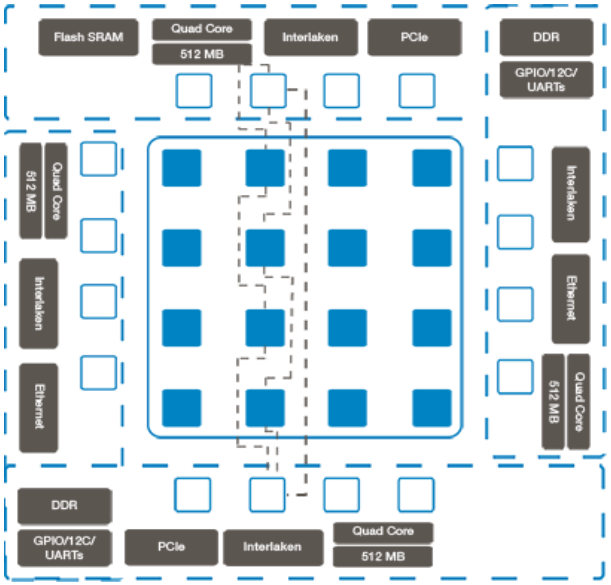
UNIVERSITÉ GRENOBLE ALPES

Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

# Data optimization for linear algebra and stencil compact on Many-core processor




Kalray MPPA-256

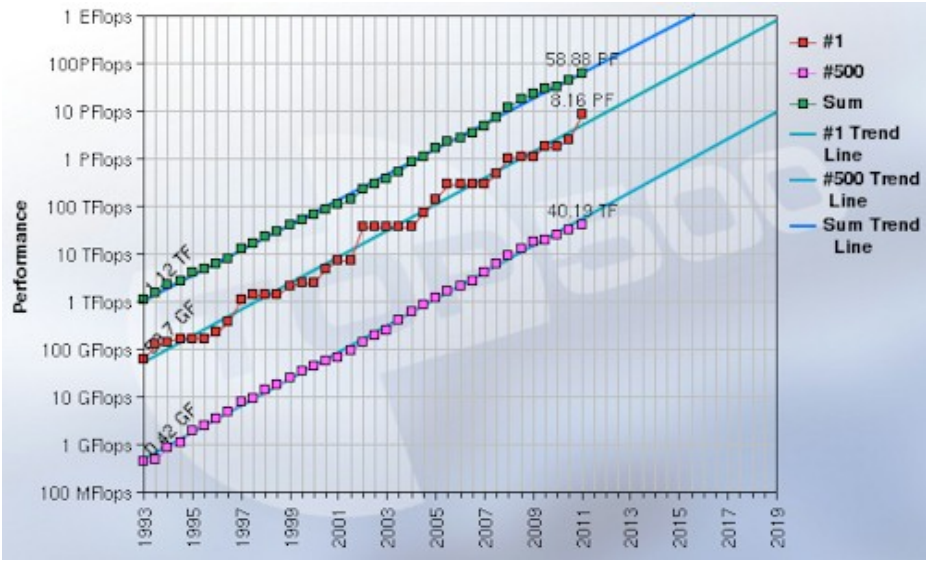

Etudiant : Minh Quan HO
Equipe : DRAKKAR
Directeur de thèse : Bernard TOURANCHEAU
Co-encadrant : Christian OBRECHT

# Outline

- Context
- Applications
- Work done this year
- Conclusion and perspectives

# Context

- Exascale ($10^{18}$) by 2020
- Power =< 20 MW
- >= 50 Gflops/W

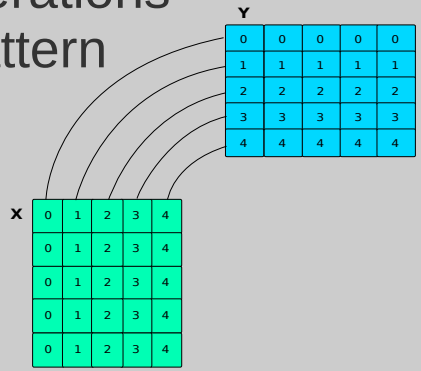



| | Intel Xeon E5-2687W | Intel Xeon Phi co-processor 31S1P | Tesla K20X | Kalray MPPA-256 | ? |
|---|---|---|---|---|---|
| Gflops/W (FP64) | 1.5 | 3.4 | 4.8 | 6.8 | >= 50 |
| Number of cores/node | 8 | 57 | 2688 | 256 | ? |
| Frequency (MHz) | 3100 | 1100 | 732 | 400 | ? |

