Automatic Tuning for Collective Communication Operations

Toshiyuki Imamura
University of Electro-Communications

Thanks to Dr.Machida and Dr.Yamada, JAEA

Introduction

- Collective communication:
  - Broadcast, allreduce, etc.

    plays a great role in parallel programming.

Example, eigensolver

- Real-Symmetric Std. eigenproblem
  - Householder-tridiagonalization

    broadcast, allreduce, re-distribution(scatter+gather)

Tab.1: Eigensolver with the vendor-tuned MPI on Altix3700Bx2 32PEs ([sec], ( )=%)

	N=1K	N=4K	N=6K
Total	.187	1.745	4.303
Bcast	.030(13)	.140(8.0)	.247(5.7)
Allreduce	**.081(43)**	**.561(32)**	**1.158(26)**
Re-dist.	.021(11)	.218(12)	.442(10)

42%!

Introduction

- Collective communication:
  - Broadcast, allreduce, etc.

    plays a great role in parallel programming.
- Optimal algorithms of collective communication:
  - depend on *the topology of the candidate processes and network, the message counts, data sets to be passed.*
- Dynamic factors:
  - Above-mentioned factors are varied
  - The better **parameter** *(=communication algorithm, routing, segments, etc.)* should be chosen in every execution.

An idea of auto-tuned collective communication arises as a new application of the auto-tuning methodology.

Collective communication operations

MPI_Bcast, MPI_Reduce, …

Collective Communication

- Algorithm (Vadhiyar04)

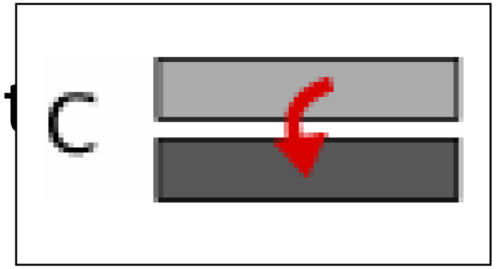
Collective Ops	Algorithms
Broadcast	Seq., Chain, Binary, Binomial, V.Geijn
Scatter	Seq., Chain, Binary, Binomial
Gather	Seq., Chain, Binary
Reduce	Gather+OP, Chain, Binary, Binomial, Rabenseifner
Allreduce	Reduce+bcast, Allgather+OP, Chain, Binary, Binomial, Rabenseifner

$O(\log_2 P)$ algorithms are know as the best!

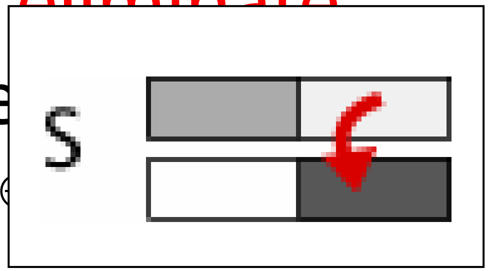
Simple re-definition of the binary algorithms

[Primitives]

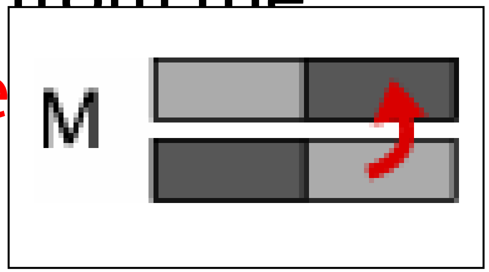
- **$C^{(d)}$** (Copy): Send all the data from t
process *r* to $r'(=r \oplus 2^d)$.



- **$S^{(d)}$** (Split): Split data and send and eliminate
the half of data (indices are specifie
N]) from the root process *r* to $r'(= r$



- **$M^{(d)}$** (Merge): Send the all the data from the
root process *r* to $r'(=r \oplus 2^d)$ and me
data specified by $f_{r,d[1:2N]}$.



