Taming Heterogeneity by Segregation

The DEEP view on Exascale

♦

Norbert Eicker

Jülich Supercomputing Centre & University of Wuppertal

The research leading to these results has received funding from the European Community's Seventh Framework Programme (*FP7/2007-2013*) under *Grant Agreements* n° 287530 and n° 610476

Outline

- Motivation
  - Why heterogeneous systems
  - How to organize heterogeneity
- DEEP
  - General concept
  - Hardware architecture
  - Programming paradigm
  - Software stack
  - (Few) Results
- Outlook on DEEP-ER
- Summary

DEEP face-to-face Leuven

DEEP-ER kickoff Jülich

Partners

Fraunhofer ITWM

CGG Passion for Geoscience

CINECA

The Cyprus Institute

EPFL École Polytechnique Fédérale de Lausanne

ASTRON

CERFACS

intel

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

EUROTECH Imagine. Build. Succeed.

German Research School for Simulation Sciences

Inria Inventors for the digital world

ParTec Cluster Competence Center

KU Leuven

Mellanox Technologies

Universität Regensburg

Seagate

Jülich Forschungszentrum

lrz

Ruprecht-Karls-Universität Heidelberg

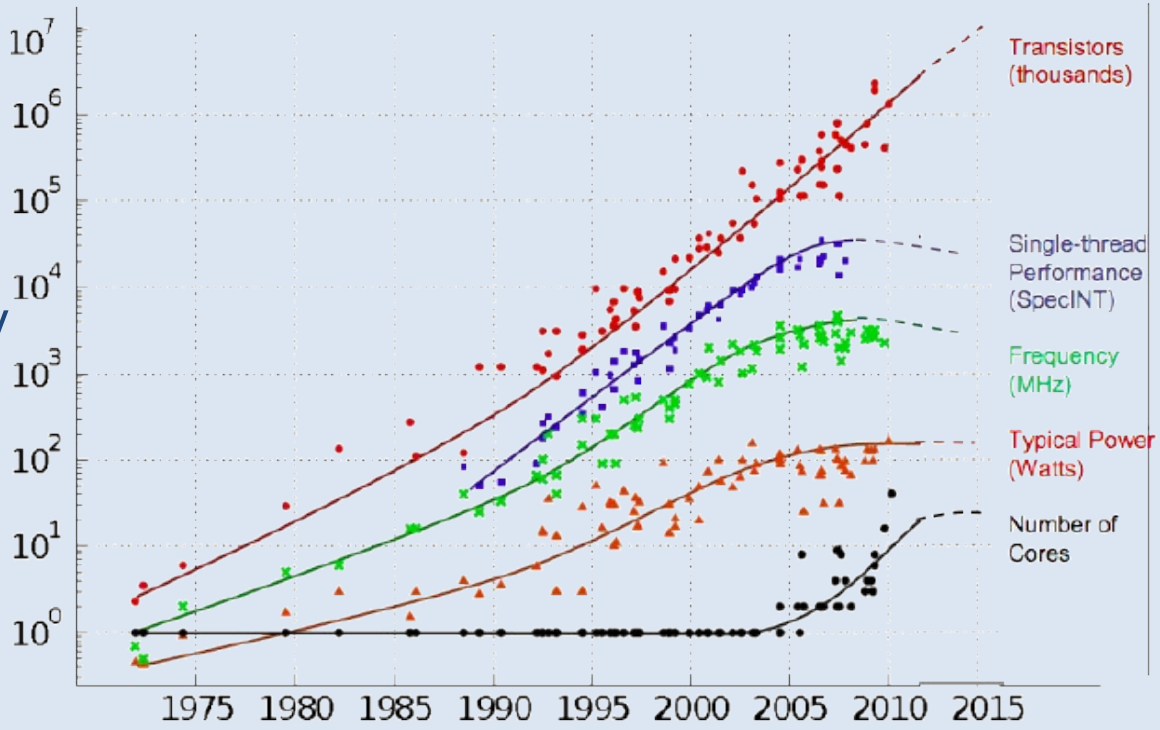
DEEP Project

- 16 Partners
  - 4 PRACE hosts
  - 3 Research Centers
  - 5 Industry Partners
  - 4 Universities
  - Coordinator JSC
- 8 Countries
- Duration: 3.5 years
- Budget: 18.3 M€
- EU funding: 8.03 M€

Heterogeneity

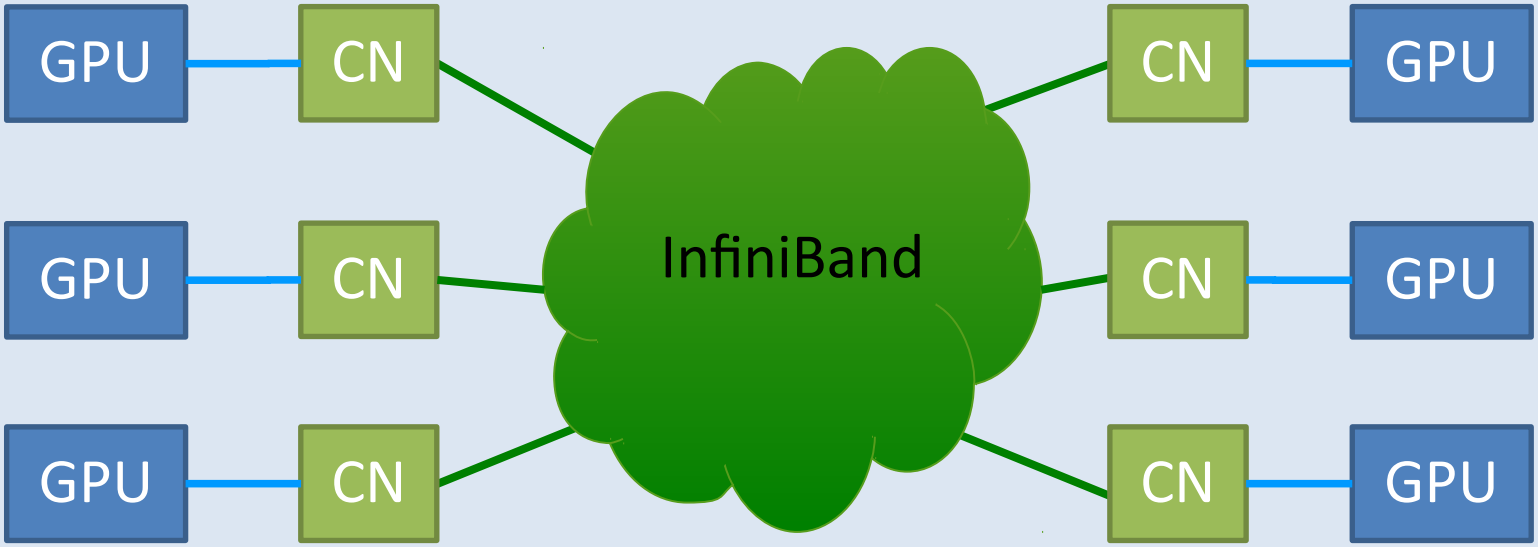
Why hetereogeneity – Moore's law

- Observation
  - Clock stagnates since 2002
  - Max. clock frequency at about 3 GHz
    - Few exceptions: Power6, Power 7 Gaming
  - # transistors still increases
- Current trends
  - Multi-Core/Many-Core processors
  - Simultaneous Multi Threading (SMT)



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

Heterogeneous Clusters



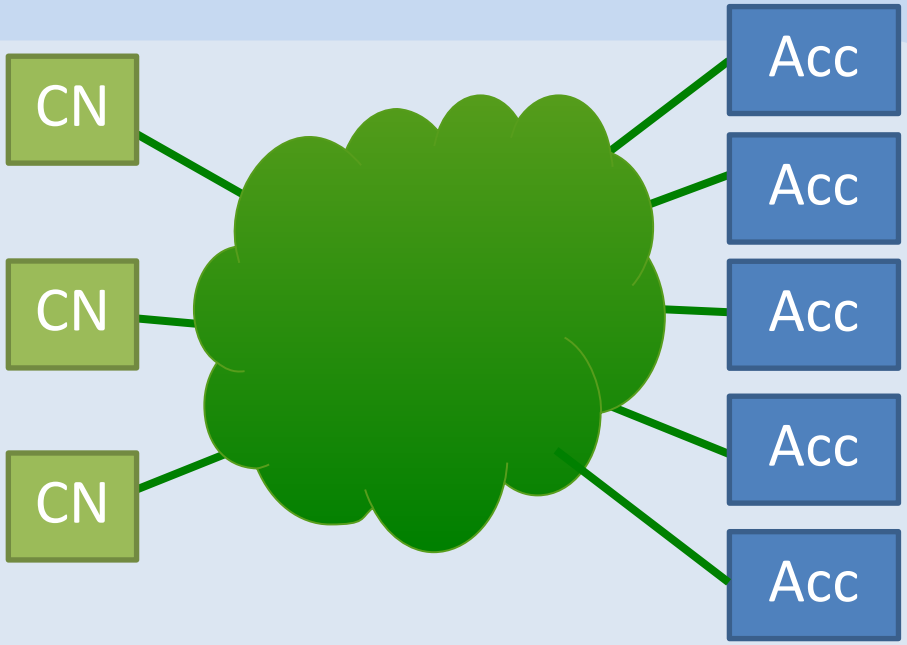
Flat IB-topology

Simple management of resources

Static assignment of CPUs to GPUs

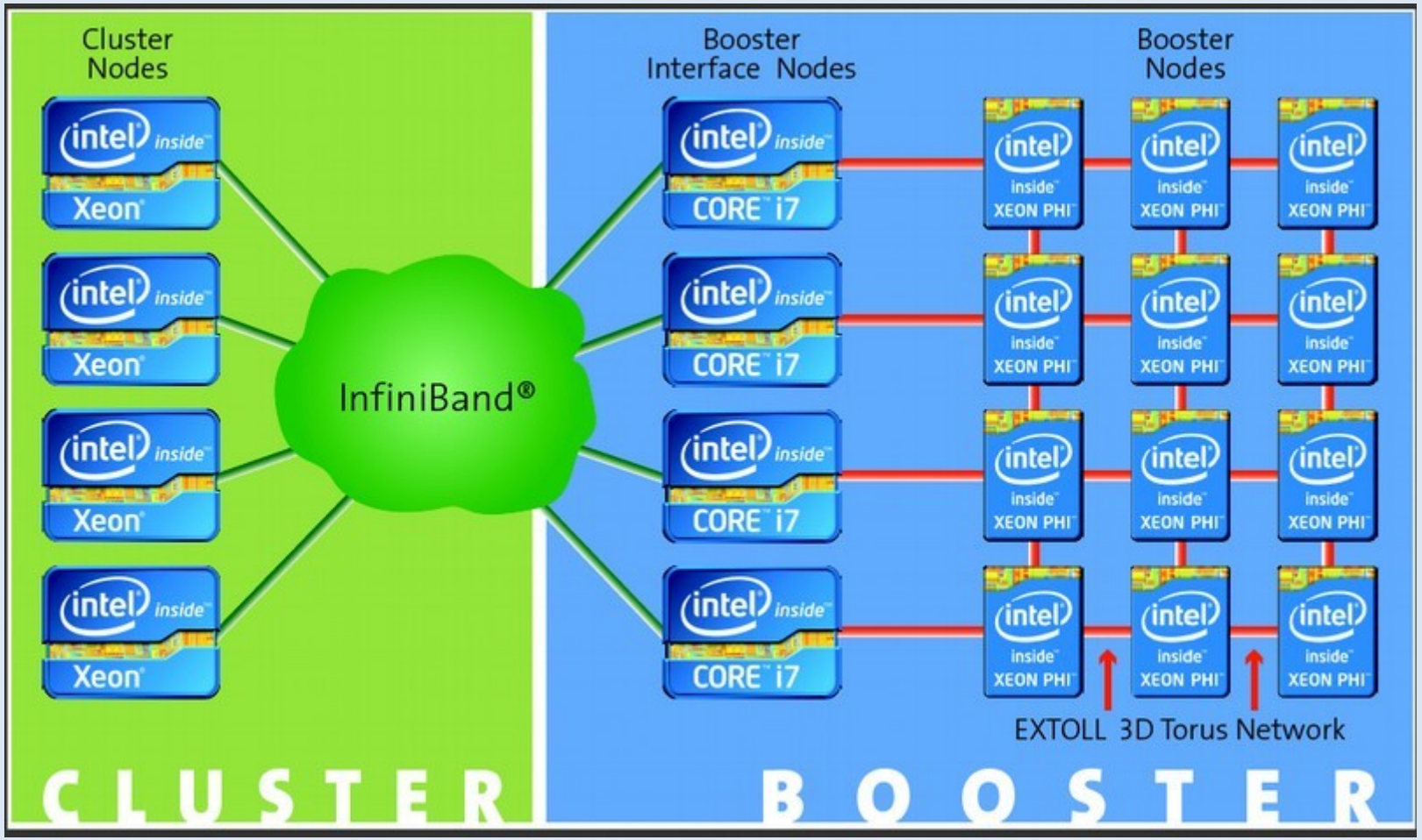
Accelerators not capable to act autonomously

Alternative Integration

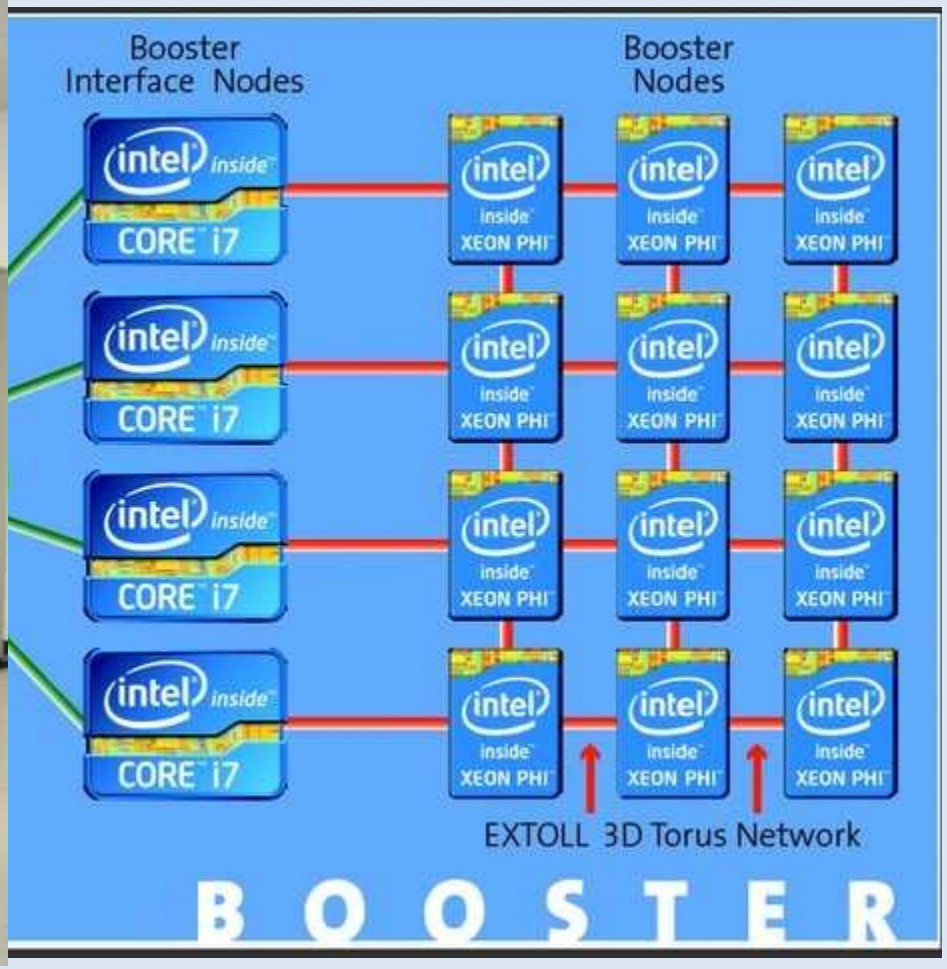


- Go for more capable accelerators (e.g. MIC)
- Attach all nodes to a low-latency fabric
- All nodes might act autonomously
- Dynamical assignment of cluster-nodes and accelerators
  - IB can be assumed as fast as PCIe besides latency
- Ability to off-load more complex (including parallel) kernels
  - communication between CPU and Accelerator less frequently
  - larger messages i.e. less sensitive to latency

