mechanical freedom of electrons manipulating a large Hamiltonian matrix.
  - **NTChem:** is a high-performance software package for the molecular electronic structure calculation for general purpose on the K computer
  - **Rokko:** Integrated Interface for libraries of eigenvalue decomposition
  - **LETKF:** data assimilation for atmospheric and oceanic systems
  - **POD:** proper orthogonal decomposition (POD) to compress data for example video data

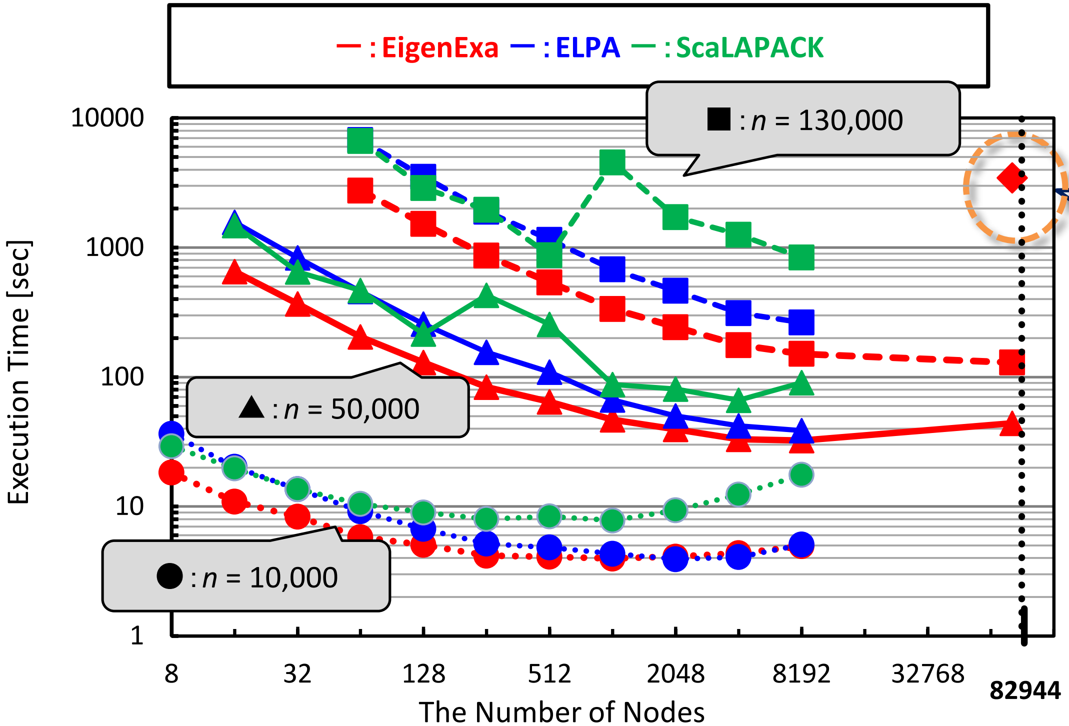






World Largest Dense Eigenvalue Computation

- We have successfully done a world largest-scale dense eigenvalue benchmark (one million dimension) by EigenExa taking advantage of the overall nodes (82,944 processors) of K computer in 3,464 seconds. Our EigenExa achieves 1.7 PFLOPS (16% of the K computer's peak performance).
- **Feasibility and Reliability for algorithm and library are confirmed, especially assumed on a post-K system.**



◆ : n = 1,000,000

EigenExa solves a world largest-scale problem.
(1.7 PFLOPS, 16% of K computer's theoretical peak performance)

$$\max_i \|A\boldsymbol{v}_i - \lambda_i \boldsymbol{v}_i\|_2 / \|A\|_F = 3.1 \times 10^{-13}$$
$$\|V^\top V - I\|_F = 2.1 \times 10^{-10}$$

✓ n is the dimension of problems.
✓ 1 MPI process * 8 threads per node.
✓ Test matrices are randomly generated.

Specification of K computer
- **Peak performance: 10.6 PFLOPS**
- **Num. of Nodes: 82,944**
- **Performance/node: 128 GFLOPS (One octa-core SPARC 64 VIIIfx)**
- **Network: Tofu interconnect (6D mesh-torus)**

Related performance report is
T.Fukaya, TI. "Performance evaluation of the EigenExa eigensolver on Oakleaf-FX: tridiagonalization versus pentadiagonalization", PDSEC2015