***Consecutive S, M** operations derives*
Vector Recursive Halving and Doubling algorithm.

Bcast (broadcast)

1. Binomial

   $$\text{Bcast} = \prod_{i=0}^{p-1} C^{(i)}$$

2. V.Geijn/Rabenseifner(recursive halving/doubling)

   $$\text{Bcast} = \prod_{i=p-1}^{0} M^{(i)} \prod_{i=0}^{p-1} S^{(i)}$$

3. Hybrid1

   $$\text{Bcast} = \prod_{i=j}^{0} M^{(i)} \prod_{i=j+1}^{p-1} C^{(i)} \prod_{i=0}^{j} S^{(i)}$$

4. Hybrid2

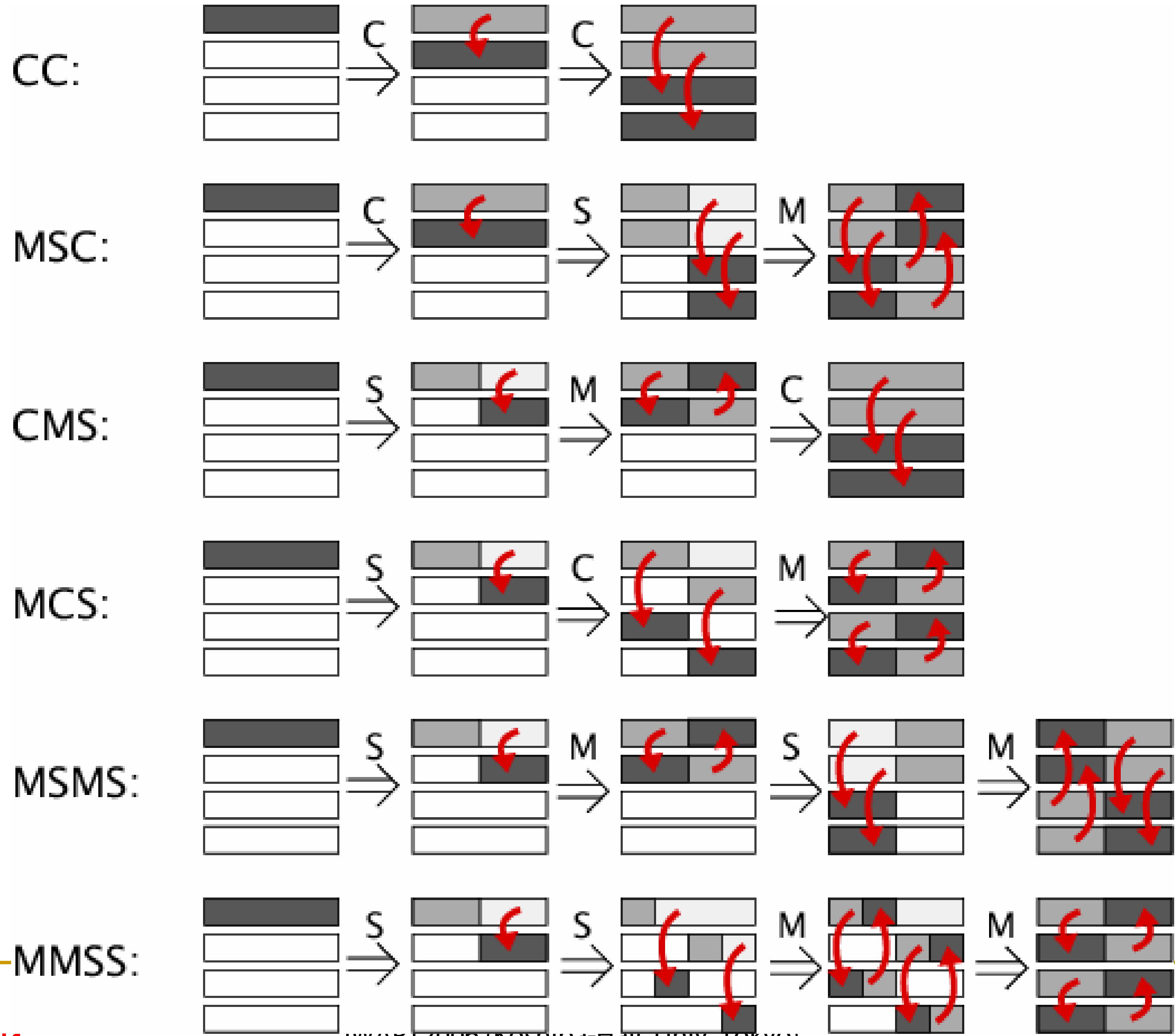
   $$\text{Bcast} = T^{(p)} F^{(p)}$$

   $$\begin{cases} F^{(i)} := C^{(i)} G_1 F^{(i-1)}, & T^{(i)} := G_2 \\ F^{(i)} := S^{(i)} G_1 F^{(i-1)}, & T^{(i)} := G_2 M^{(i)} \end{cases}$$

Bcast (broadcast)

