Performances et évolution de DACCOSIM sur architecture distribuée

Cherifa Dad, **Stéphane Vialle,** UMI GT-CNRS, CentraleSupélec, Université Paris-Saclay, Metz

Jean-Philippe Tavella, Mathieu Caujolle, EDF Lab Paris-Saclay

Workshop POMME
29-03-2016

- Plan

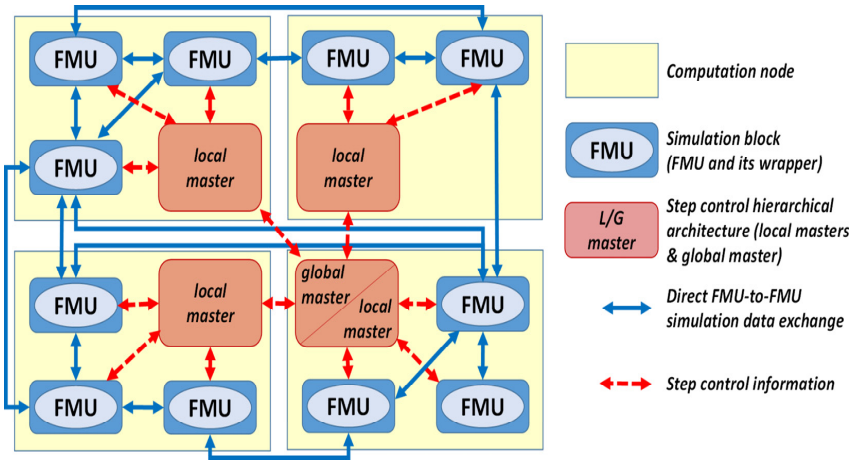
1. DACCOSIM
2. Problématique
3. Méthodologie
4. Use Case
5. Expérimentations et optimisations
7. Conclusion
8. Synthèse du projet

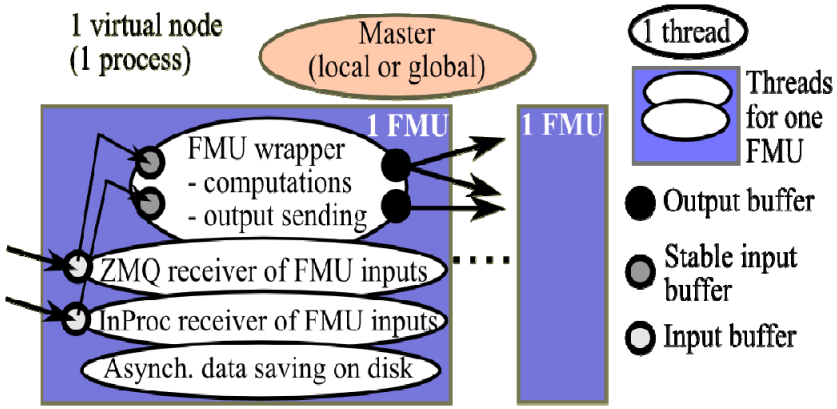
- Distributed Architecture for Controlled CO-SIMulation