# Applications

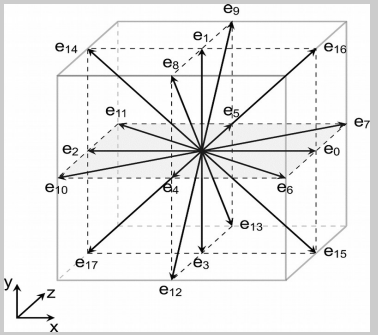
## Linear algebra (BLAS)

- Vector & Matrix operations
- Irregular access pattern




## Compact stencils PDE solvers (e.g Lattice Boltzmann Method)

- Access to neighbor nodes
- Non-contiguous memory access




- Cache coherence. How and Where ?
  - Hardware complexity & latency increases with # of cores
  - Incoherence-aware software design – Non-trivial
- Memory
  - Non-Uniform-Memory-Access (NUMA)
  - Difficult to do and maintain Global Memory + overhead
  - Bandwidth could hardly follow # of cores and clock
- Programming framework ?

# Work done

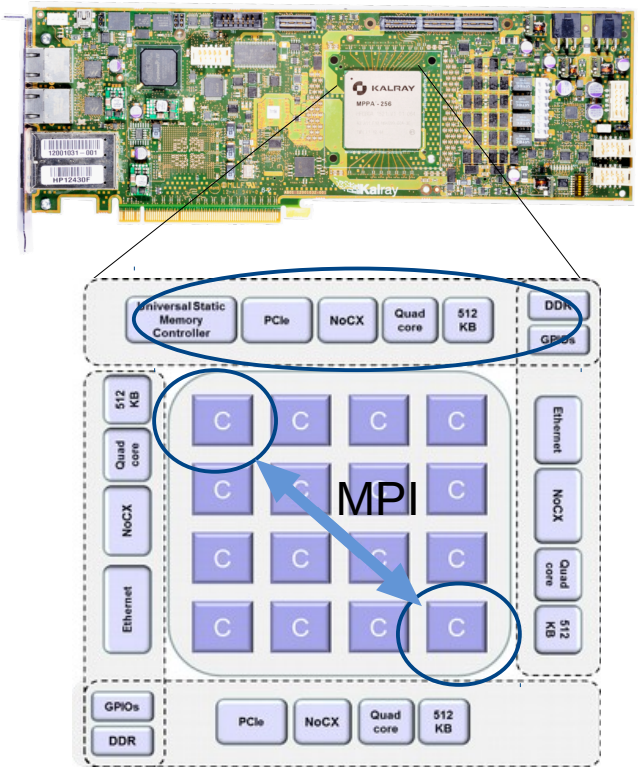
1. Many-core as Stand-alone node
   - MPI design and implementation
   - MPI optimizations comparison