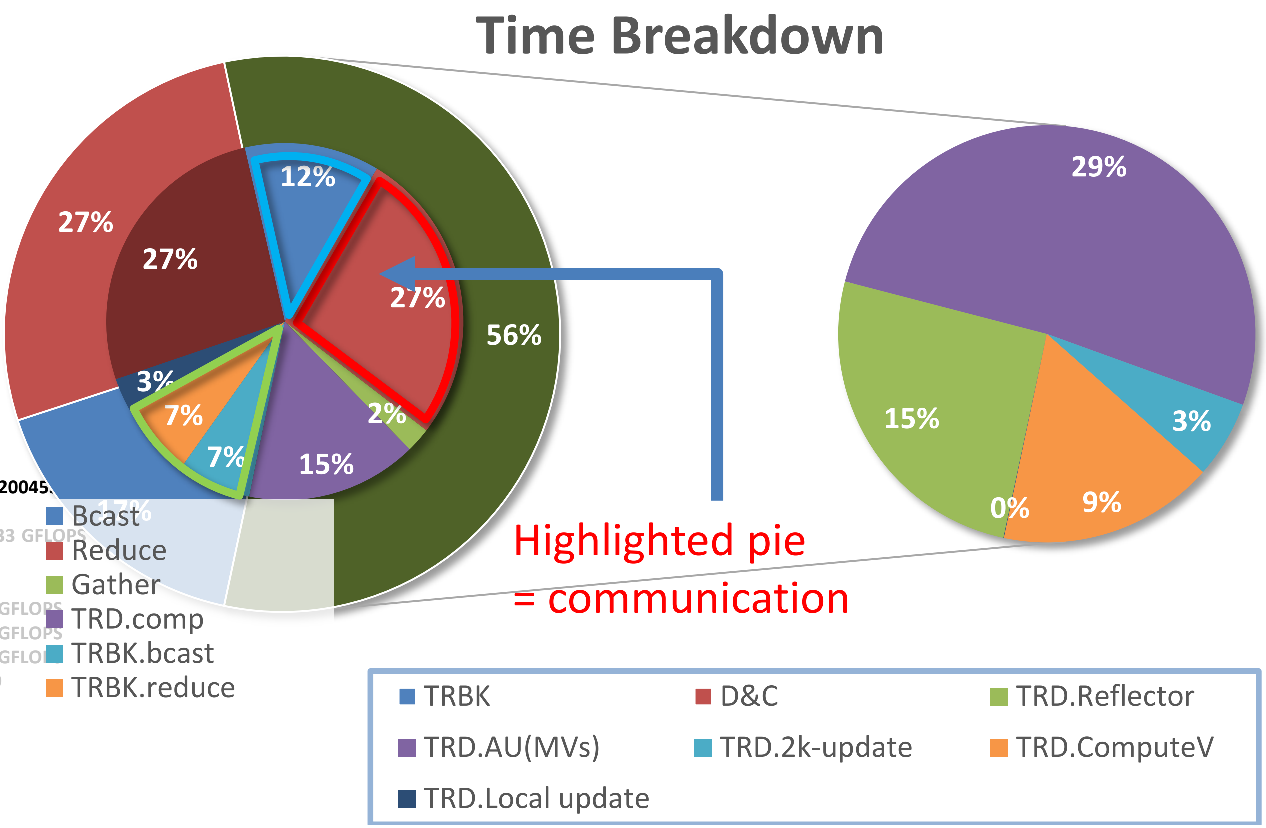
Agenda

1. Quick Overview of Project
   – Past and Present of EigenExa
   – Diagonalization of a 1million x 1million matrix on K computer
2. **The latest updates**
3. Future direction
4. Summary