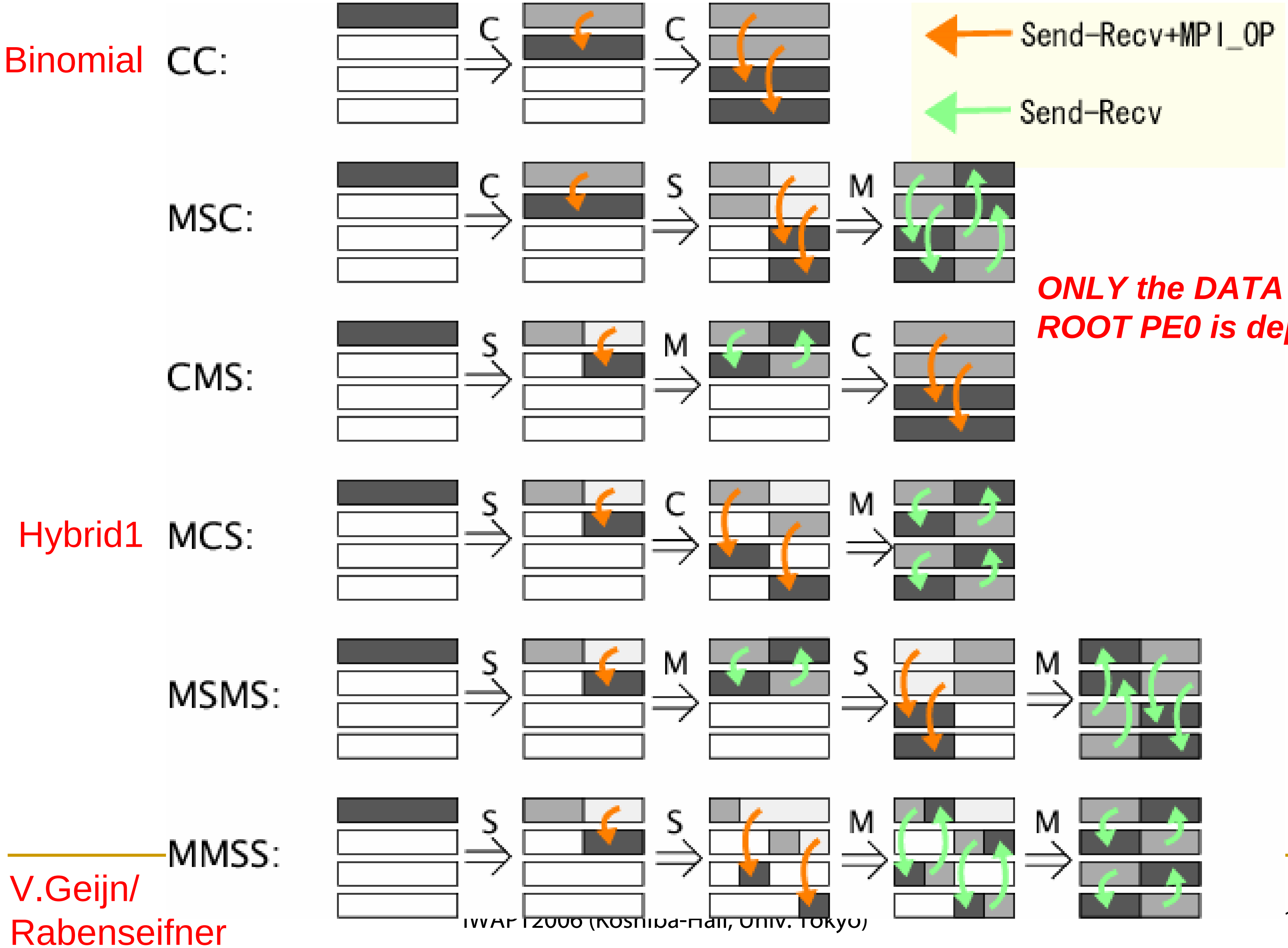
Binomial

Hybrid1



V.Geijn/
Rabenseifner

All_Reduce (all reduction)



ONLY the DATA from ROOT PE0 is depicted.

Optimization

- From the viewpoint of AT

Given (MPI_OP, size, group of PE),

"determine a better algorithm (=a combination of the primitives C,S,M) and a send-recv mechanism"

- Problem: the total number of Hybrid2 algorithm grows exponentially!

  Example：$N_{binary}(4)=6$, $N_{binary}(8)=22$, $N_{binary}(16)=90$, ….