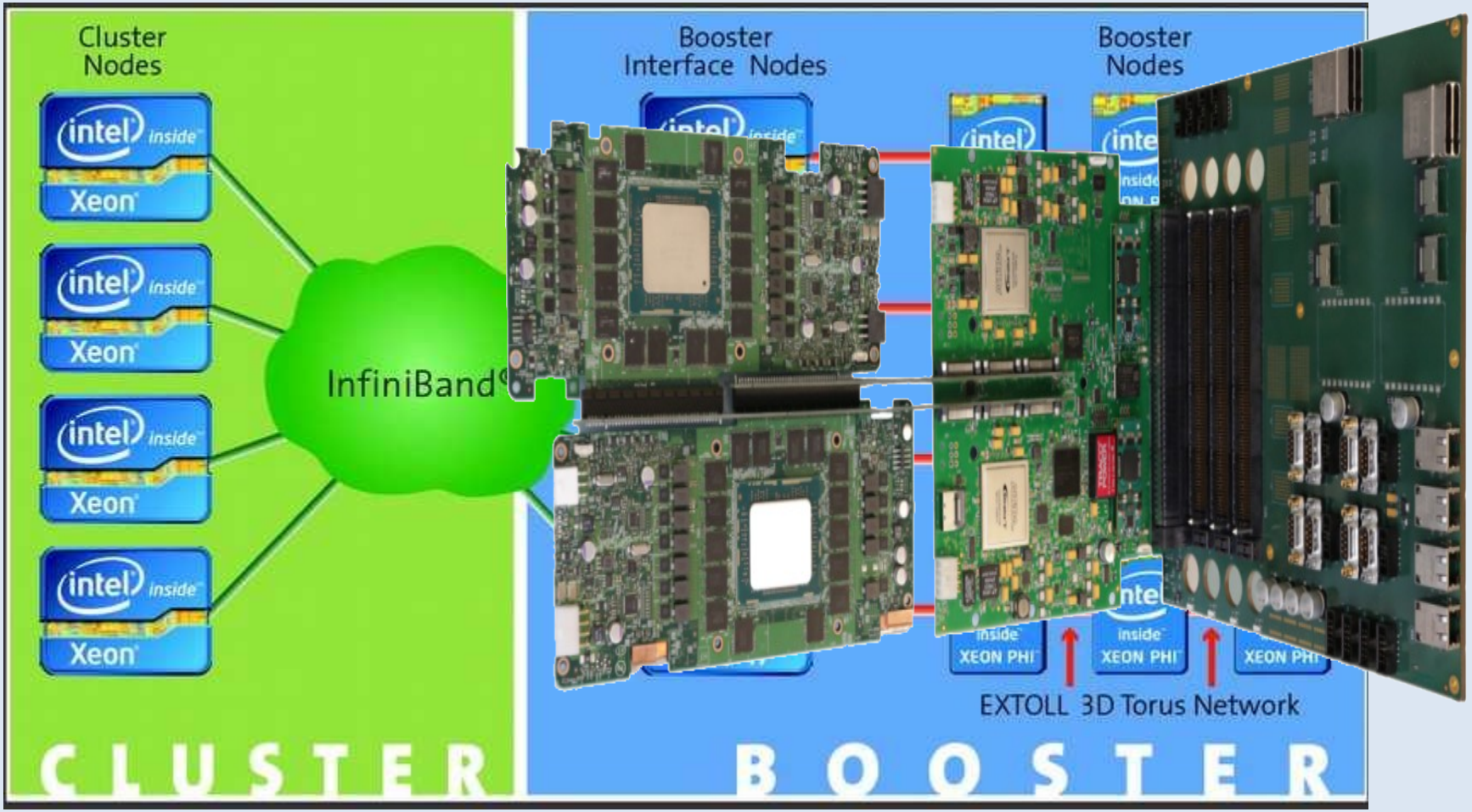
Hardware Architecture



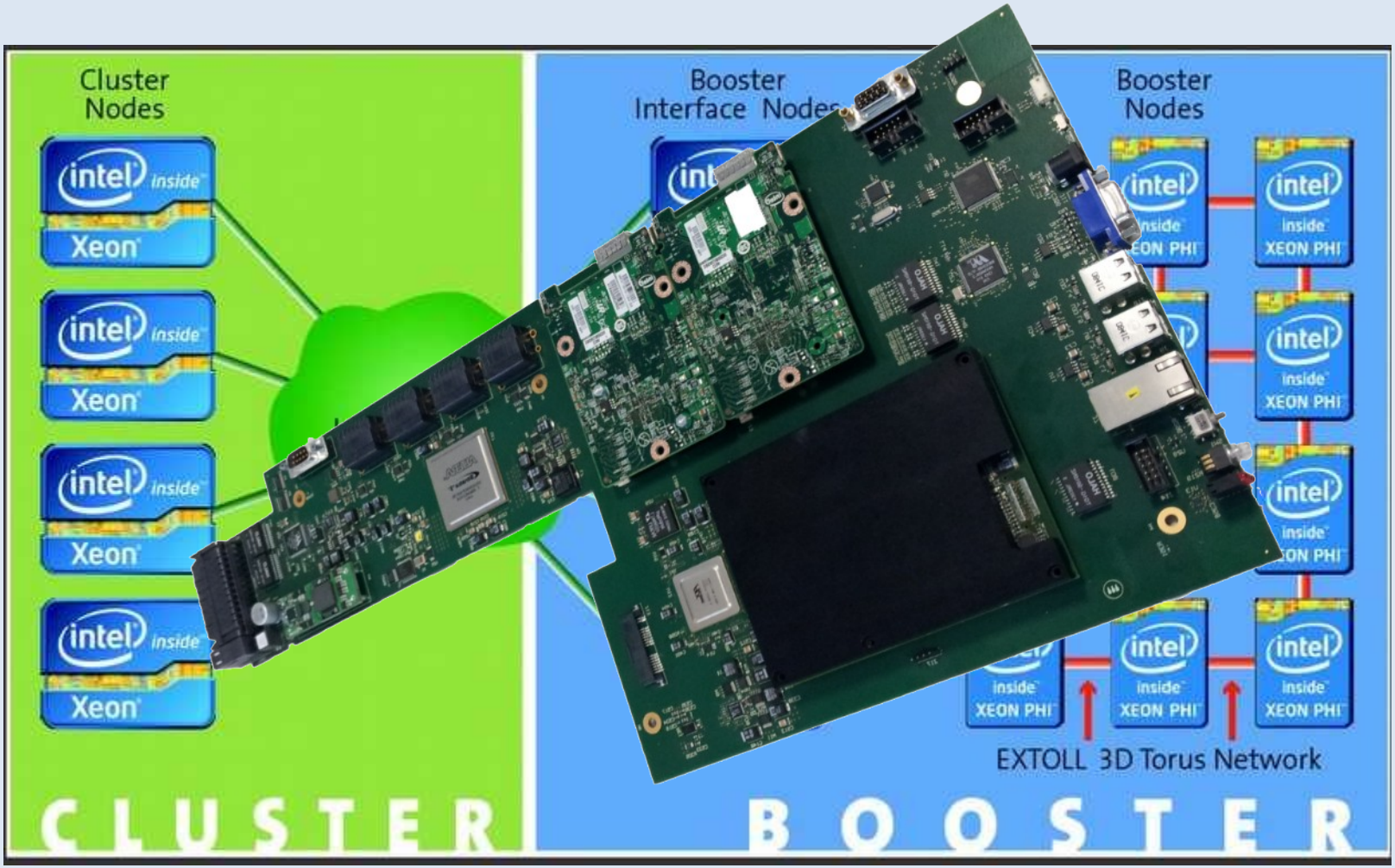
DEEP Hardware



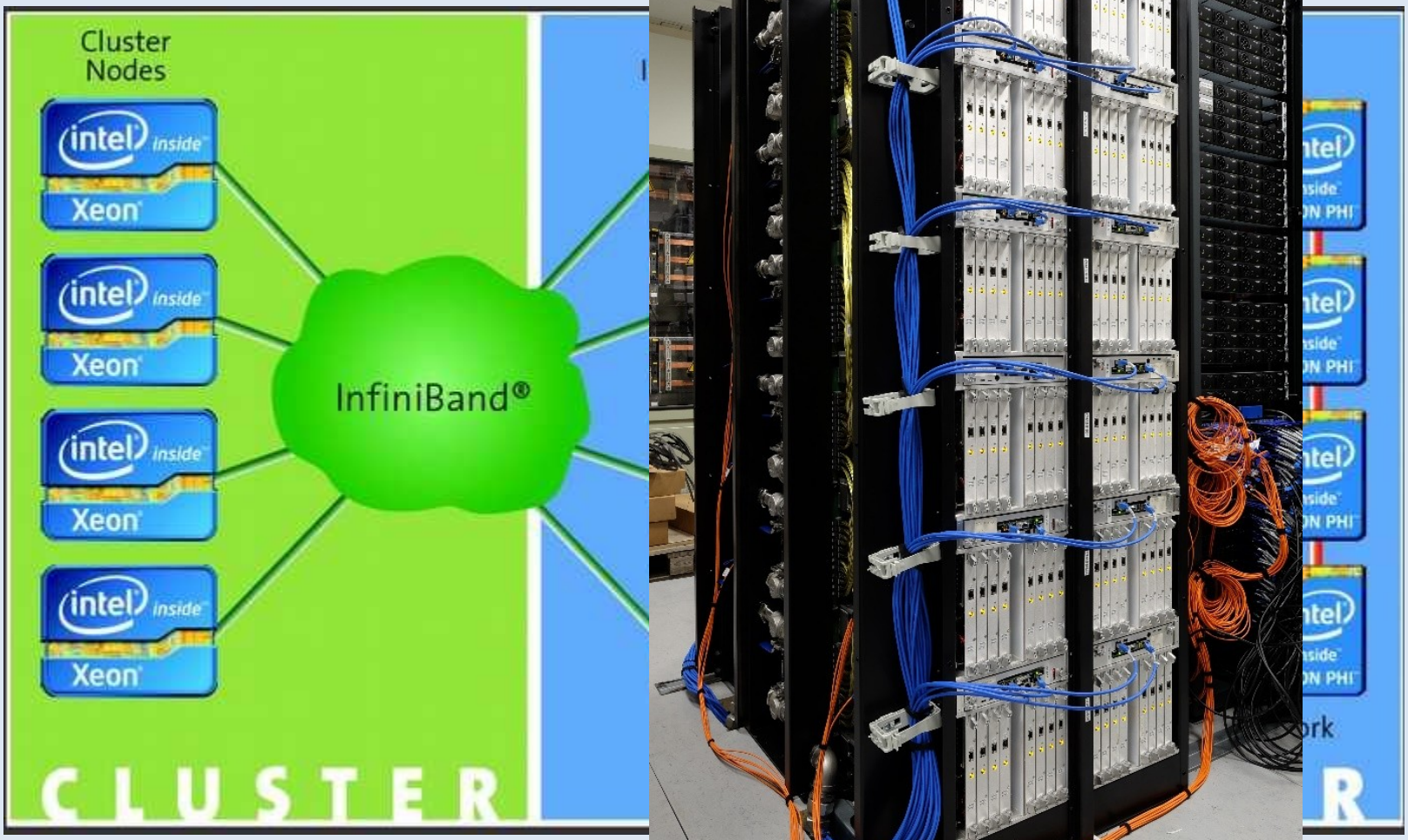
DEEP Hardware



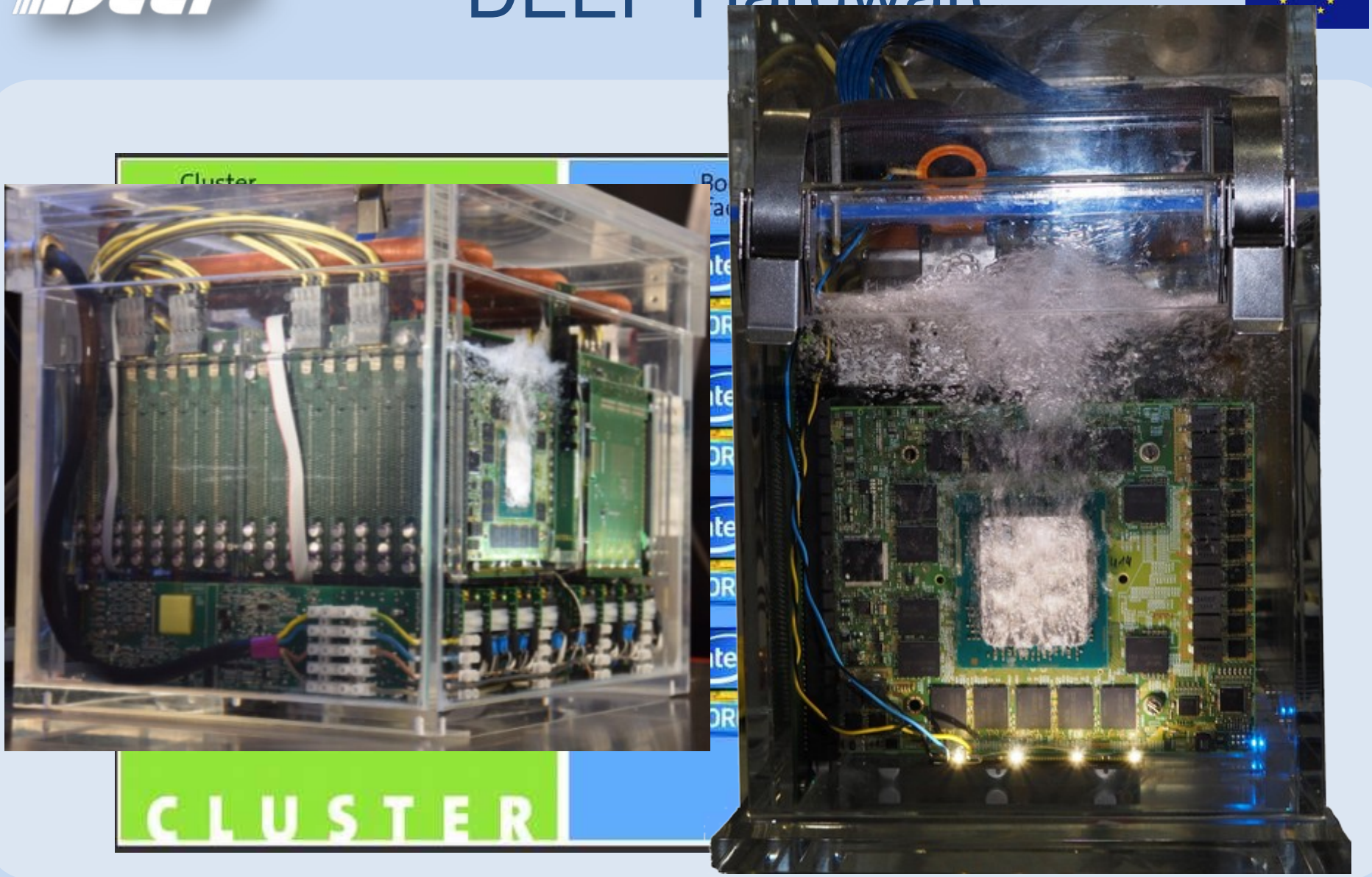
DEEP Hardware



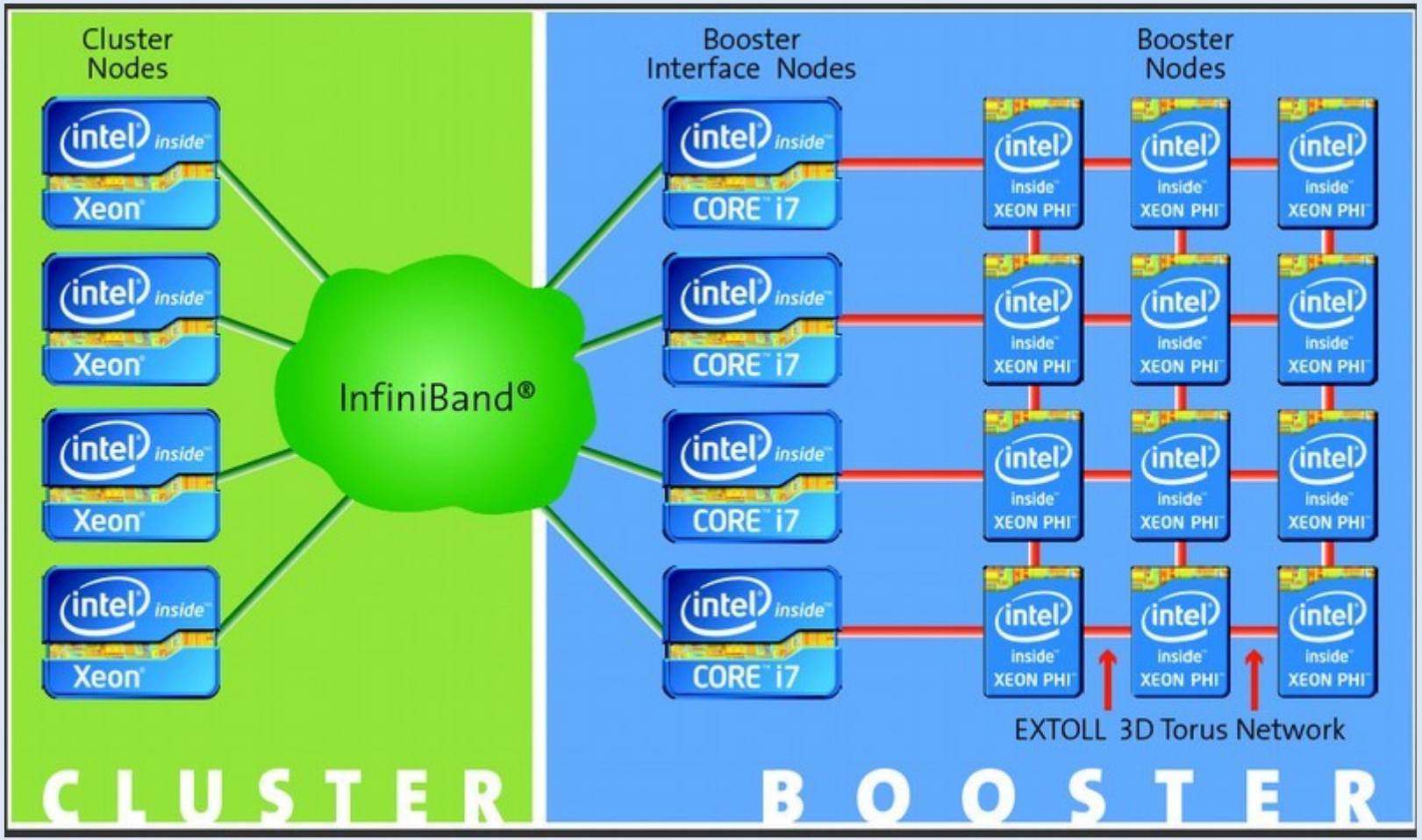
DEEP


Cluster
Nodes
intel inside
Xeon
intel inside
Xeon
intel inside
Xeon
intel inside
Xeon
InfiniBand®
CLUSTER

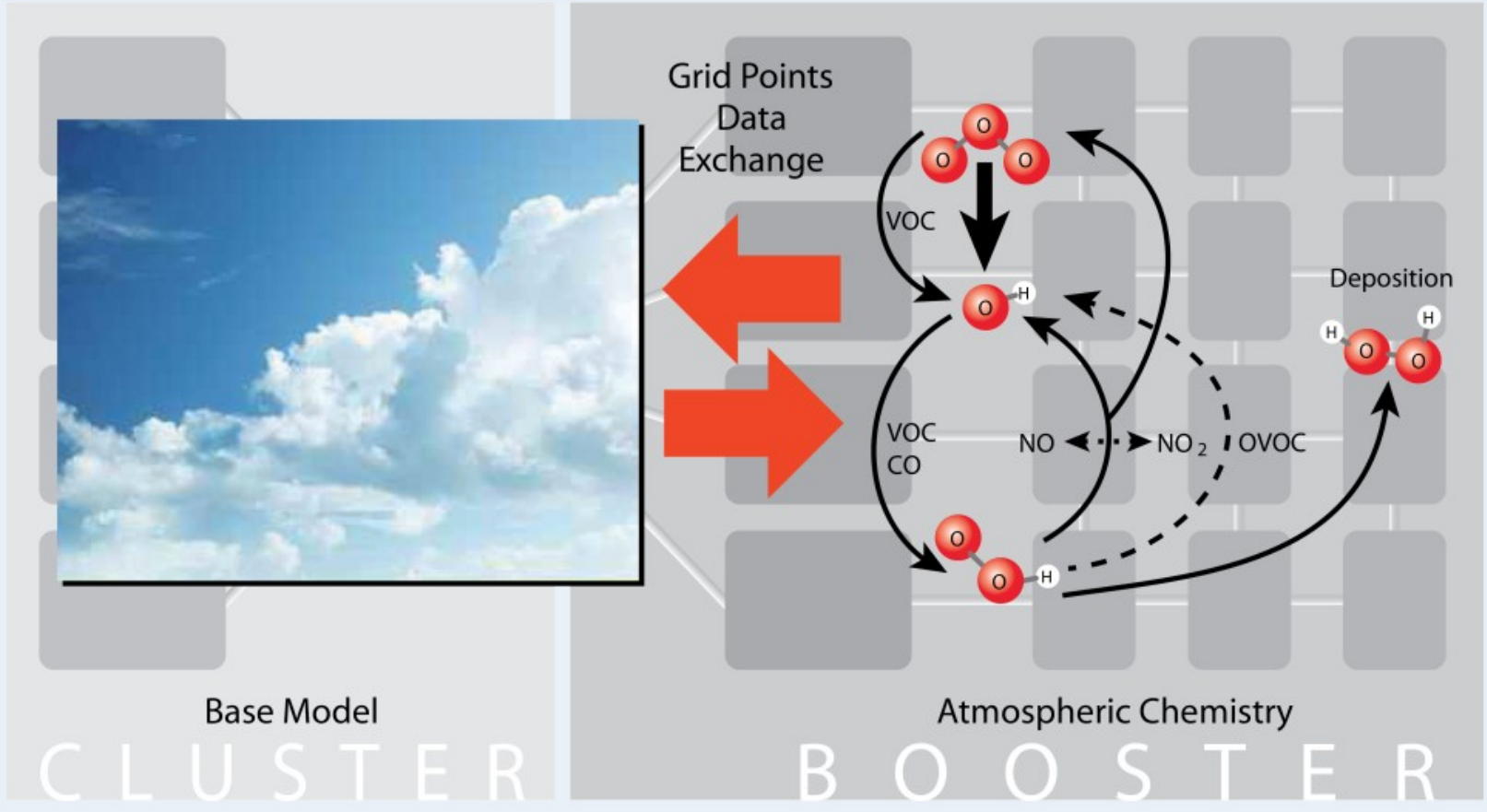
DEEP Hardware



Hardware Architecture