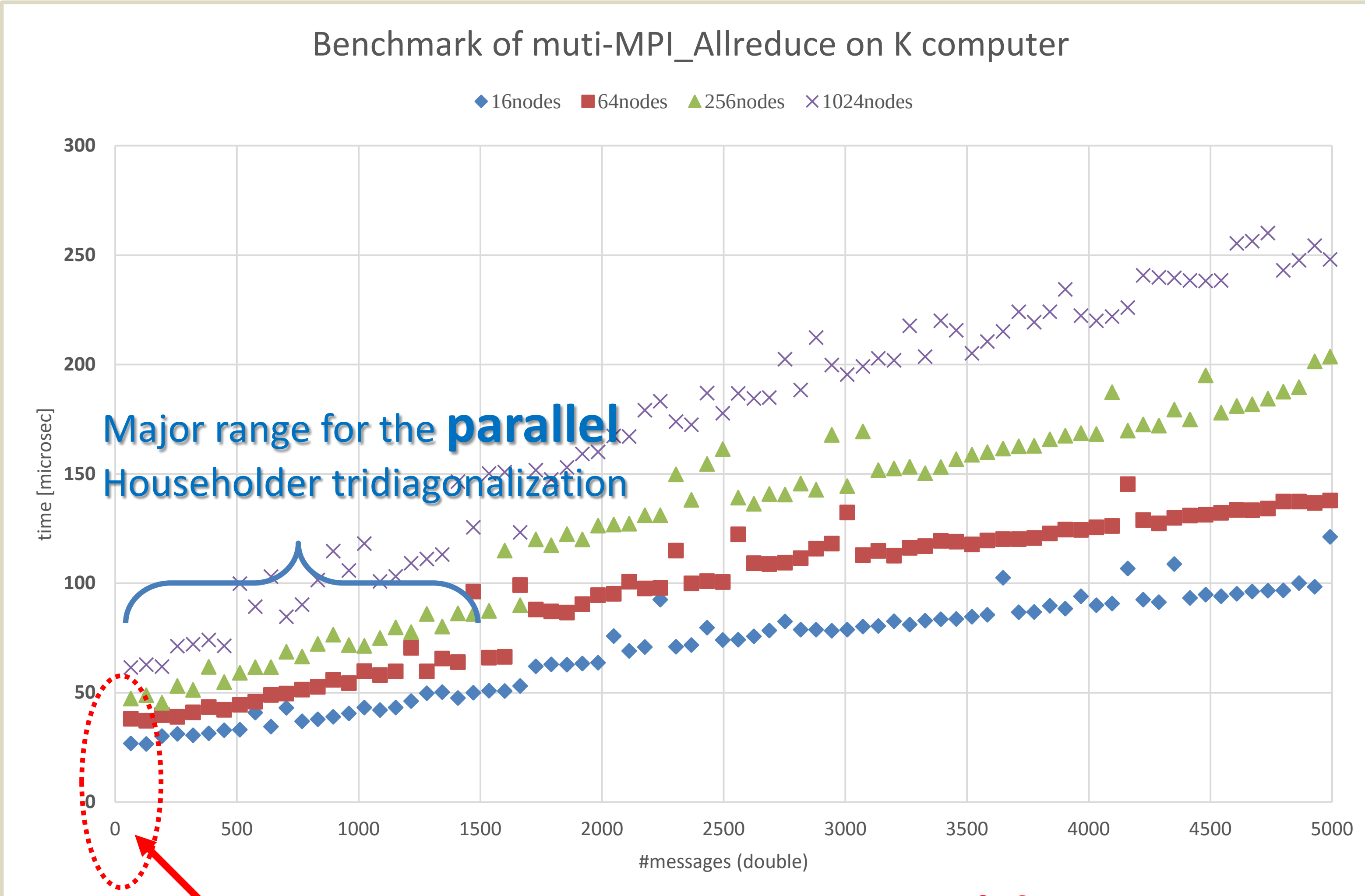
What we got from the ultra-scale experiments

```
NUM.OF.PROCESS= 82944 ( 288 288 )
NUM.OF.THREADS= 8
calc (u,beta)    503.0970594882965
mat-vec (Au)     1007.285000801086 661845.1244051798
2update (A-uv-vu) 117.4089198112488
5678160.294281102
calc v          0.000000000000000
v=v-(UV+VU)u      328.3385872840881
UV post reduction 0.6406571865081787
COMM_STAT
   BCAST  ::  424.3022489547729
   REDUCE ::  928.1299135684967
   REDIST ::  0.000000000000000
   GATHER ::  78.28400993347168
TRD-BLK 1000000 1968.435860157013
677356.7583893638 GFLOPS
TRD-BLK-INFO 1000000   48
before PDSTEDC 0.1448299884796143
PDSTEDC 905.2210271358490
MY-REDIST1 1.544256925582886
MY-REDIST2 14.75343394279480
RERE1 4.861211776733398E-02
COMM_STAT
   BCAST  ::  4.860305786132812E-02
   REDUCE ::  2.155399322509766E-02
   REDIST ::  0.000000000000000
   GATHER ::  0.000000000000000
PDGEMM 532.6731402873993 5417097.56520045
GFLOPS
D&C 921.8044028282166 3130319.580211733 GFLOPS
TRBAK= 573.9026420116425
COMM= 533.7601048946381
    573.9026420116425 3484911.644577213 GFLOPS
    182.3303561210632 5484550.248648792 GFLOPS
    152.0370917320251 6577342.335399065 GFLOPS
    0.1022961139678955 7.379654884338379
COMM_STAT
   BCAST  ::  229.3666801452637
   REDUCE ::  234.4477448463440
   REDIST ::  0.000000000000000
   GATHER ::  0.000000000000000
TRBAKWY 573.9029450416565
TRDBAK 1000000 573.9216639995575 3484796.141101135
GFLOPS
Total 3464.162075996399 1795203.448396145 GFLOPS
Matrix dimension =  1000000
Internally required memory =  480502032  [Byte]
Elapsed time =  3464.187163788010  [sec]
```

World Largest Dense Eigenvalue Computation



Allreduce is an expensive operation



startup cost is 25~60 microseconds!

～equivalent to pure time of allreduce with 1500 words

CA for EigenExa

- **Communication Avoiding algorihtm**
  - Blocking technique, increasing locality by data replication, and exchange the operation order.
  - Introducing an extended form of vector ‘A’.
  - Computing Au and u^Tu, simultaneously.