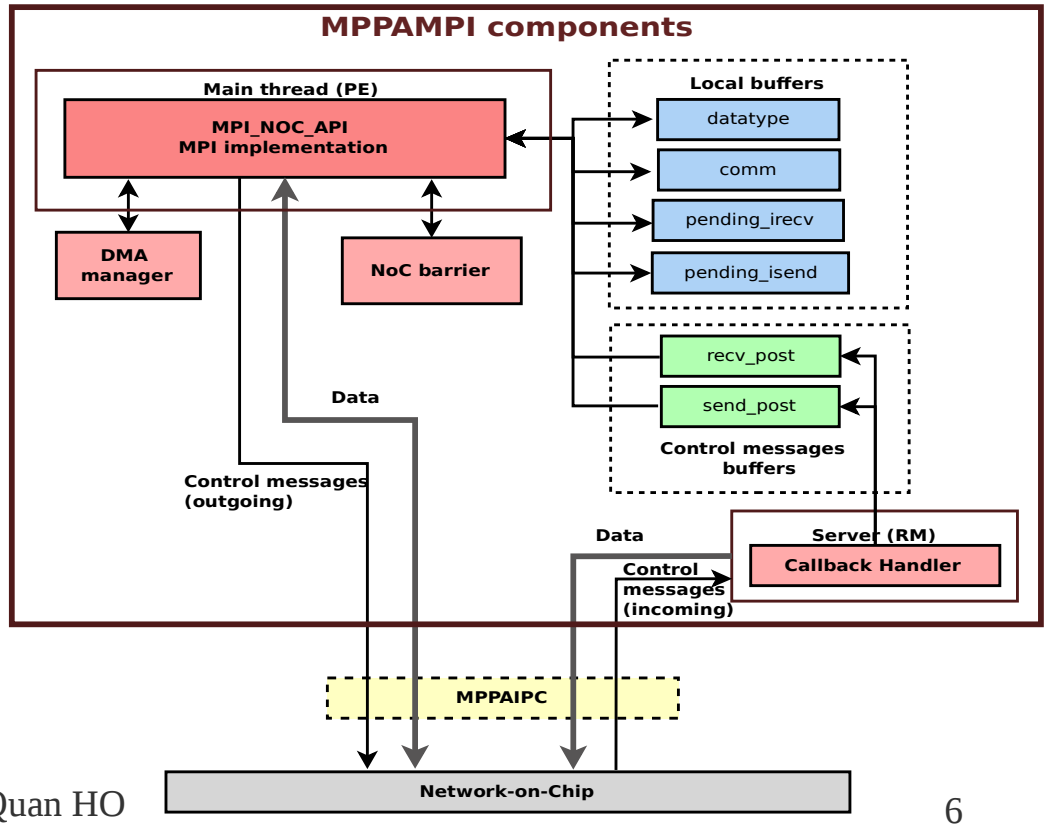
2. Many-core as Accelerator
   - clBLAS
   - clMAGMA

3. Benchmark
   - High Performance Linkpack (HPL)

# 1. Many-core as Stand-alone node

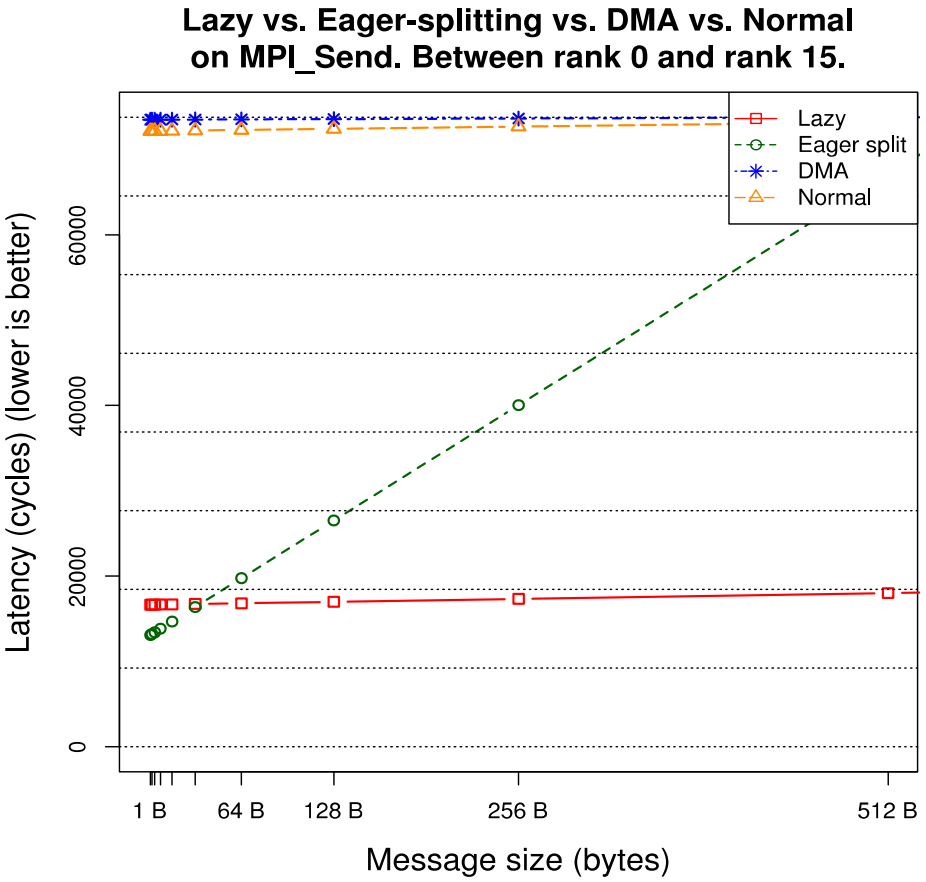
- Issues : Isolated cores, limited memory
- Design : 16 + 1 MPI processes per Many-core
  - MPI compute ranks : 16 x 2 MB on-chip memory
  - MPI I/O rank : 1 x 2 GB DDR