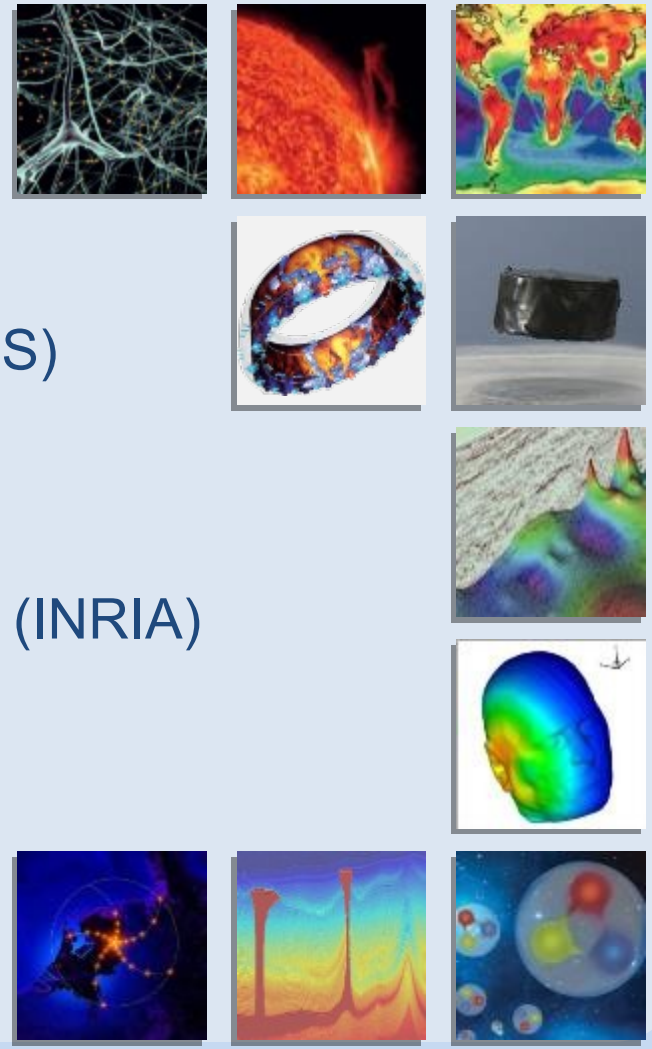


CyI – Climate Simulation



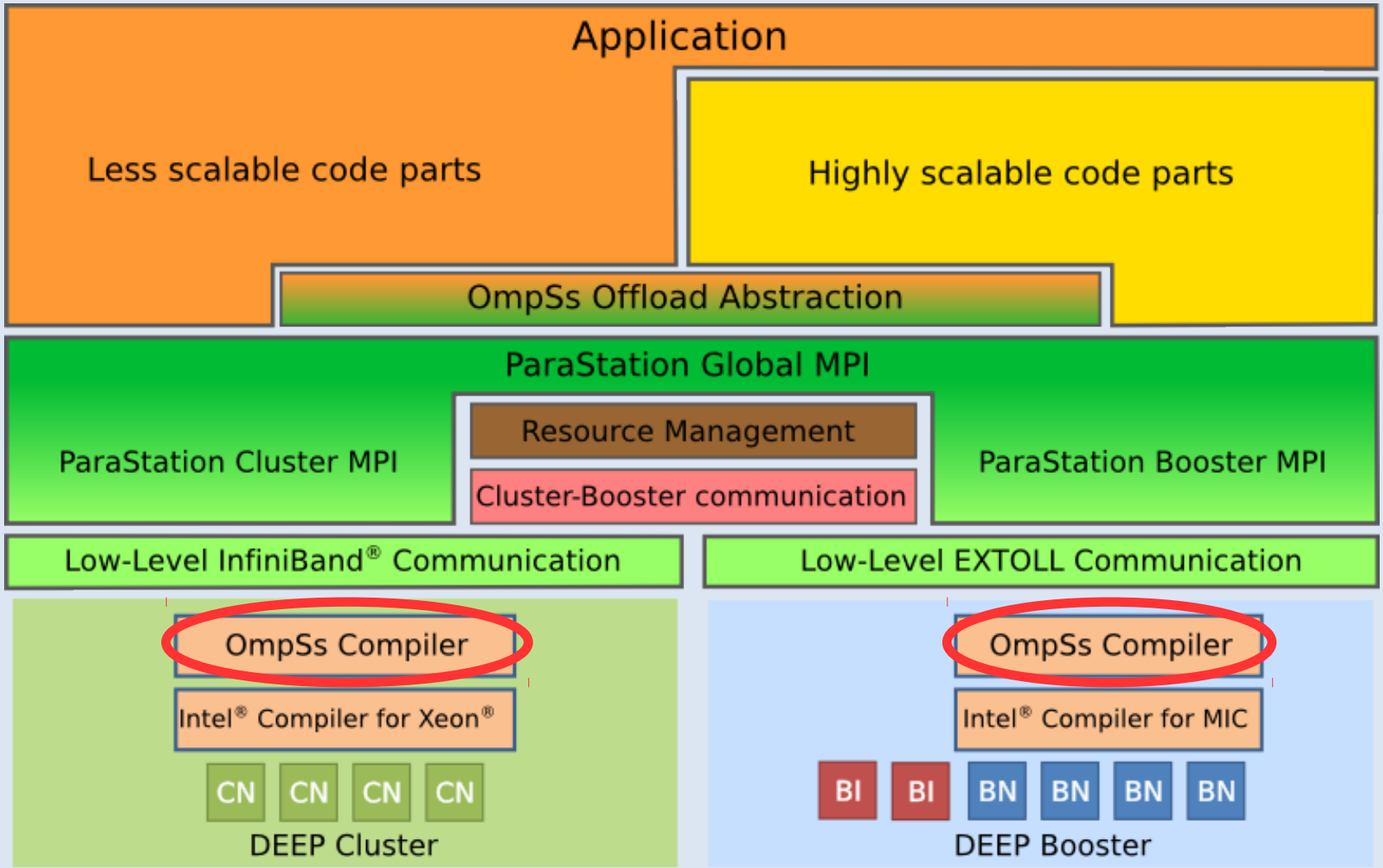
Guiding Applications

- Brain simulation (EPFL)
- Space weather simulation (KULeuven)
- Climate simulation (CYI)
- Computational fluid engineering (CERFACS)
- High T_c superconductivity (CINECA)
- Seismic imaging (CGG)
- Human exposure to electromagnetic fields (INRIA)
- Geoscience (BADW-LRZ)
- Radio astronomy (Astron)
- Oil exploration (BSC)
- Lattice QCD (UREG)