TI, etc, “CAHTR: Communication-Avoiding Householder Tridiagonalization”, ParCo15

Naïve version of the 2-sided HH trans.

$$s = \mathrm{sign}(\|u\|, -(u,e))$$
$$u := u - se$$
$$v = Au$$
$$[C_U; C_V] = [U^T; V^T]u$$
$$v := v - (UC_V + VC_U)$$
$$f = (u,v)$$
$$v := v - afu$$

Single or several word allreduce

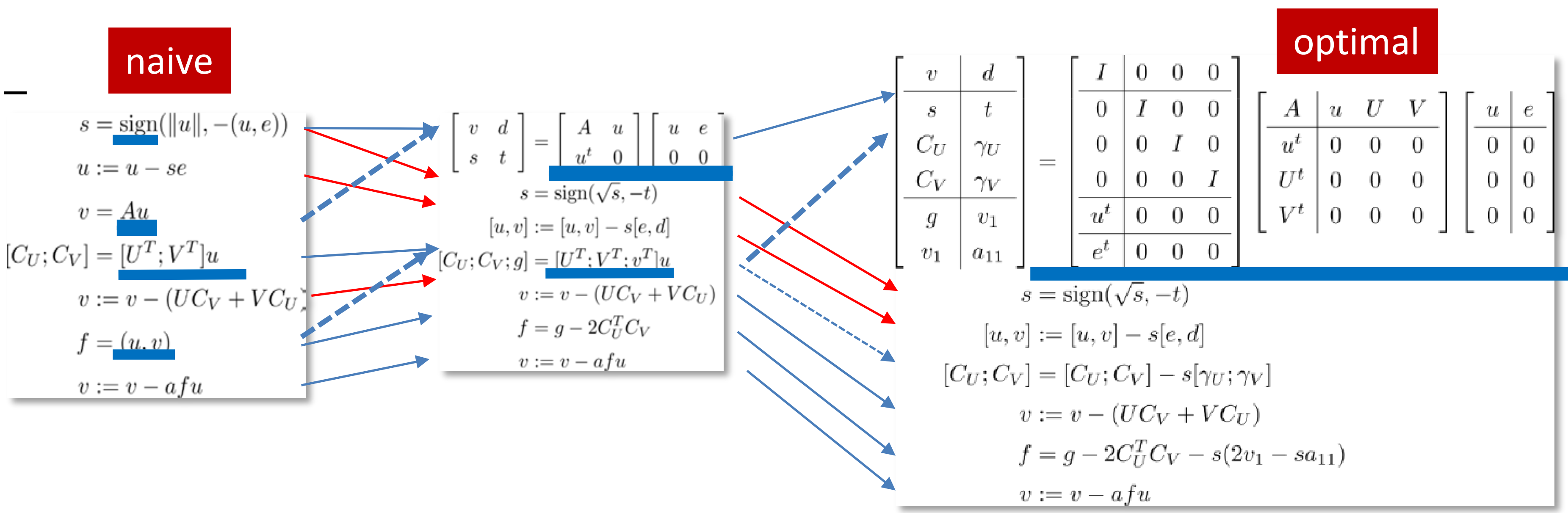
CA for EigenExa

- Communication Avoiding for Householder transformation unlike CAQR
  - Blocking technique, increasing locality by data replication, and exchange the operation order.
  - Introducing an extended form of vector ‘A’.
  - Computing Au and u^Tu, simultaneously.

TI, etc, “CAHTR: Communication-Avoiding Householder Tridiagonalization”, ParCo15

Principles :

Distributive Law && Exchange order && Introducing correction terms && Combine couples of collective operations into one