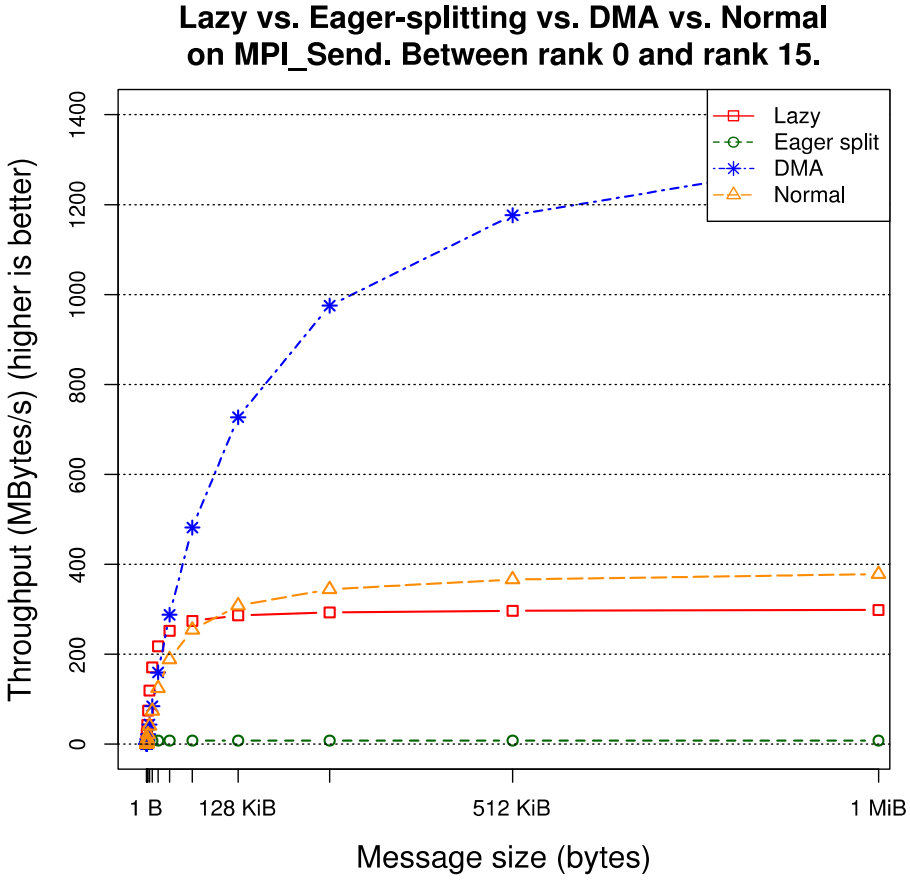


# 1. Many-core as Stand-alone node

- Optimizations :
  - Eager send for short messages
  - Lazy send for medium messages