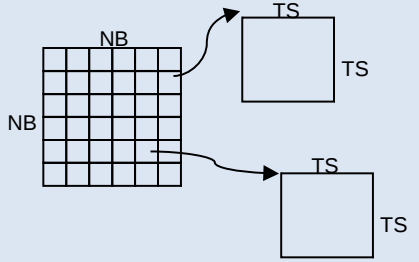
Programming Paradigm

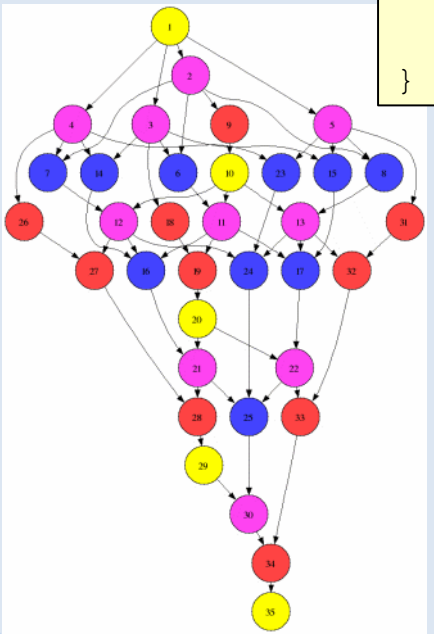
Software Architecture



OmpSs:
tasks, dependencies, heterogeneity

```
void Cholesky( float *A[NT] ) {
int i, j, k;
for (k=0; k<NT; k++) {
   spotrf (A[k][k]) ;
   for (i=k+1; i<NT; i++)
      strsm (A[k][k], A[k][i]);
   for (i=k+1; i<NT; i++) {
      for (j=k+1; j<i; j++)
         sgemm( A[k][i], A[k][j], A[j][i]);
      ssyrk (A[k][i], A[i][i]);
   }
}
```

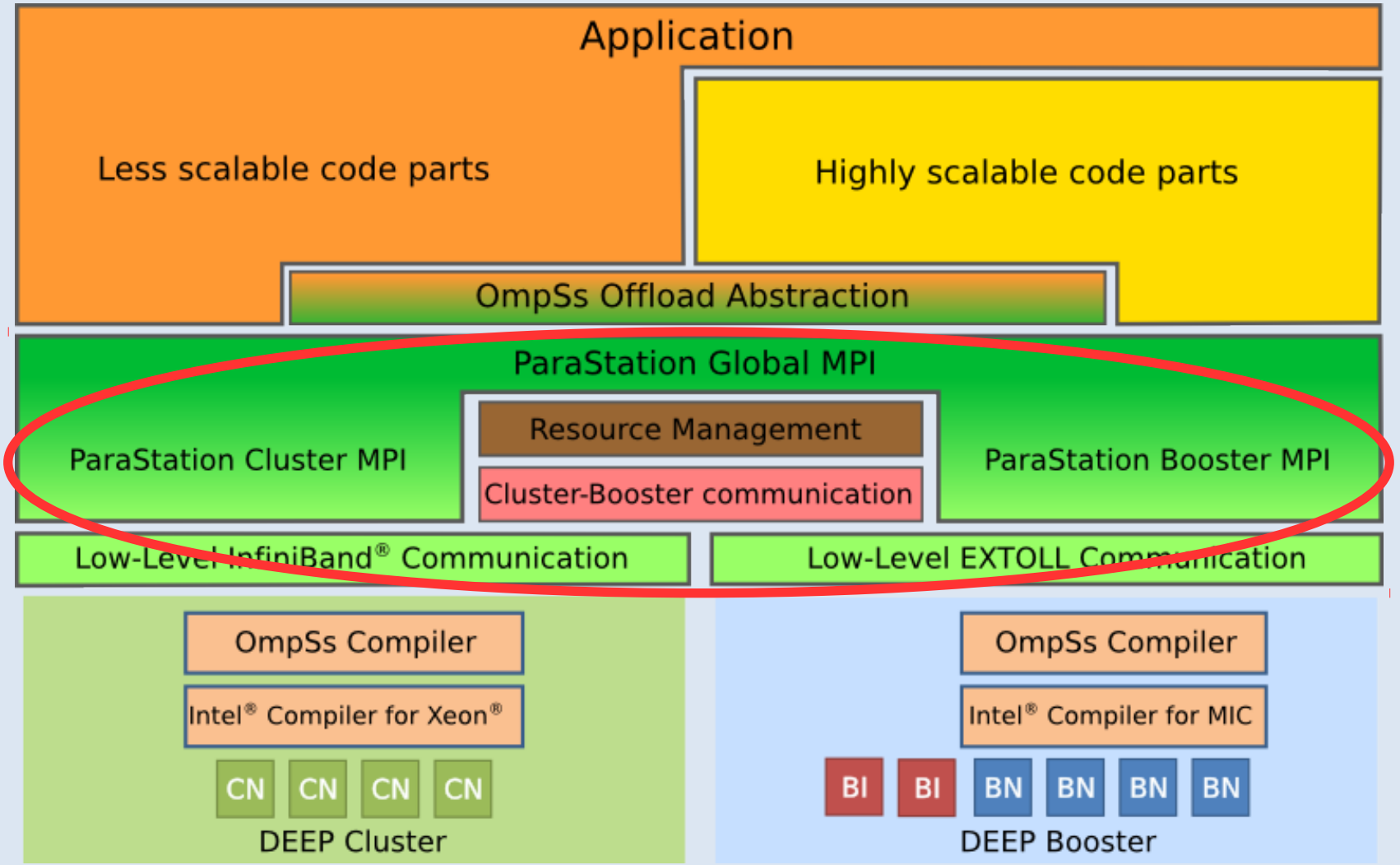




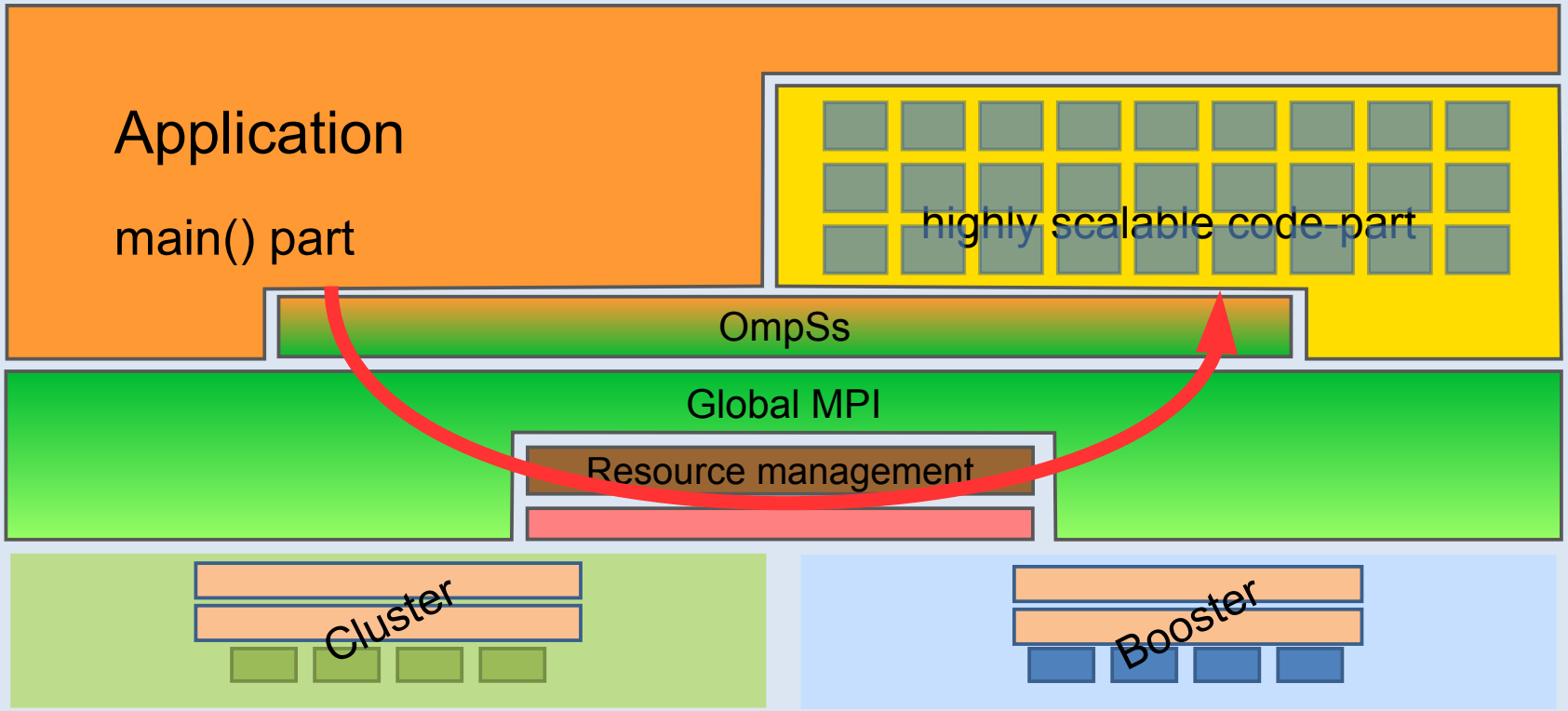
```
#pragma omp task inout ([TS][TS]A)
void spotrf (float *A);
#pragma omp task input ([TS][TS]T) inout ([TS][TS]B)
void strsm (float *T, float *B);
#pragma omp task input ([TS][TS]A,[TS][TS]B) inout ([TS][TS]C)
void sgemm (float *A, float *B, float *C);
#pragma omp task input ([TS][TS]A) inout ([TS][TS]C)
void ssyrk (float *A, float *C);
```

Decouple how we write (think sequential) from how it is executed

Software Architecture



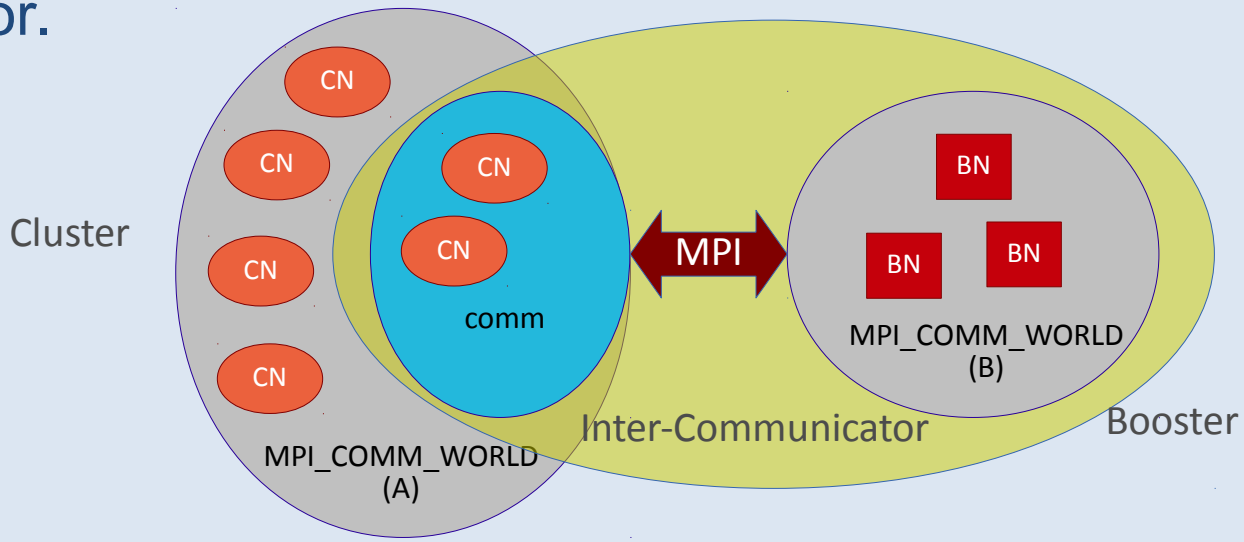
Application Startup



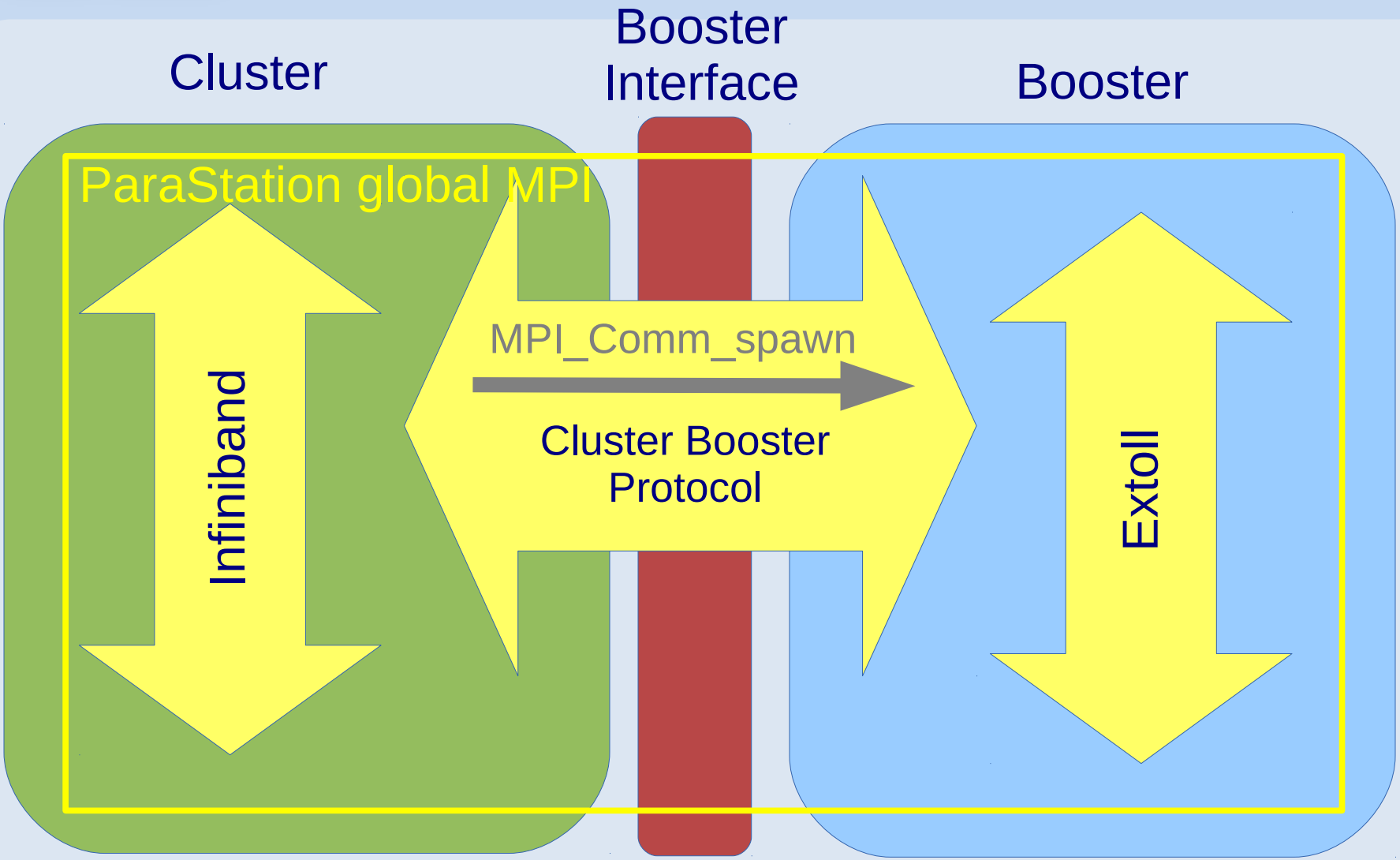
- Application's main()-part runs on Cluster-nodes (CN) only
- Actual spawn done via global MPI
- OmpSs acts as an abstraction layer
- Spawn is a collective operation of Cluster-processes
- Highly scalable code-parts (HSCP) utilize multiple Booster-nodes (BN)