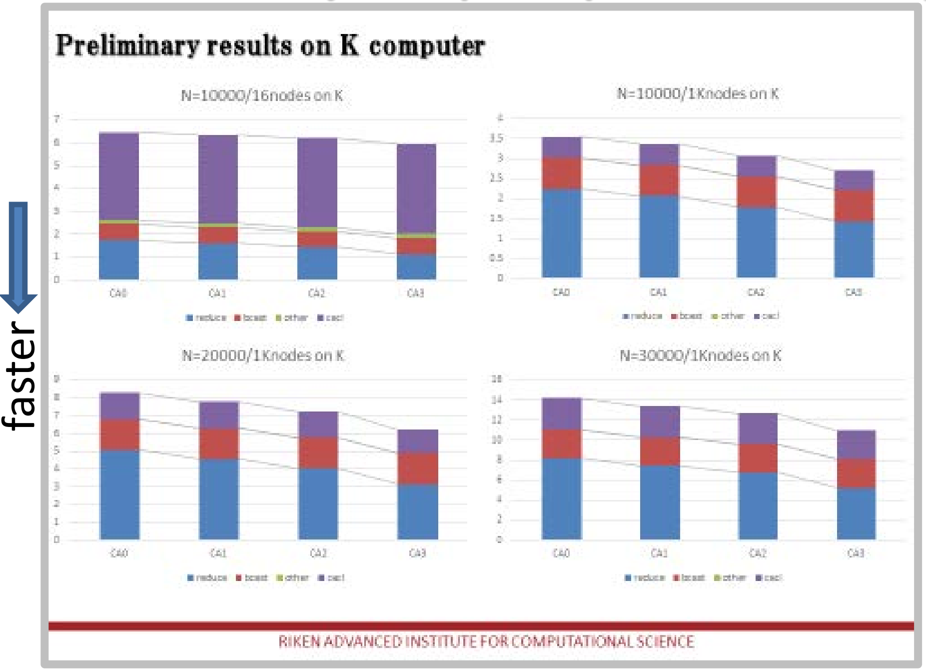


CA for EigenExa

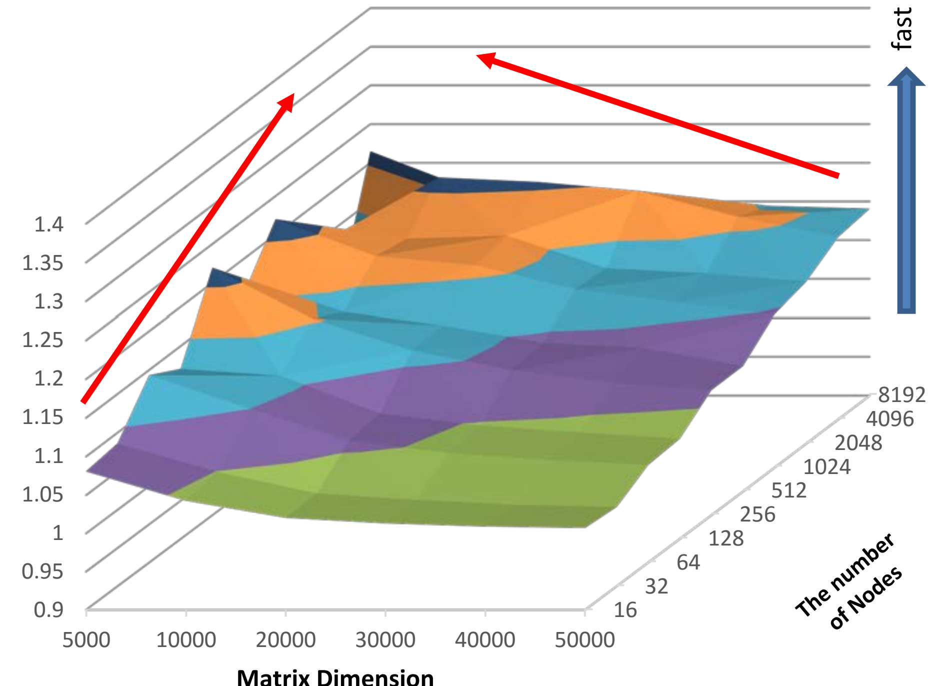
- Communication Avoiding for Householder transformation unlike CAQR
  - Blocking technique, increasing locality by data replication, and exchange the operation order.
  - Introducing an extended form of vector 'A'.
  - Computing Au and u^Tu, simultaneously.

TI, etc, "CAHTR: Communication-Avoiding Householder Tridiagonalization", ParCo15

- **20% decrement is observed, it is a fine result. BUT More AGGRESSIVE decrement is necessary to improve parallel scalability!**



Elapsed time ratio (non-CA/CA)

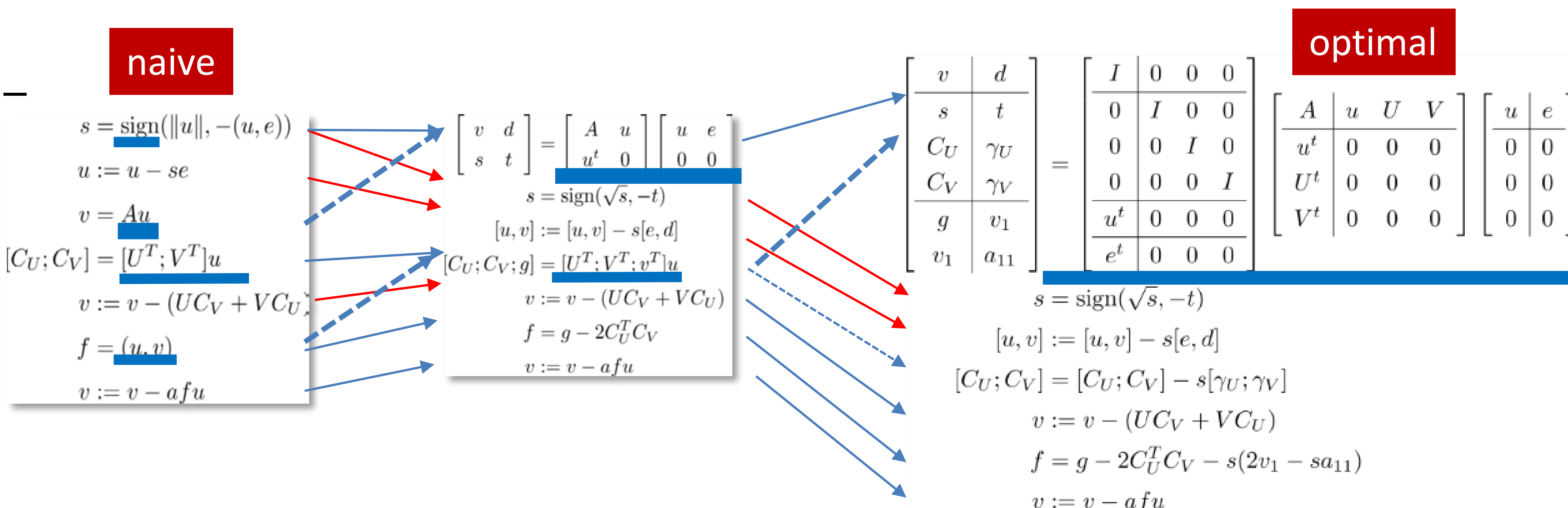


CA for EigenExa

- Communication Avoiding for Householder transformation unlike CAQR
  - Blocking technique, increasing locality by data replication, and exchange the operation order.
  - Introducing an extended form of vector 'A'.
  - Computing Au and u^Tu, simultaneously.