[2-step approach]

⇒ i) Sieving the parameter space（Before EX）

＋ii) {Algorithm+Send-Recv Mechanism}（EX）

Fixed model or dynamic (feedback) model

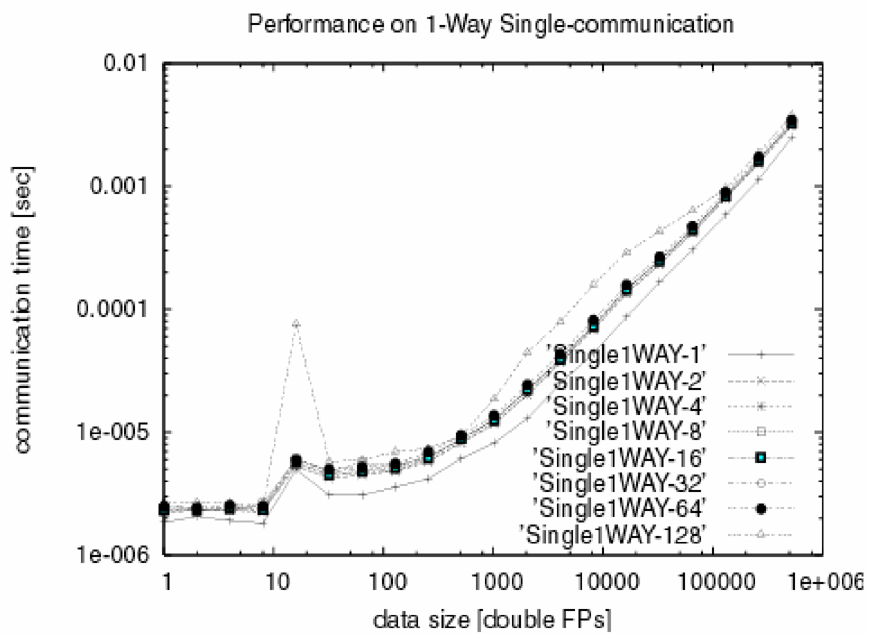
Optimization

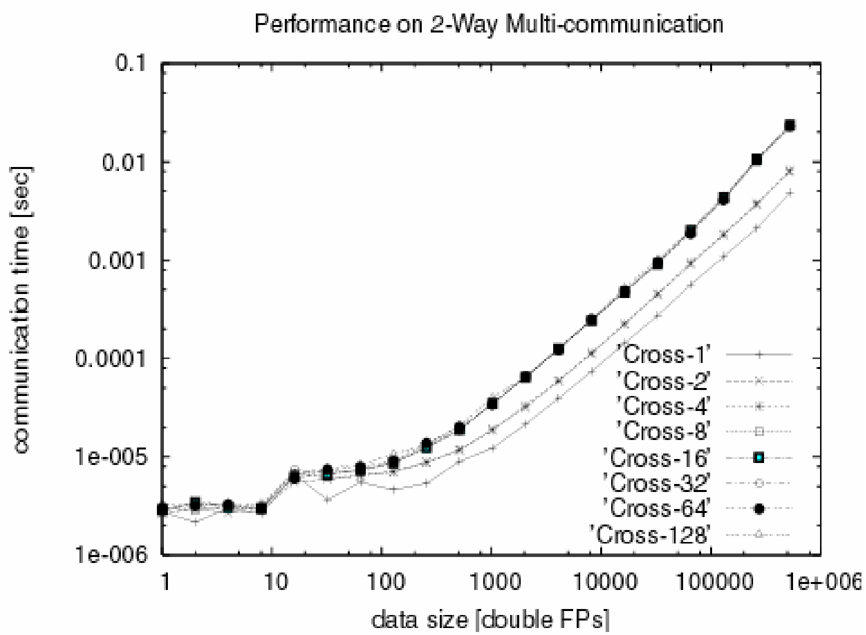
- Estimation with a Communication model

  $T=\alpha+\beta N$

  Coeffs. must be determined on all the possibles.

  {PE} × {1way, 2way, mutiple}





Optimization

- Estimation with a Communication model

  $T=\alpha+\beta N$

  Coeffs. must be determined on all the possibles.
  {PE}×{1way, 2way, mutiple}
- Rank ordering

  Possible ordering is P! too much. For example
  - NFFL : Near-First Far-Last
  - FFNL : Far-First Near-Last

⇒ Algorithms+Communication mechanics+Rank order × AT(before EX=static EVA + EX=dynamic EVA) ≒ Best Coll. func.