MPI Process Creation

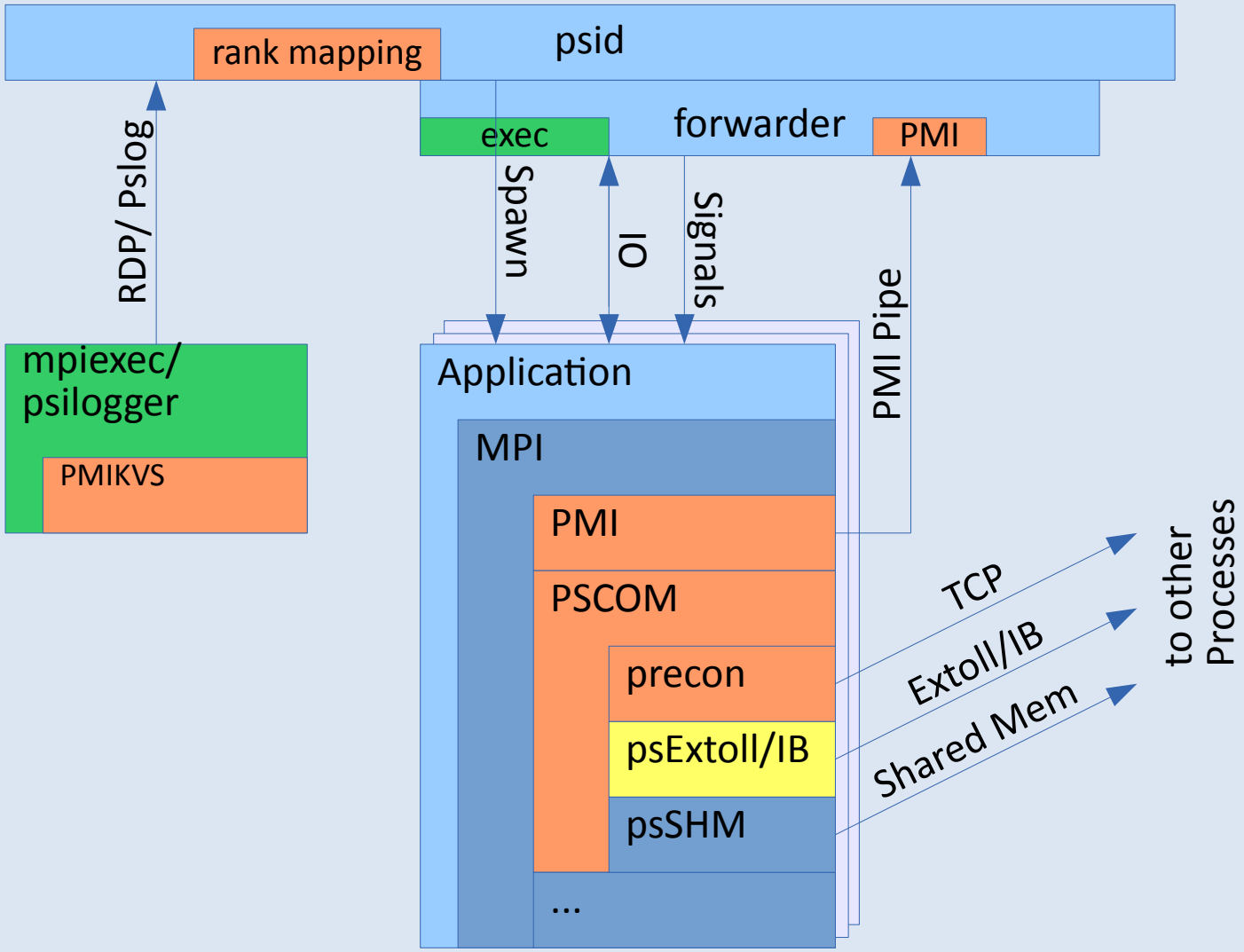
- The inter-communicator contains all parents on the one side and all children on the other side.
  - Returned by MPI_Comm_spawn for the parents
  - Returned by MPI_Get_parent by the children
- Rank numbers are the same as in the the corresponding intra-communicator.



Programming

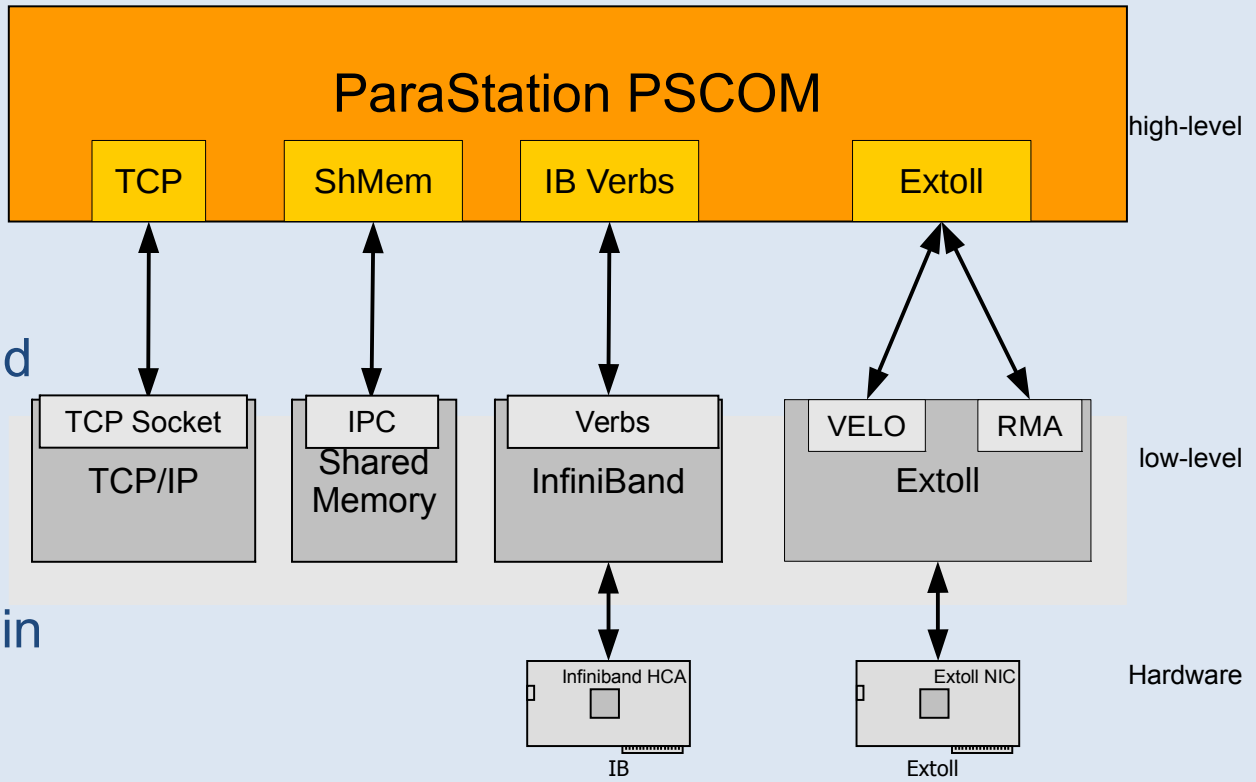


ParaStation MPI

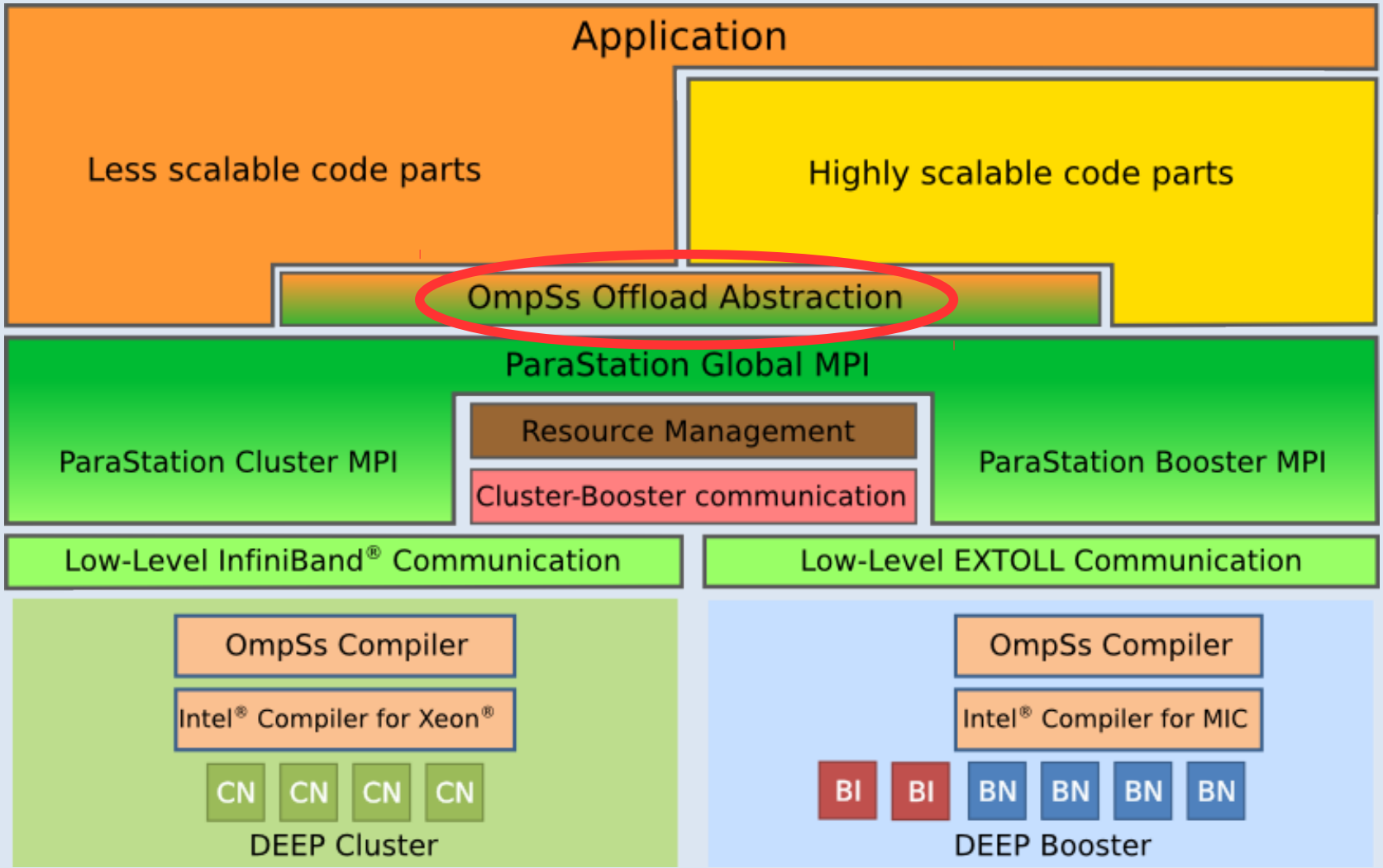


ParaStation pscom

- Unified comm layer
- pscom plugins
  - Modular
  - Flexible to extend
  - Verbs plugin
  - VELO/RMA plugin
  - Easy enabling of Cluster-Booster protocol