TI, etc, "CAHTR: Communication-Avoiding Householder Tridiagonalization", ParCo15

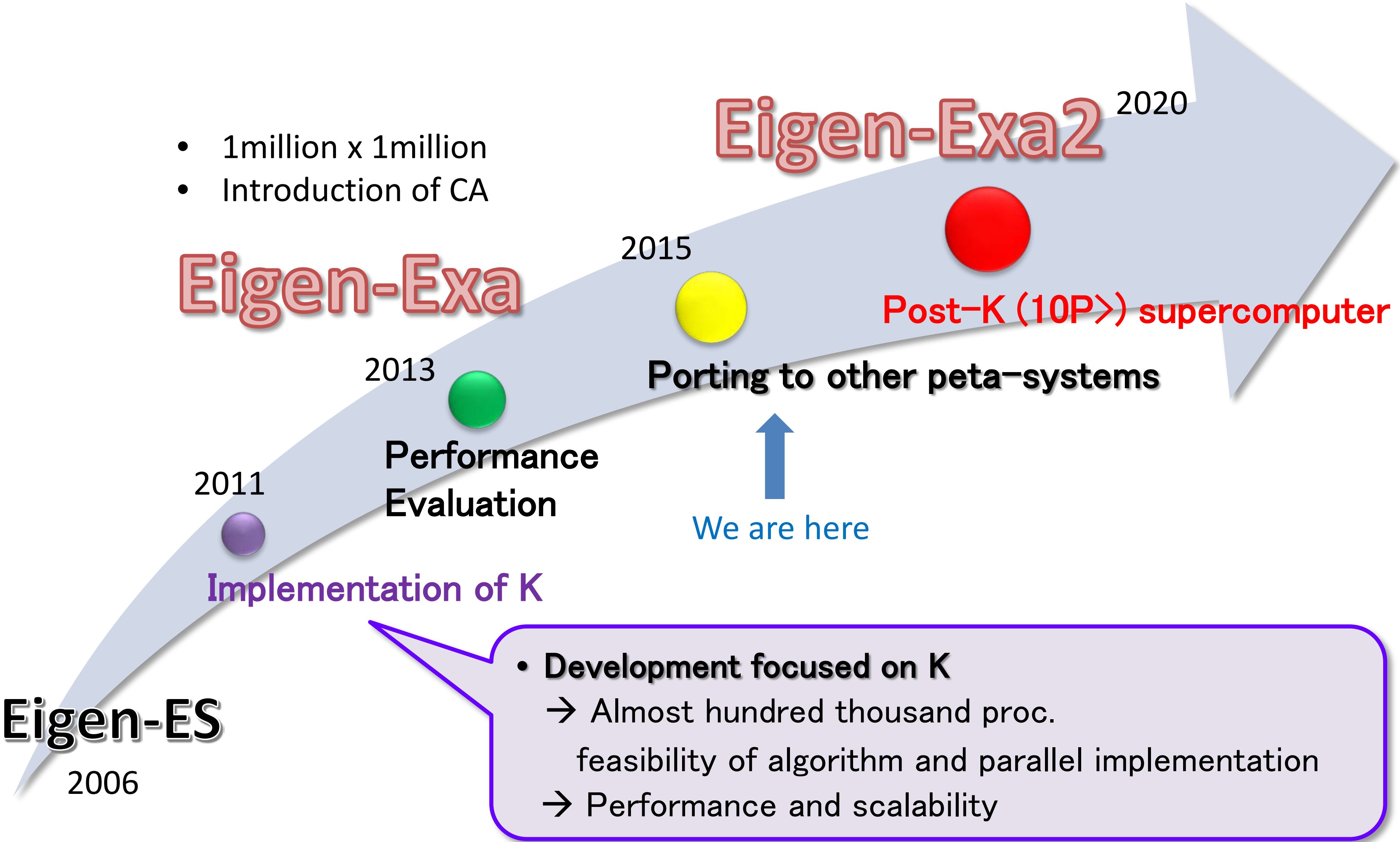
Principles : Distributive Law && Exchange order && Introducing the correction terms && Combine couples of collective ops.



Agenda

1. Quick Overview of Project
   - Past and Present of EigenExa
   - Diagonalization of a 1million x 1million matrix on K computer
2. The latest updates
3. Future direction
4. Summary

Current status

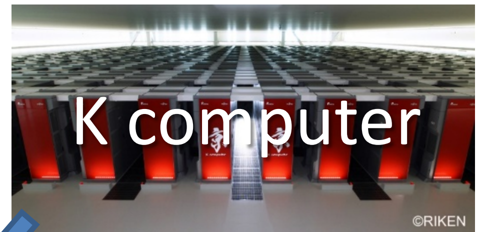


future of the EigenExa Project

- Porting the EigenExa library from K to other systems.