Related works

- Automatic tuning for MPI collective functions
  1. ACCT by Fagg, Vadhiyar / Faraj, Yuan
     - 2phase tuning approach
       - Rough sampling
       - Exact sampling
  2. Wu / Thakur&Rabenseifner
     - To Explore the parameter space
       {Binary,Binomial, …}*{pipeline, shmem,…}*…
  3. Fagg et al. (2006)
     - Rule-base approach, dynamical feedback, noisy networks
       {Eager,Pairwise} for ALLtoALLv with uncertain V patterns.
     - Decision quadtree

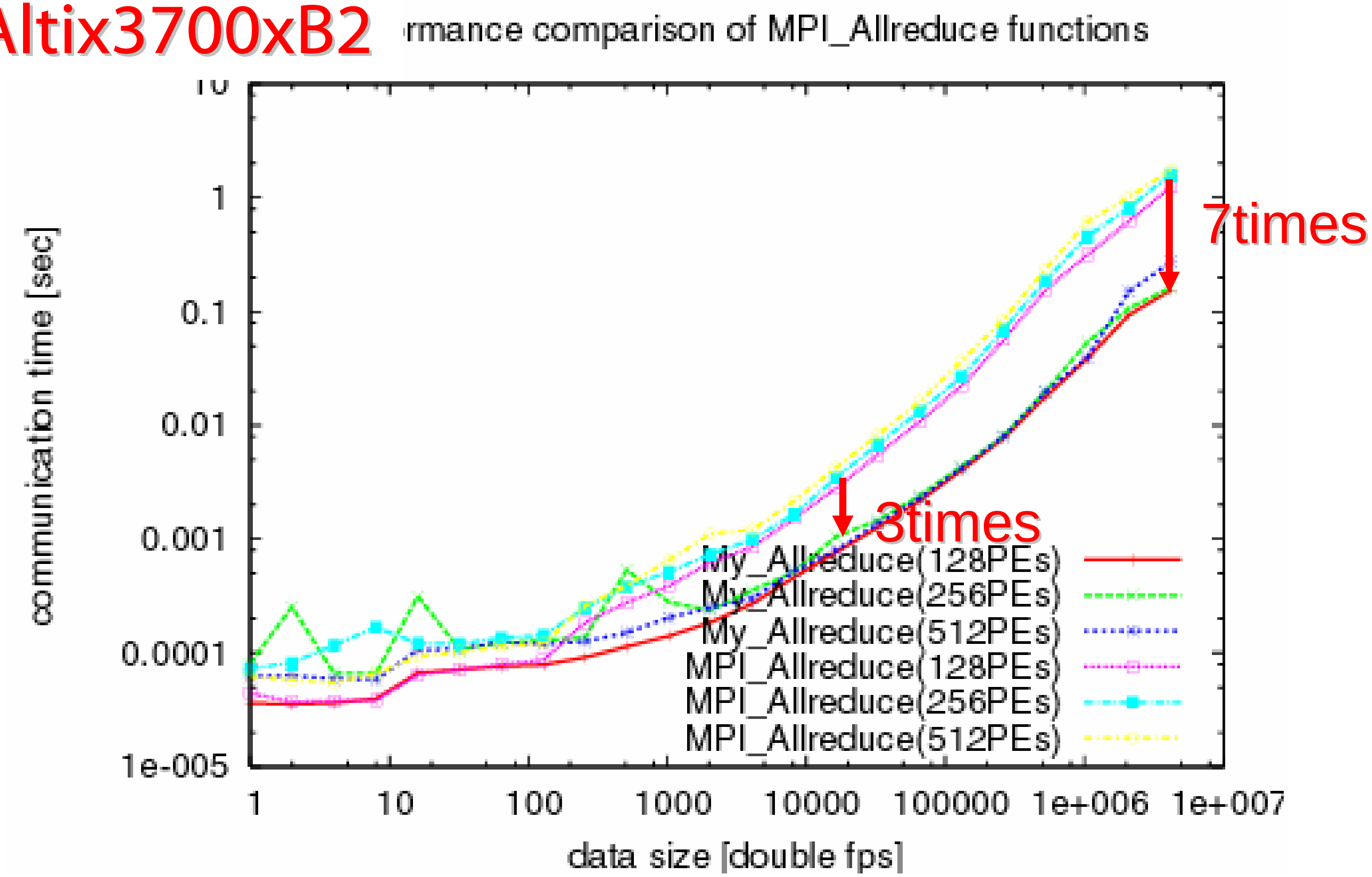
Evaluations

MPI_Bcast / MPI_Allreduce

On an Altix3700xB2 at CCSE JAEA

Allreduce

- Altix3700xB2



Householder with tuned-coll.funcs.