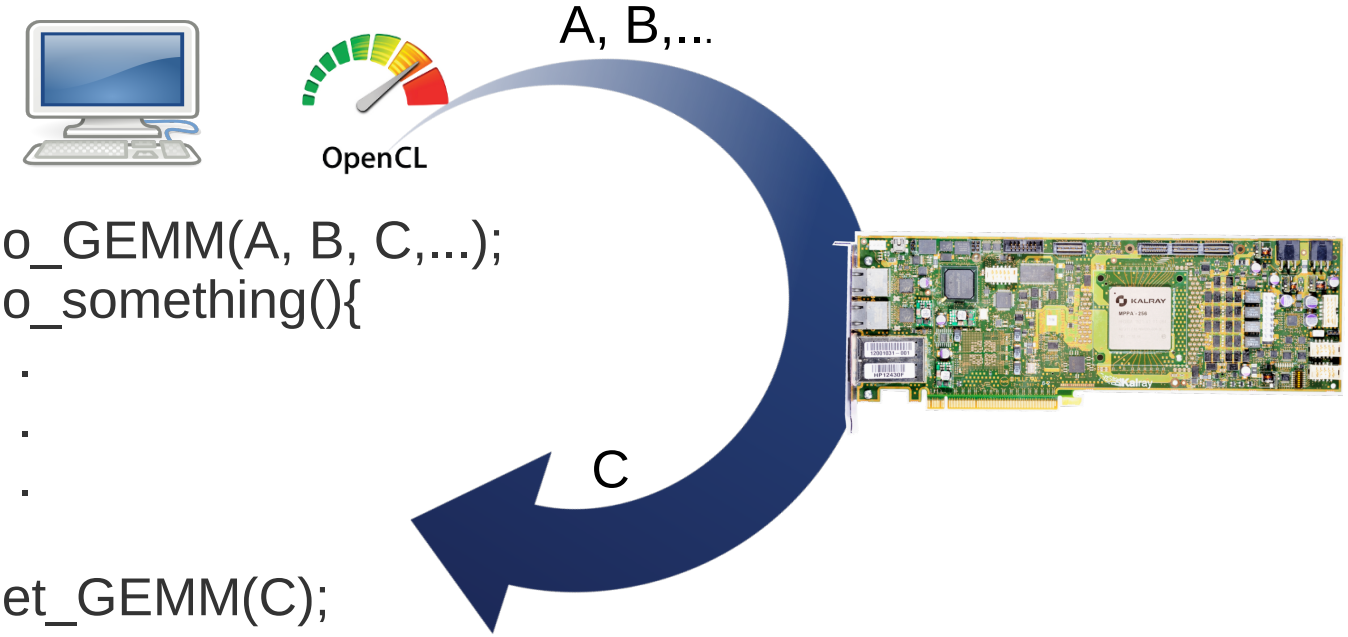


# 2. Many-core as Accelerator

## CLBLAS

- Offloading computation from Host to Many-core
  - Static kernels for level 1-2
  - Dynamic code generation for level 3
- 2.6 Gflops (SGEMM) – 1.3 Gflops (DGEMM)