Sorry photo is ES2

T2K
PC cluster



Exa-scale
System??

SX-ACE @ U. Osaka

BlueGene/Q @ Juelich

Photo from:
http://www.hpc.cmc.osaka-u.ac.jp/sx_ace_intro/

Photo from:
http://www.fz-juelich.de/SharedDocs/Bilder/IAS/JSC/EN/galeries/JUQUEEN/juqueen-full.jpg

Persistent Evolutions

- **Hardware**
  - Near-future architecture, such as GPUs, MICs, FPGAs, accelerator boards, …
  - We always change and adapt the target architecture…

    for example, distributed parallel of multi-vector processors, on ES1

    the second generation was cluster of commodity processor and interconnect.

    present version is the third generation.

- **Target problems** (Complex, Tensor, Higher precision)
  - Standard type eigenvalue problems is currently supposed.
  - Generalized version is optional.
  - Not only building IEEE754 double but wider format QP (quadruple precision) is being developed by taking advantage of double-double or multiple-double data format.

- **Algorithm** (revival of old but solid idea to post-Moore era's processing elements)
  - Non-block algorithm but Titling when we focus on local computing
  - Hierarchical block strategy for a case of distributed computing

Target Architecture in near future

- We also have two branch project from EigenExa on the K computer architecture
  - GPU:
    - Eigen-G = Experimental code on a single node + a single GPU environment
    - ASPEN.K2 = Automatic-tuning GPU BLAS kernels, especially, SYMV kernel
  - Intel Xeon Phi
    - Divide and conquer algorithm for GEVP focused on a pair of banded matrices
  - FPGAs ?

GPU

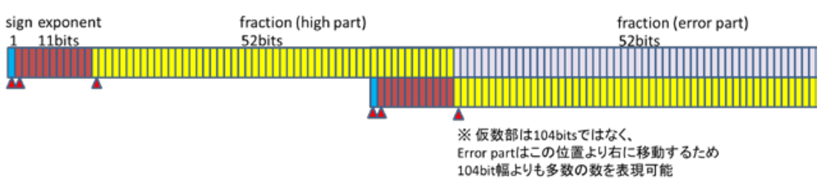
TI, etc, "Eigen-G: GPU-based eigenvalue solver for real-symmetric dense matrices", PPAM2013, LNCS8384
TI, etc. "High Performance SYMV Kernel on a Fermi-core GPU", VECPAR 2012, LNCS 7851,
TI, etc. "Automatic-tuning for CUDA-BLAS kernels by Multi-stage d-Spline Pruning Strategy", @^2HPSC 2014

MIC

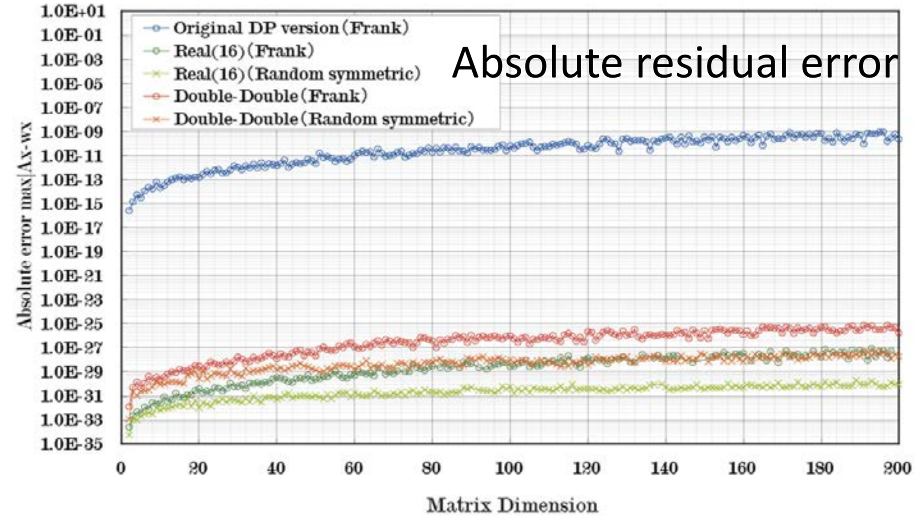
Y.Hirota, etc, "Divide-and-Conquer Method for Symmetric-Definite Generalized Eigenvalue Problems of Banded Matrices on Manycore Systems",SIAM LA15
Y.Hirota, etc. "Acceleration of Divide and Conquer Method for Generalized Eigenvalue Problems of Banded Matrices on Manycore Architectures, PMAA14.