Software Architecture



OmpSs Offload Abstraction

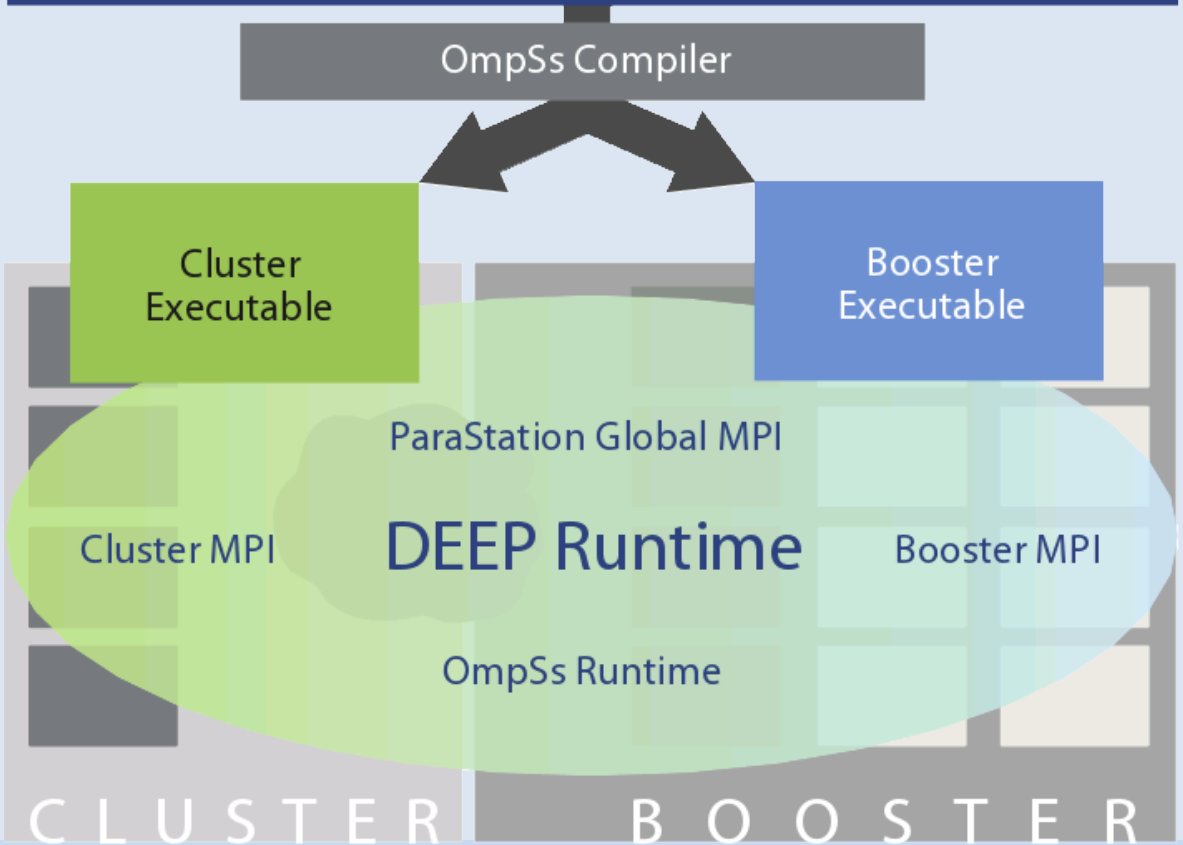
Source Code

Compiler

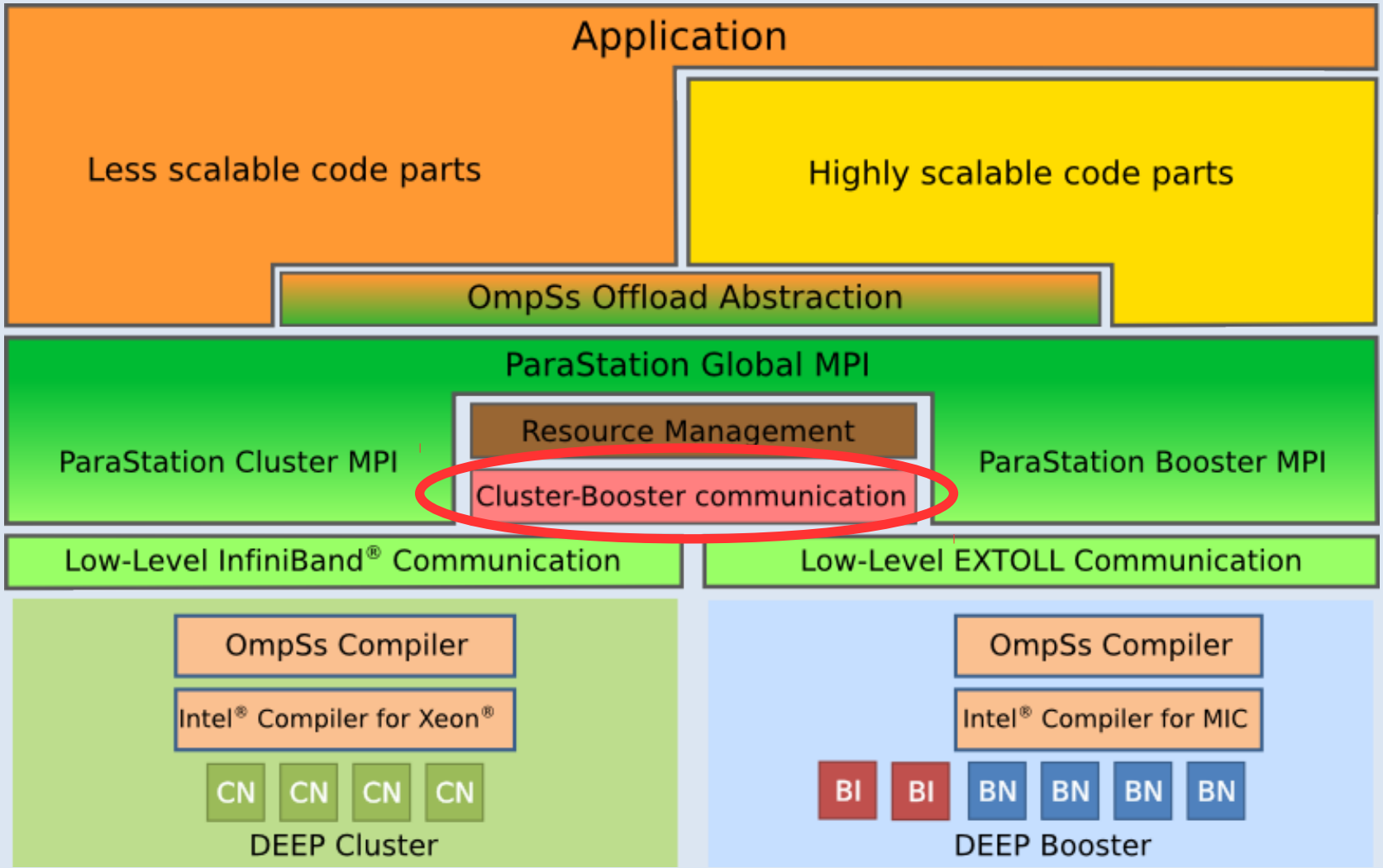
Application Binaries

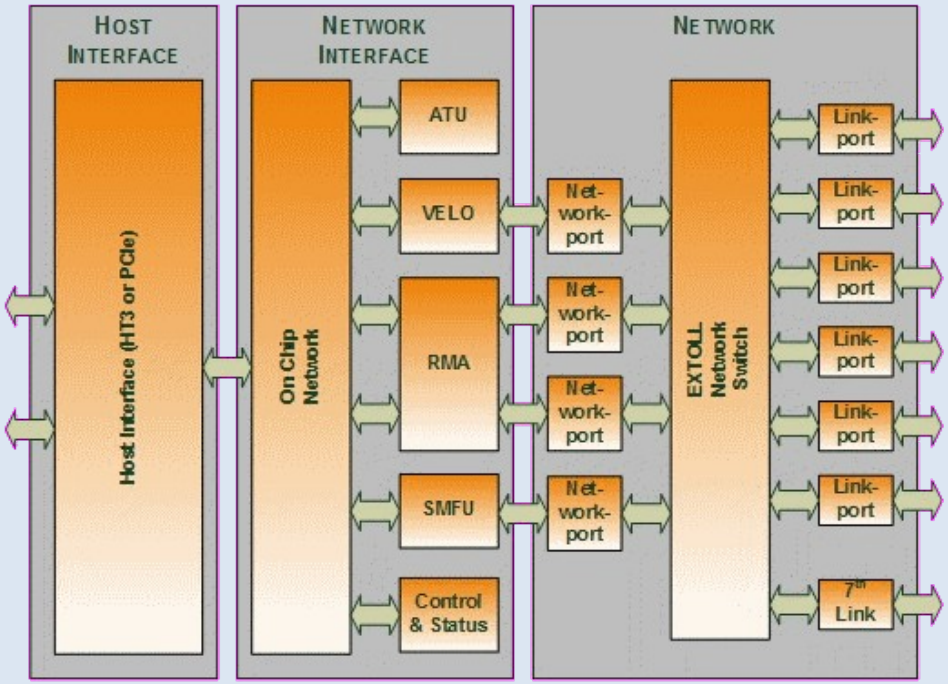
DEEP Runtime

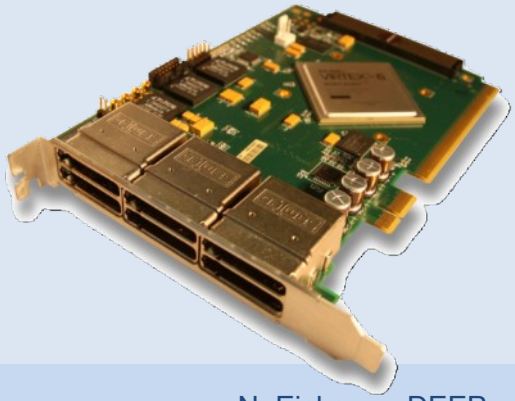
```
int main(int argc, char *argv[]){
    /*...*/
    for(int i=0; i<3; i++){
        #pragma target device (comm:size*rank+i) copy_deps
        #pragma omp task input(…) output(…)
        foo_mpi(i, …);}}
```



Software Architecture



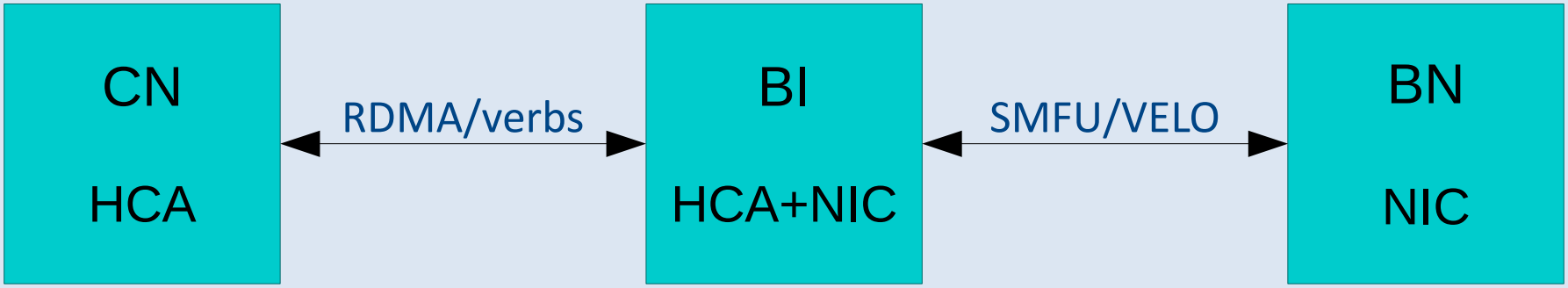




Relevant features for DEEP

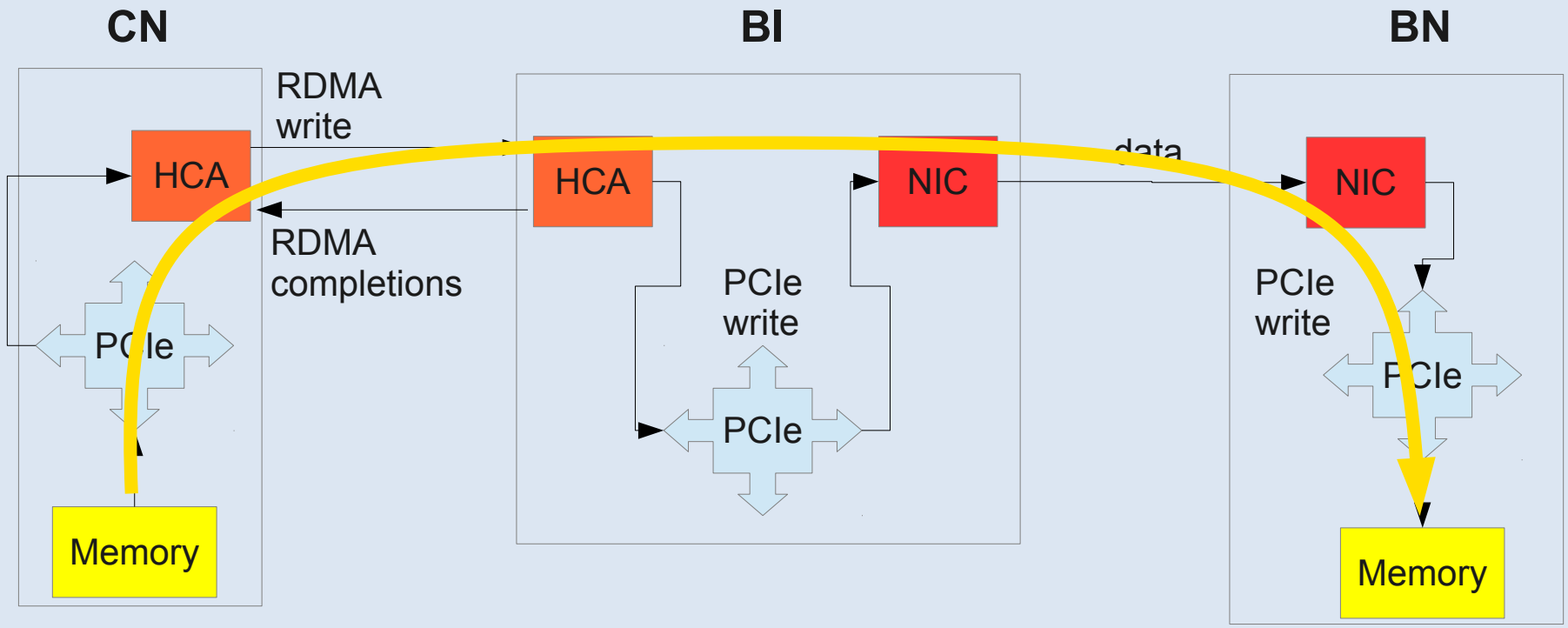
- Low latency, high bandwidth
- RMA engine for remote memory access, bulk data transfer
- VELO communication engine (zero-copy MPI)
- SMFU engine for bridging to InfiniBand
- 6 links for 3D torus topology
- 7th link for general devices
- Built-in PCIe root-port
- RAS features: CRC/ECC protection, link level retransmission
- Many status & control registers
- Access from host, via I2C bus or over EXTOLL

Cluster-Booster Communication



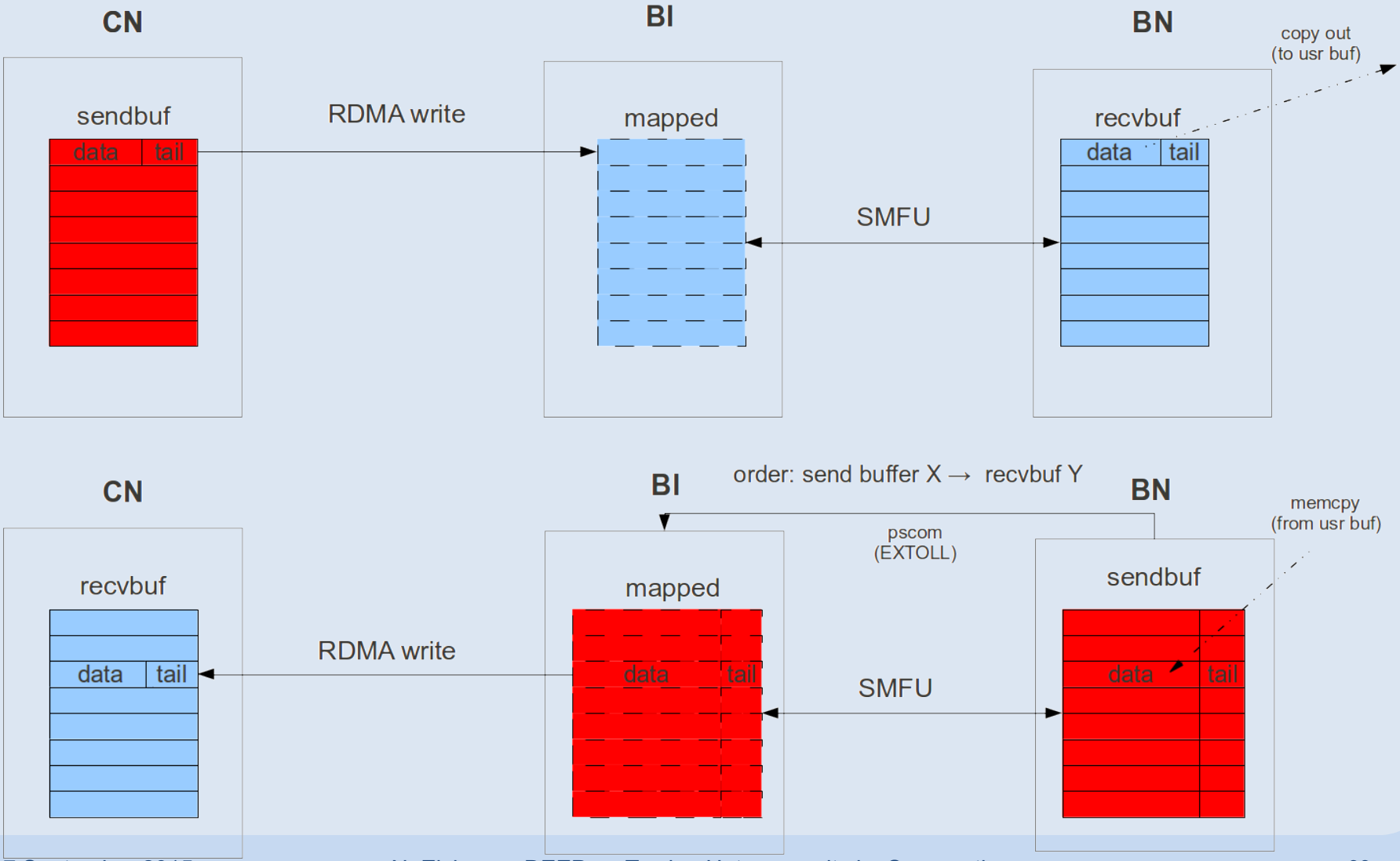
- High-bandwidth, low latency communication between Cluster Nodes and Booster Nodes needed
- Booster Interfaces bridge between InfiniBand and EXTOLL
- A Booster Interface can act in two different modes:
  - Bridge: receive packets, actively forward to end-point
  - Set-up and enable RDMA via SMFU directly between memory locations on CN and BN