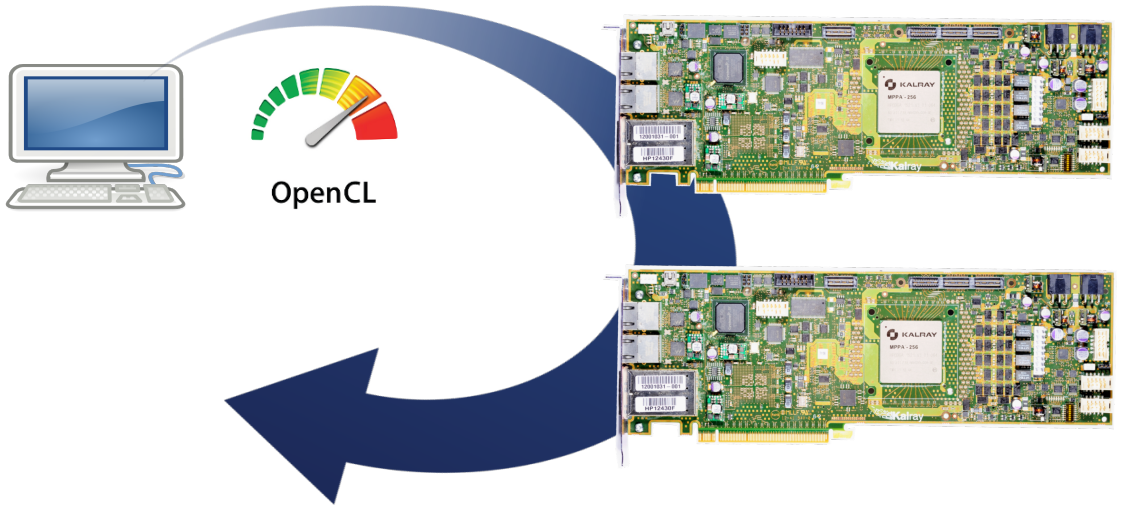



```
do_GEMM(A, B, C,...);
do_something(){
    .
    .
    .
}
get_GEMM(C);
```

# 2. Many-core as Accelerator

## CLMAGMA

- LAPACK-based interface (QR, LU, CHOL … )
- Auxiliary kernels written in OpenCL
- Use clBLAS to offload computation on device(s)