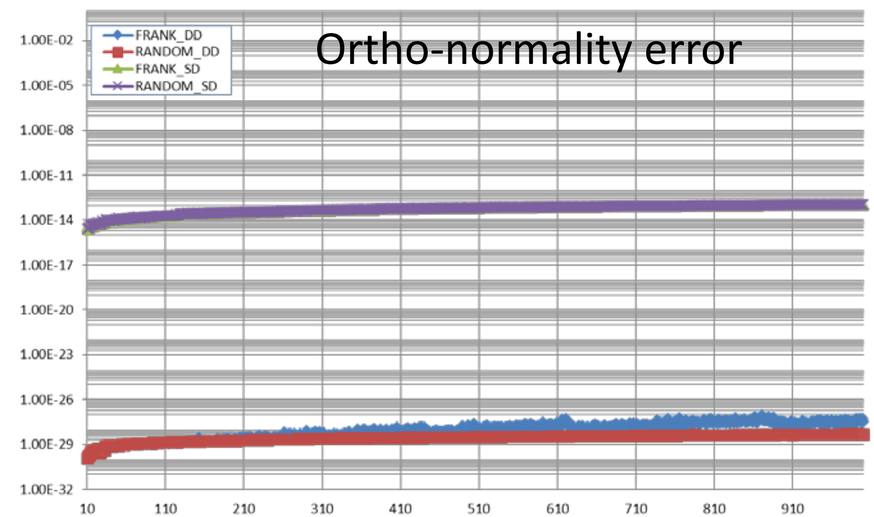
QP(Quadruple Precision)

- Emerging long-time and large-scale computation, **rounding error** on the IEEE754 ‘double’ floating point format with O(10^15) operations will be a considerable issue. The **DD, double-double, format** (D.H.Bailey, DDFUN90, http://crd.ldl.gov/~dhbailey/mpdist) is one of promising technologies to ensure higher precision without the help of special hardware. The DD format consists of the ‘high’ and the ‘error’ parts, and their summation represents higher precision data.
- $$a_{dd} := a_{hi} + a_{err}, \quad (|a_{hi}| > |a_{err}|)$$



- Addition and multiplication of two DD-format data are defined simply with approximately 20 double-precision floating operations. It is expected to help several issues on multicore platforms like accuracy and utilization problems. In this study, we are developing a double-double precision (**quadruple precision**) eigenvalue solver, ‘**QPEigenK**’. It performs on distributed memory parallel computers. In addition, **OpenMP and MPI parallel models** are supported.





S. Yamada, etc. High Performance Quad-Precision Eigenvalue Solver: QPEigenK, ISC15 Poster Presentation.

Other topics for numerical linear algebra to be discussed in Exa-scale computing

- Reproducibility
  - We recognize that round off error is naturally included in the results.
  - But,
    - Even, initial data and HW/SW configurations are same, the results might have a bit-wise difference due to un-deterministic behavior of thread or other factors.
    - In MPI, data-distribution over nodes, process grid, and data size also affect the results.
  - By introducing
    - QP libraries mentioned in last slide or Error Free Transformation in basic linear kernels such as BLAS, we can **guarantee full-bit accuracy of the IEEE754 double-format.**
- Resiliency
  - ABFT (Algorithm-Based Fault Tolerance)
    - We take advantage of Algorithmic Redundancy for cross-check and **Error-detection-and-correction of fault of memory traffics and floating-point calculations.**
- Higher order or abstract data format
  - **Tensor** analysis, etc.

Collaboration

- **The project in JST CREST (2011-2016), has been extended 2-year duration with the international collaboration France (ANR), Germany (DFG), and Japan (JST).**
  > Numerical algorithm, higher precision eigensolver
  – Prof. Dr. Bruno Lang, Univ. Wuppertal
- **Joint Laboratory for Extreme Scale Computing**
  > Porting dense eigenvalue solvers to various systems
  – Ms. Inge Gutheil, and Prof. Dr. Johannes Grotendorst , Juelich Supercomputer Center
- Personal relations
  – Dr. Hermann Lederer, Max Planck Computing and Data Facility, and Prof. Dr. Thomas Huckele, Technische Universitaat Muenchen
    > Exchange technical information between ELPA and EigenExa
  – Dr. Osni Marques, Lawrence Berkley National Laboratory
    > Discussion of future SVD algorithms
  – Prof. Weichung Wang, National Taiwan University
    > Discussion of application of a GPU-eigensolver
  – Dr. Roman Lakymchuk, KTH Royal Institute of Technology
    > Discussion of Reproducibility

Summary of talk

- **EigenExa project (2011-2016)**
  - The first milestone : 1million order eigenvalue computation with full nodes of K computer.
  - Second milestone : optimization of communication
- **We struggled against 2 types of bottleneck**
  - **Memory bandwidth → Block algorithm**
  - **Network Latency → Communication avoiding（CA) and communication hiding（CH)**
- **Near-Future work**
  - Establish the CA technology for total performance of EigenExa
  - Quadruple Precision version
  - Vector computers, other platforms
  - GPU cluster, MIC cluster, etc.
- **Topics for Collaboration is broad,**
  - **New target architectures, FPGA ? or ?**
  - **New topics must be also concerned like Reproducibility and FT**
  - **New Collaboration with CS and Applications!**

THANKS!
ありがとうございました