CN-to-BN-Write Operation

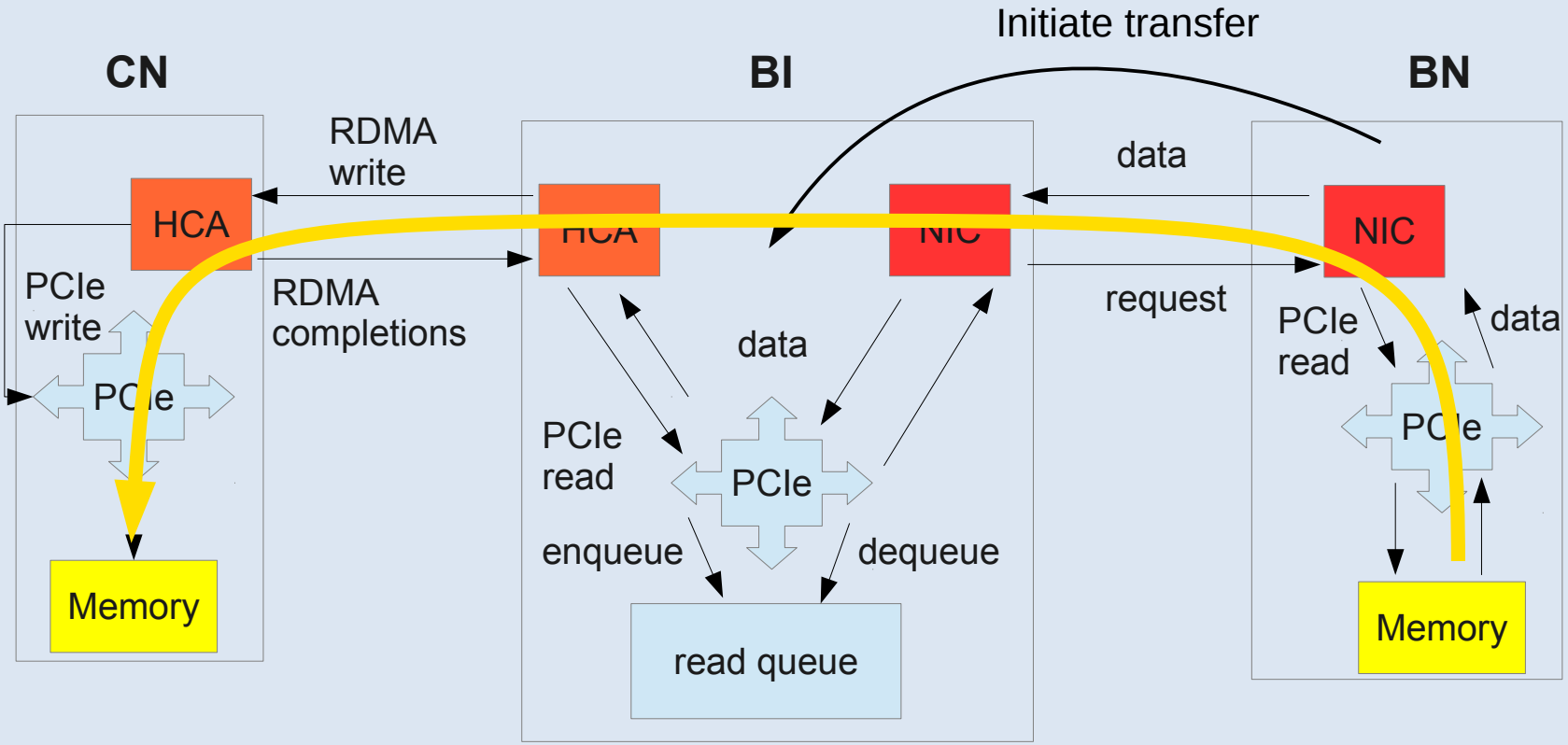


The RDMA write operation *initiated on the Cluster Node* is "forwarded" via PCIe and EXTOLL to the Booster Node

CBP address mapping

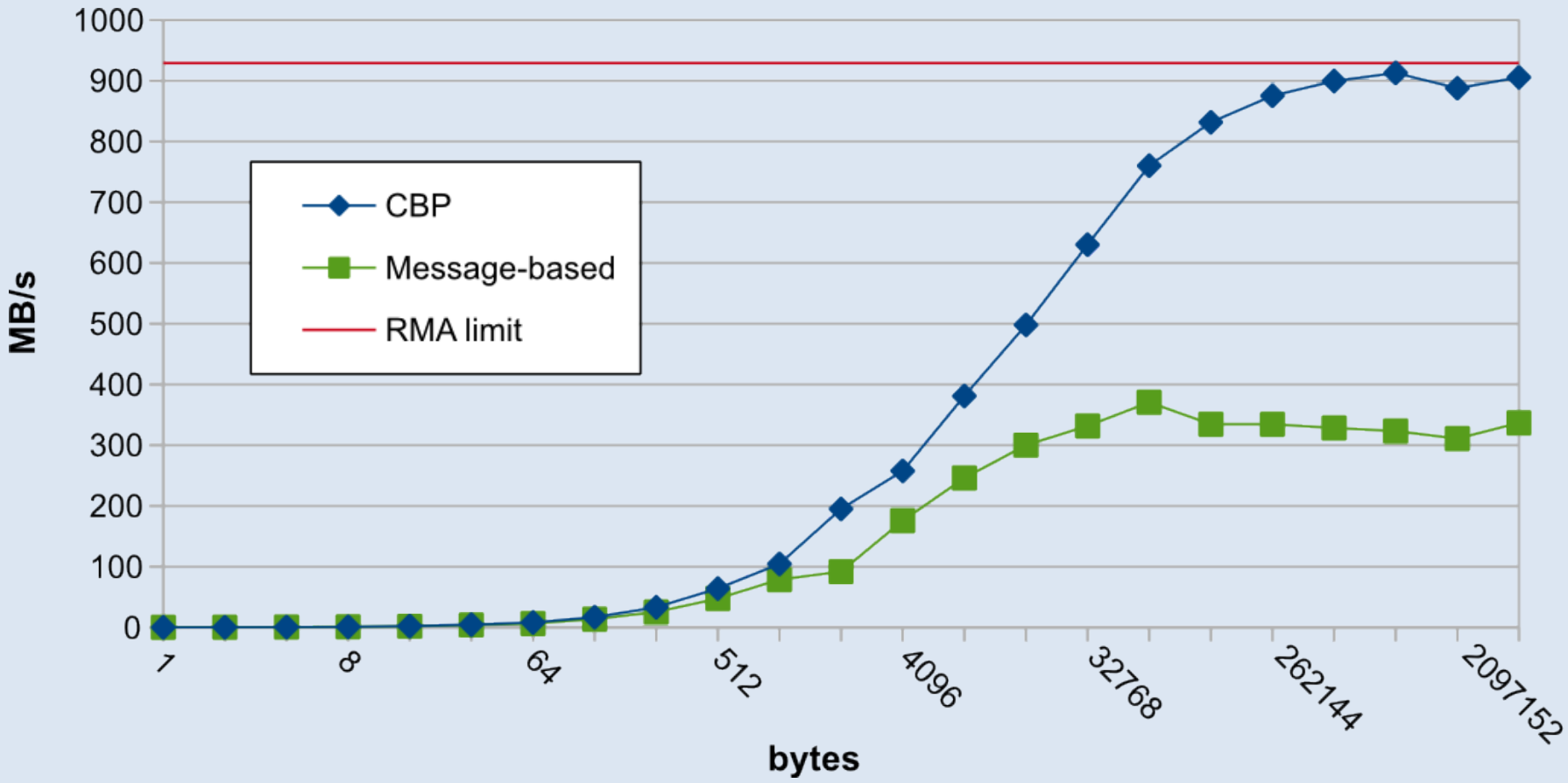


BN-to-CN Write Operation

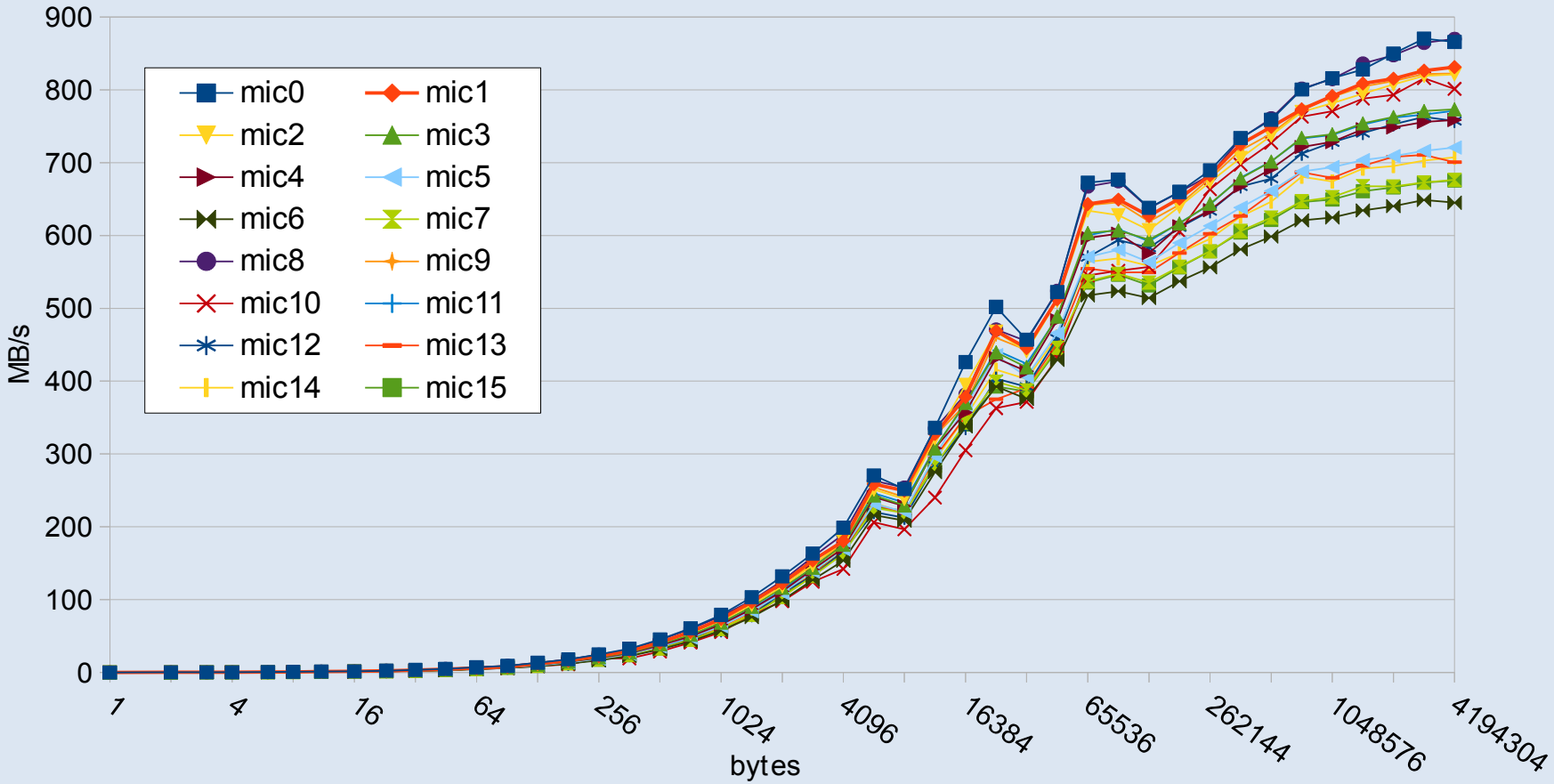


- The *initiating Booster Node* asks the Booster Interface's HCA to do an RDMA write operation to the Cluster Node's Memory
- The resulting PCIe read operation is “forwarded” by Extoll