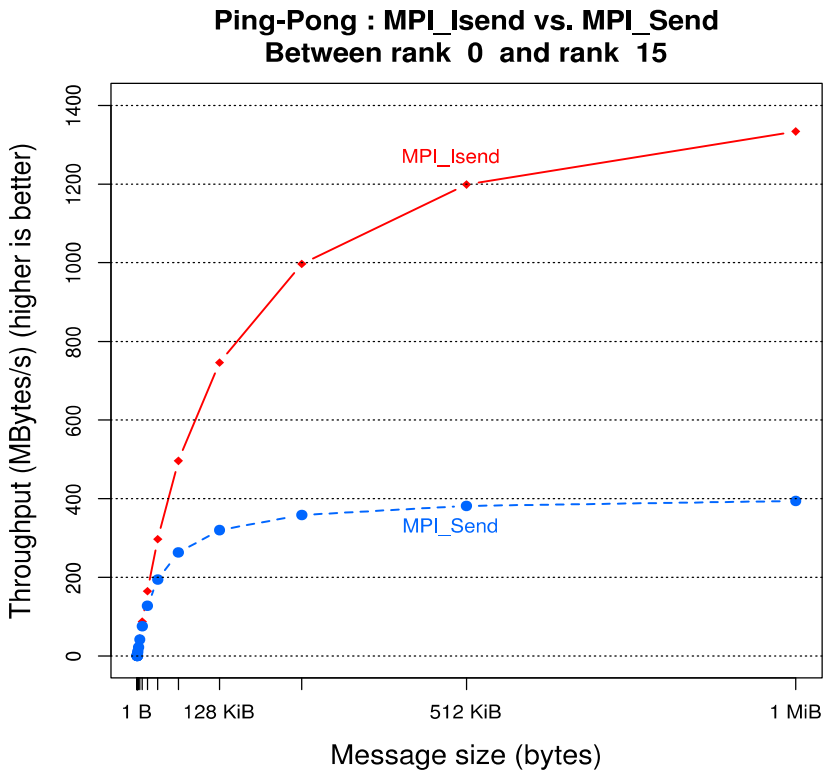
# 2. Many-core as Accelerator

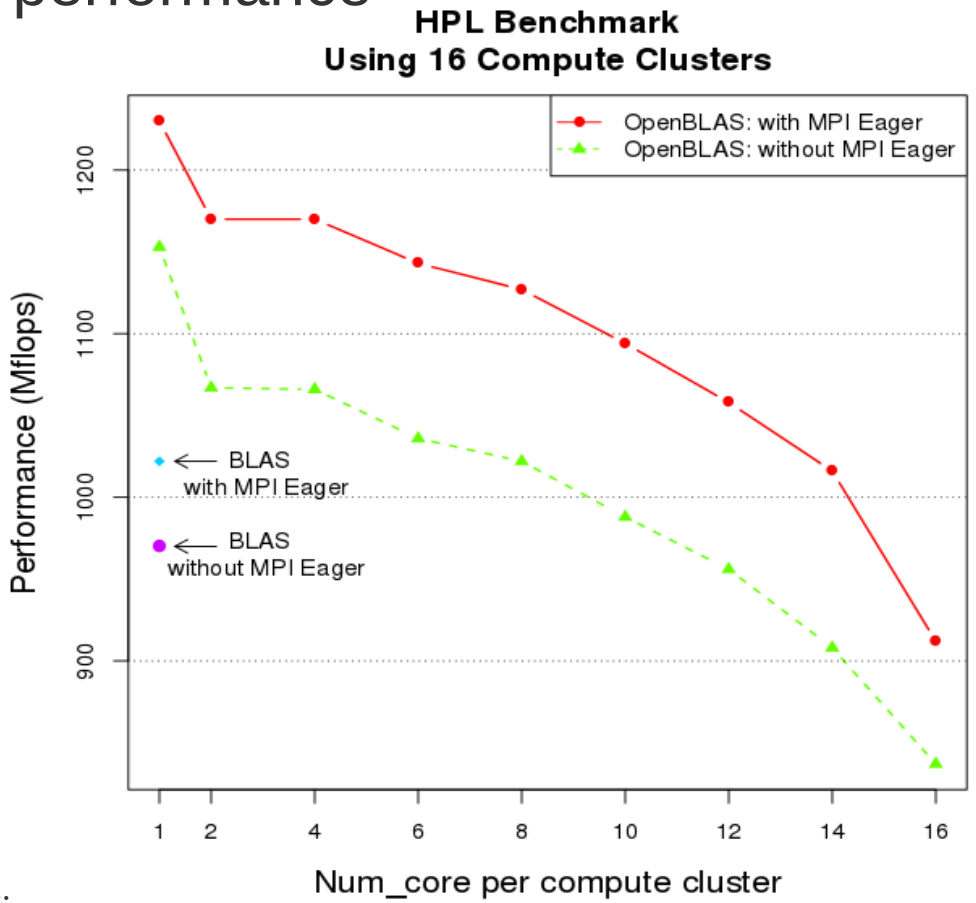
## CLBLAS / CLMAGMA issues

- Block decomposition too small compared to page-size
- Concurrent accesses to same memory block
- Kernels written for GPU, non-optimal for Many-core

# 3. Benchmark

- HPL benchmark with 16 MPI ranks, BLAS / OpenBLAS
  - Limited local memory space (16 x 2 MB)
  - Matrix size 1500 x 1500
  - MPI Eager send gains 10% performance



# Conclusion

- Many-core as Stand-alone node
  - MPI mono-Many-core, BLAS/OpenBLAS and HPL
  - Next :
    - MPI + DSM to increase memory space via I/O DDR + optimization modeling
    - MPI multi-Many-core for scalability
- Many-core as Accelerator
  - clBLAS + clMAGMA
  - Next :
    - Block decomposing tuning for Many-core
    - {Page|Cache}-size-aware kernels
    - LBM design & accelerating

[1] M. Q. Ho, B. Tourancheau, C. Obrecht *et al.*, “MPI communication on MPPA Many-core NoC: design, modeling and performance issues”, *PARCO*, Edinburgh, Scotland, UK, 1-4 September 2015.

[2] Kalray Inc., “ManycoreLabs report”, vol. 6, pp. 39-41, 2015

# Question ?