- Replace broadcast and allreduce by tuned ones

Tab.1: Eigensolver with the vendor-tuned MPI on Altix3700Bx2 32PEs ([sec], ()=%)

	N=1K	N=4K	N=6K
Total	**.187**	**1.745**	**4.303**
Bcast	.030(13)	.140(8.0)	.247(5.7)
Allreduce	**.081(43)**	**.561(32)**	**1.158(26)**
Re-dist.	.021(11)	.218(12)	.442(10)

-14%

Tab. 2: With tuned coll.funcs. (conditions are the same as above)

	N=1K	N=4K	N=6K
Total	**.180**	**1.519**	**3.709**
Bcast	.029(16)	.183(12)	.333(8.9)
Allreduce	**.074(41)**	**.426(26)**	**.836(23)**
Re-dist.	.021(11)	.205(13)	.410(11)

Householder with tuned-coll.funcs.

- Replace broadcast and allreduce by tuned ones

Tab.3: Eigensolver with the vendor-tuned MPI on Altix3700Bx2 128PEs ([sec], ()=%)

	N=4K	N=8K	N=12K
Total	**1.589**	**5.194**	**11.83**
Bcast	.207(13)	.533(10)	.981(8.3)
Allreduce	**.729(46)**	**2.171(42)**	**4.114(35)**
Re-dist.	.311(20)	.843(16)	1.914(16)

-18%

Tab.4: With tuned coll.funcs. (conditions are the same as above)

	N=4K	N=8K	N=12K
Total	**1.298**	**4.114**	**9.727**
Bcast	.193(15)	.509(12)	.958(9.8)
Allreduce	**.507(39)**	**1.274(31)**	**2.793(29)**
Re-dist.	.308(24)	.972(23)	1.713(18)

Householder with tuned-coll.funcs.

- Replace broadcast and allreduce by tuned ones

Tab.5: Eigensolver with the vendor-tuned MPI on Altix3700Bx2 256PEs ([sec], ()=%)

	N=4K	N=8K	N=10K
Total	**1.184**	**6.344**	**8.960**
Bcast	.355(30)	1.252(20)	1.596(18)
Allreduce	**.665(56)**	**3.133(49)**	**4.707(52)**
Re-dist.	--	--	--

-35%

Tab.6: With tuned coll.funcs. (conditions are the same as above)

	N=4K	N=8K	N=10K
Total	**1.109**	**4.341**	**5.804**
Bcast	.340(31)	.807(19)	1.090(19)
Allreduce	**.610(55)**	**1.863(43)**	**1.891(33)**
Re-dist.	--	--	--

Back-transform

- Vendor version vs. our-tuned version

Tab.7: Eigensolver with the vendor-tuned MPI on Altix3700Bx2 ([sec], ()=%)

	N=4K	N=8K	N=12K
Total (P=32)	**1.556**	**9.310**	**--**
Bcast (P=32)	.299(19)	1.771(19)	--
Total (P=128)	**1.154**	**5.287**	**14.095**
Bcast (P=128)	.641(55)	2.477(46)	5.603(40)

Tab.8: With tuned coll.funcs. (conditions are the same as above)

	N=4K	N=8K	N=12K
Total (P=32)	**1.373**	**8.434**	**--**
IBcast (P=32)	.152(11)	.309(3.6)	--
Total (P=128)	**.935**	**3.973**	**10.911**
IBcast (P=128)	.403(43)	.928(23)	1.533(14)

-22%

Conclusion

Finally (i)

- Simple redefinition of Binary algorithms (=complex is O(log2 P)) by introducing primitive Ops.
  - algorithm-explorsion

    $N_{binary}(4)=6$, $N_{binary}(8)=22$, $N_{binary}(16)=90$, ….
- Auto-tuning:
  - Exploring the parameter space (manually|automatically)

    {**combinations of C,S,M**} ∗ {**send-recv mechanism**} ∗ {**Ranking**}
  - Static performance modeling and Estimation + Dynamic algorithm selection
    - *7 times faster than the vendors on MPI_Allreduce*
    - *On long messages, big benefit of AT collective routines is obtained from14 to 35%!*

Finally (ii)

- Future-work
  - Other Collective operations

    (All-)gather, (All-)scatter, Alltoall
  - Noisy or un-symmetric network

    Global definition -> local {C,S,M} management

    How (who) organize the algorithm?
  - Dynamic feedback to the performance modeling

    to predicate parameters

    to reduce data-noise and fluctuations from the hardware environment.

Thank you …

for your patient. Any Question?