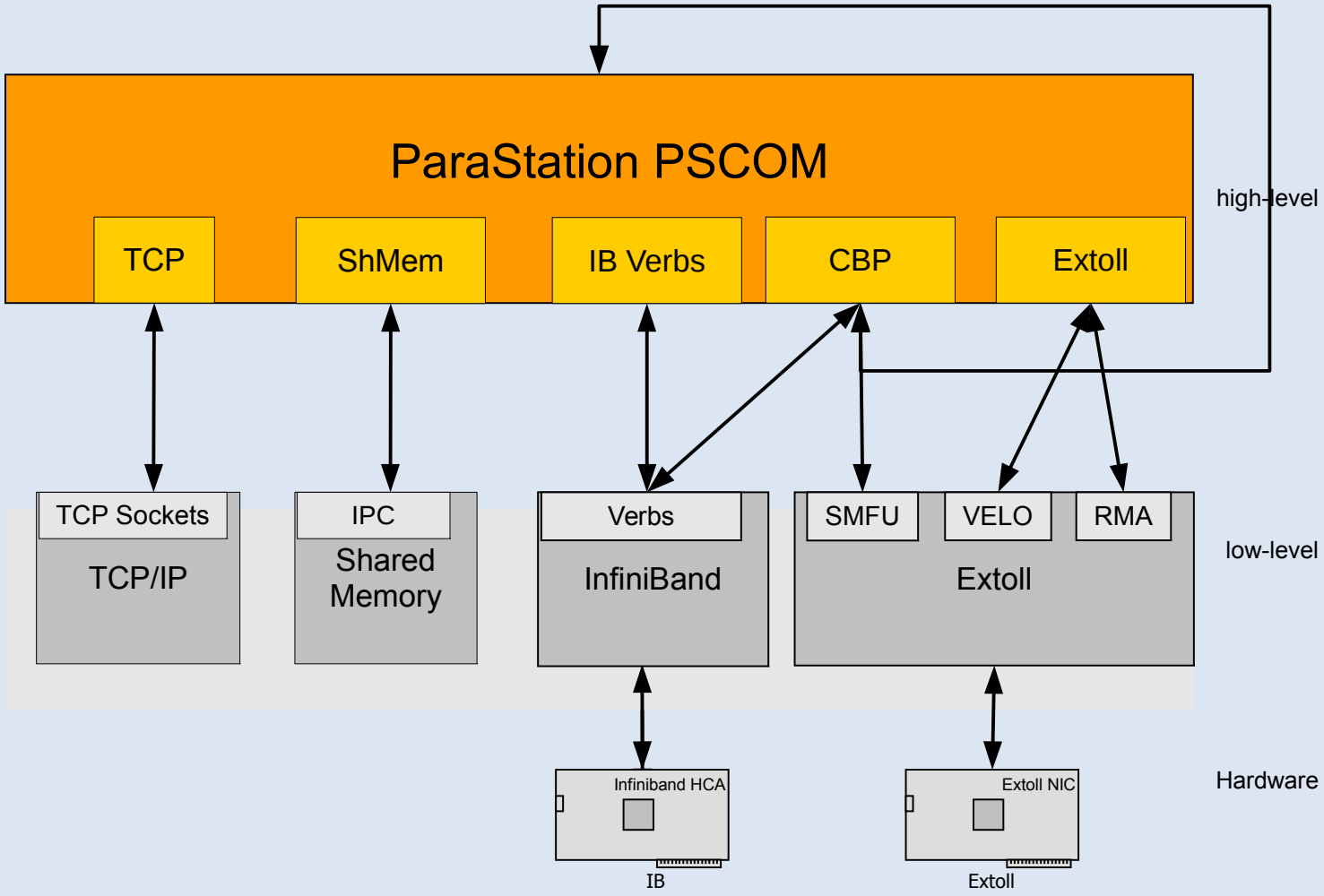
CBP Ping-Pong



CBP

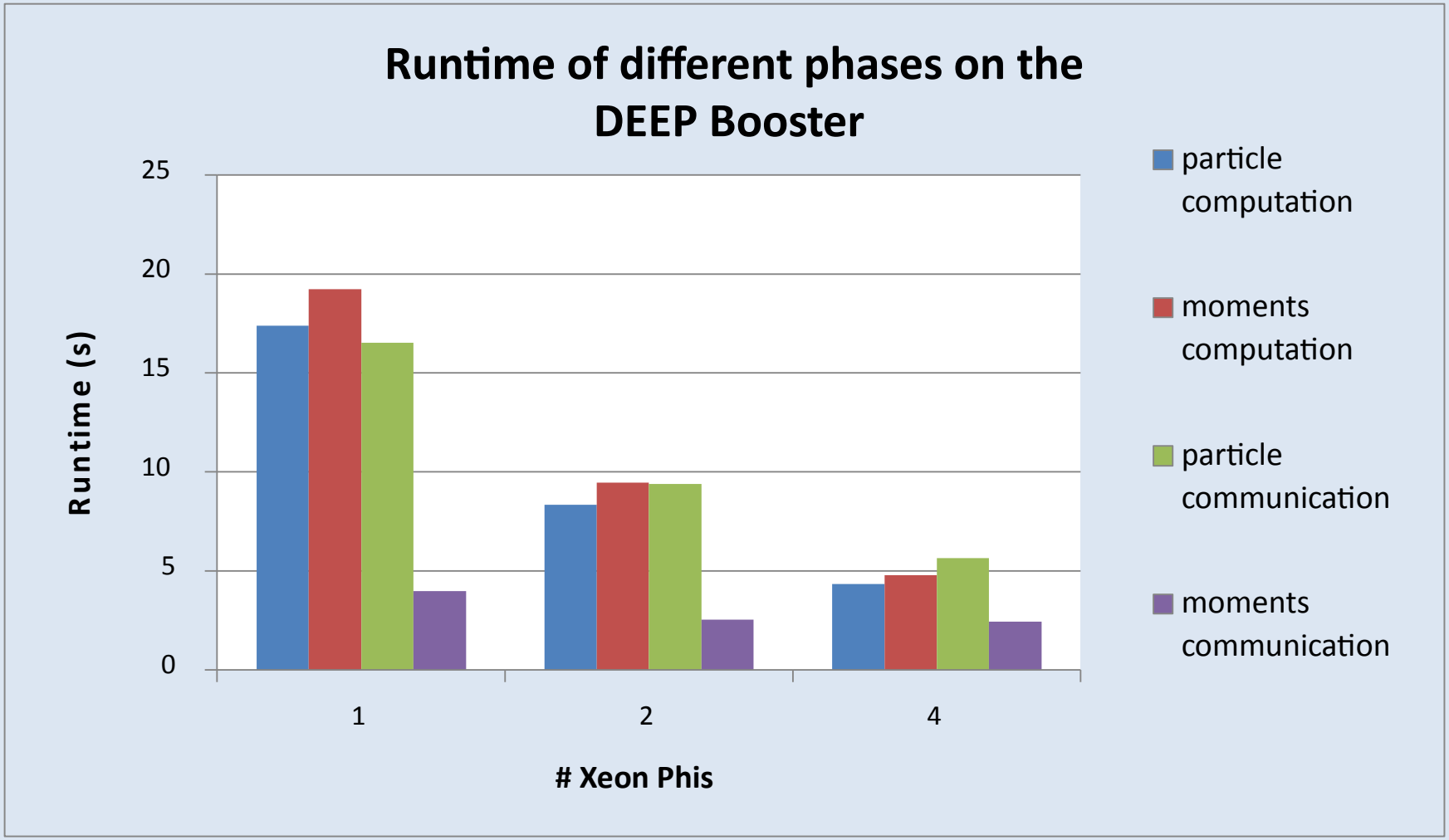

900
800
700
600
500
400
300
200
100
0
MB/s
mic0
mic1
mic2
mic3
mic4
mic5
mic6
mic7
mic8
mic9
mic10
mic11
mic12
mic13
mic14
mic15
1
4
16
64
256
1024
4096
16384
65536
262144
1048576
4194304
bytes

CBP Integration

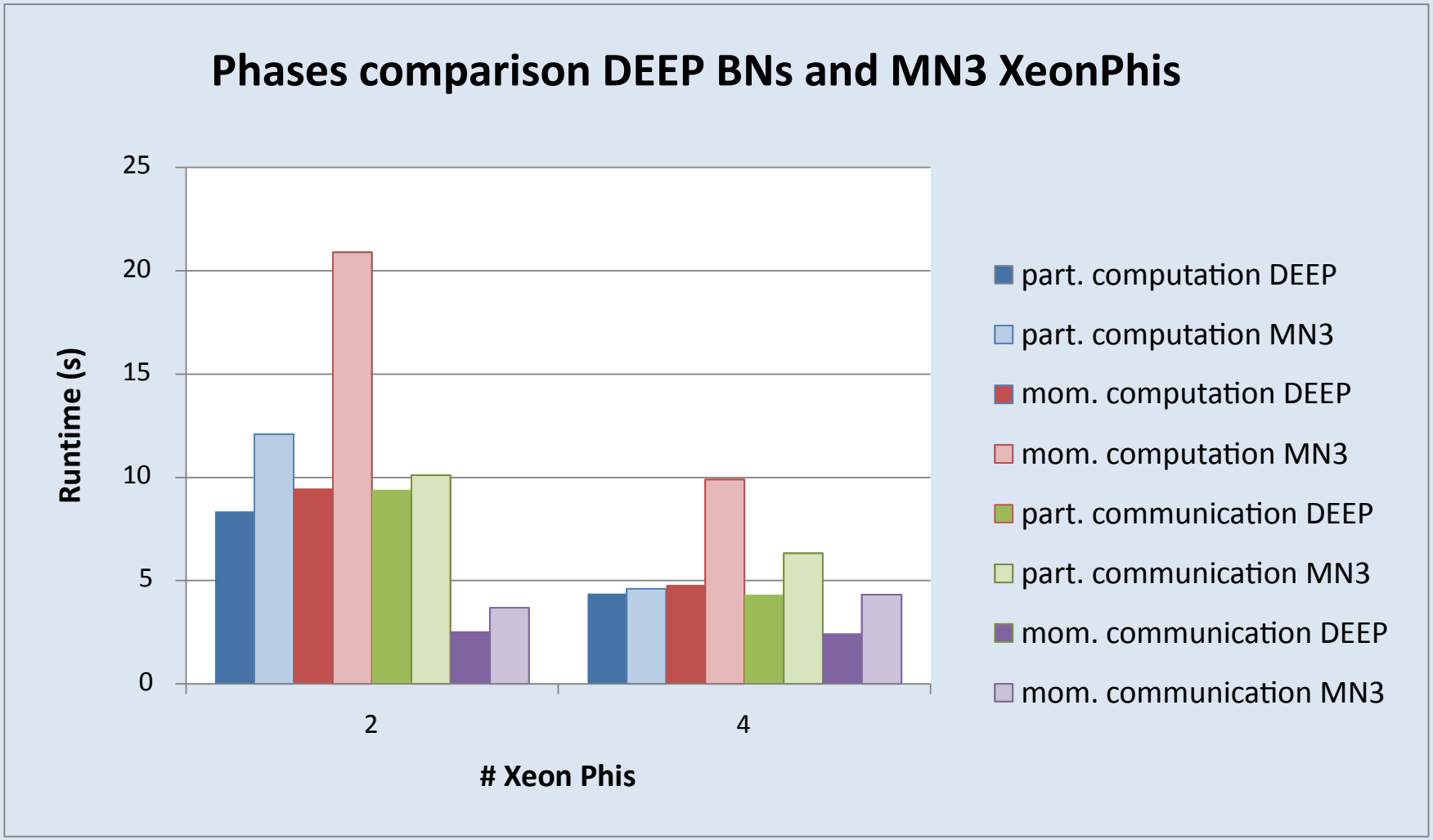


(Few) Results

iPIC3D


Runtime of different phases on the DEEP Booster
Runtime (s)
25
20
15
10
5
0
1
2
4
Xeon Phis
particle computation
moments computation
particle communication
moments communication

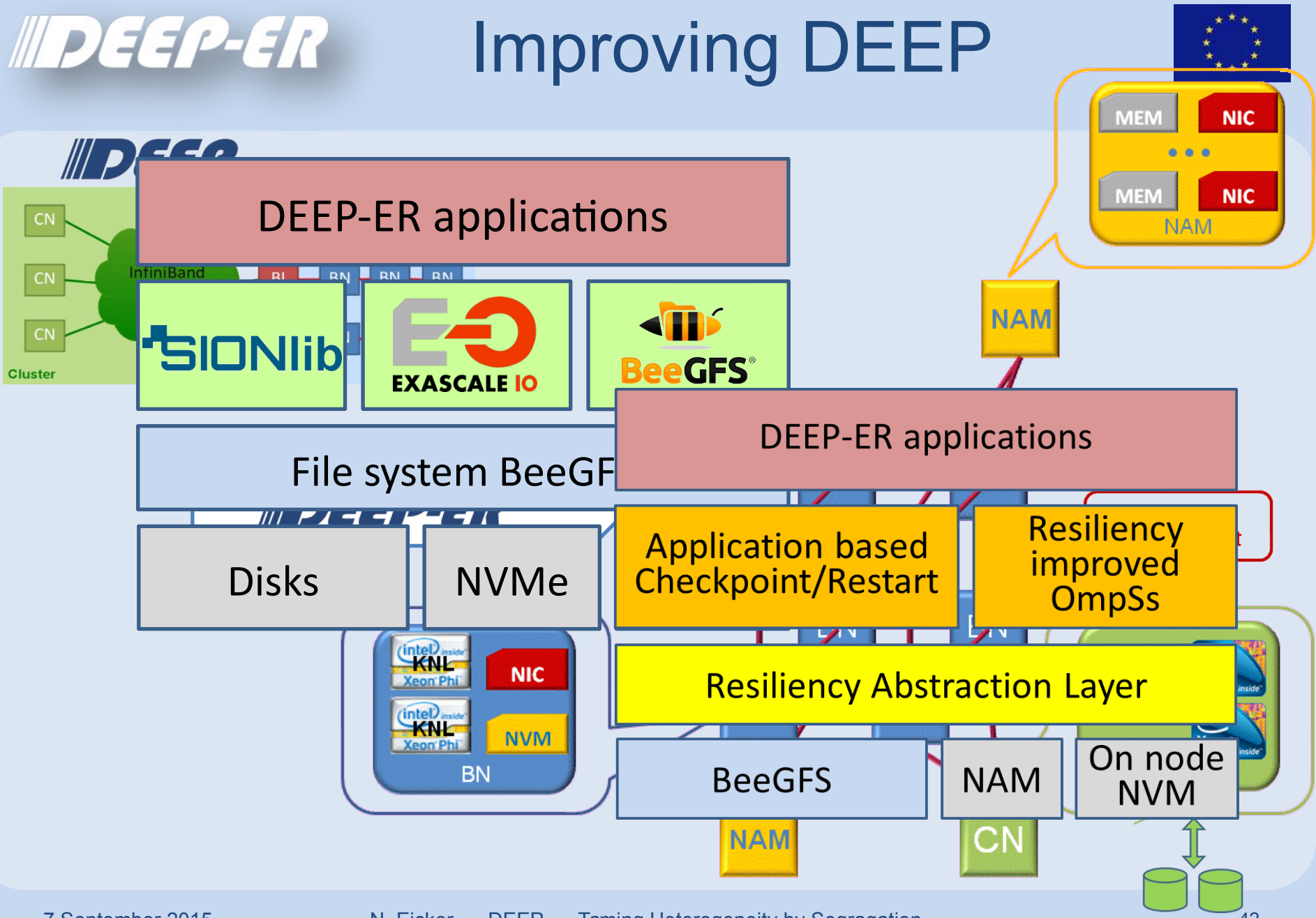
iPIC3D


Phases comparison DEEP BNs and MN3 XeonPhis
Runtime (s)
25
20
15
10
5
0
2
4
Xeon Phis
part. computation DEEP
part. computation MN3
mom. computation DEEP
mom. computation MN3
part. communication DEEP
part. communication MN3
mom. communication DEEP
mom. communication MN3

DEEP-ER Project

- 14 Partners
  - 4 PRACE hosts
  - 4 Research Centers
  - 4 Industry Partners
  - 4 Universities
  - Coordinator JSC
- 7 Countries
- Duration: 3 years
- Budget: 10.0 M€
- EU funding: 6.4 M€

Improving DEEP



Take aways

- DEEP and DEEP-ER explore new ways to use and manage heterogeneity
  - Cluster Booster Architecture and it's implementation
  - Programming Model
- DEEP's 384-KNC Booster is up and running – mostly
- As second 32-KNC Booster utilizing immersion-cooling is on the way
- Both are application driven – co-design is important
- Try to hide the details via OmpSs' abstraction layer
- DEEP-ER extends the concept to I/O, Resiliency and innovative memory technologies like NVMe and NAM
- Jülich plans to extend its new Cluster JURECA by a 10 PF/s Booster in 2016 (based on KNL)
- More info: http://www.deep-project.eu
  http://www.deep-er.eu