- Concevoir, distribuer une multi-simulation sur des nœuds multi-cœurs

- Dédiée pour les systèmes continus discrétisés à pas de temps

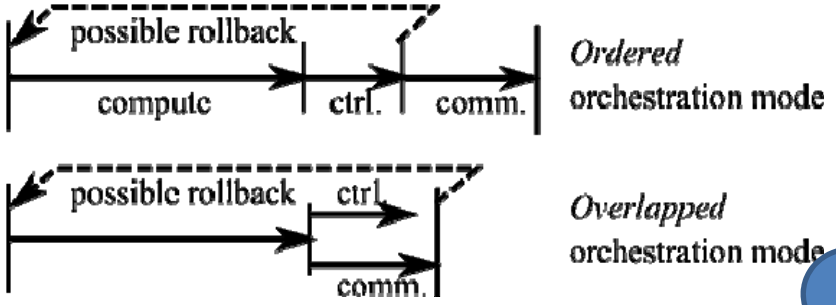
- FMU type co-simulation





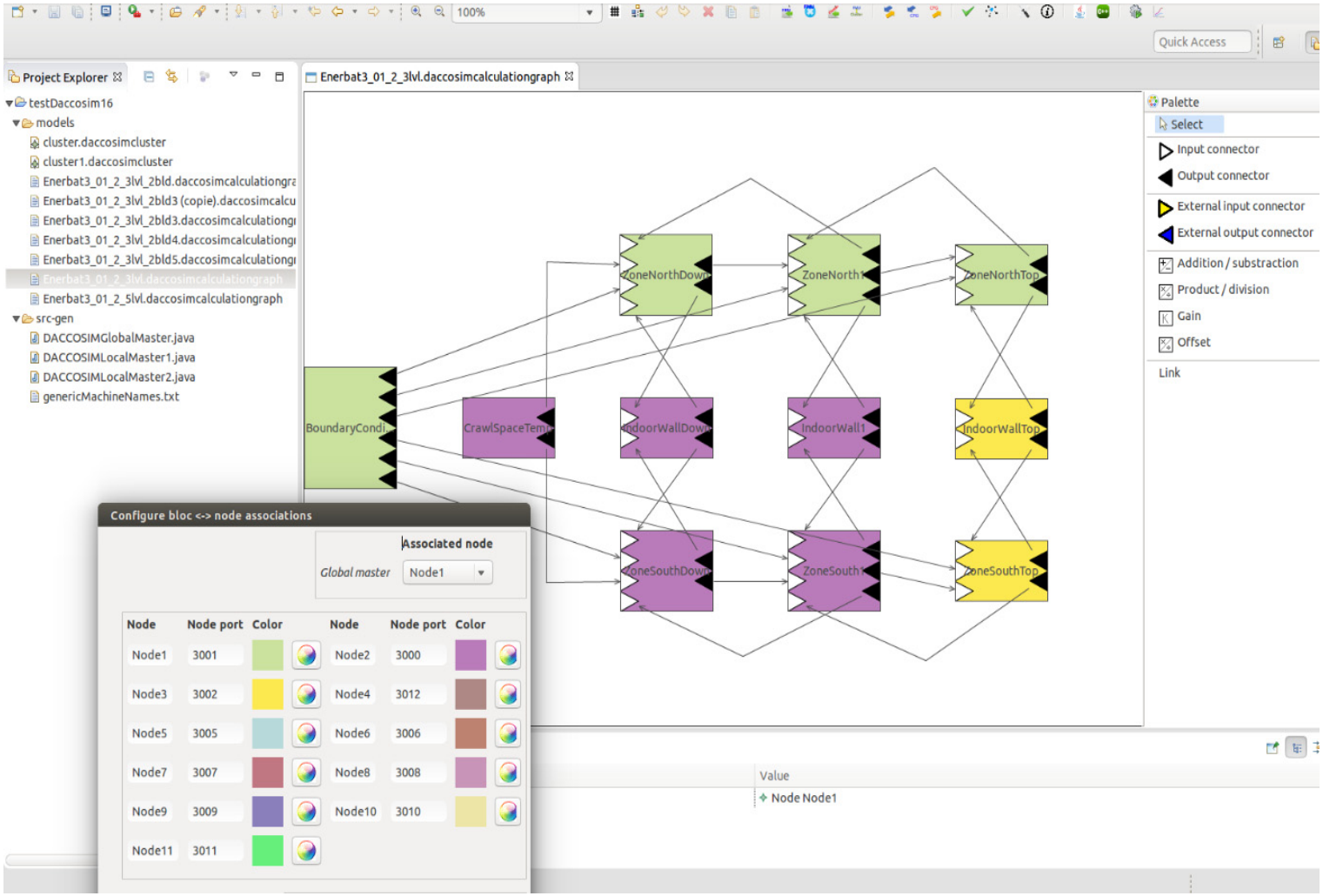
Cartographie des threads :
- thread pour les calculs
- threads de communication (inter- ou intra-nœud)
- thread de sauvegarde (IO)

- Deux mode de synchronisation :
  - *Ordered*
  - *Overlapped*



-GUI : Graphic User-friendly Interface

- Les FMU d'une même couleur → le même nœud
- Copier / coller de parties du modèle



Graphe de FMU

- Composants hétérogènes
- Composants fortement couplés
- Aucune information sur *l'intensité arithmétique* des composants
- Graphe de tâches contraint par des pas de temps

→ problème atypique à distribuer (pas un *graphe de tâches* std)

Cluster de PC multi-coeurs

- Temps de calcul sur 1 cœur ?
- Temps de calcul sur 1 nœud ?
- Temps de calcul sur 1 cluster ?

(le parallélisme ne sera pas parfait…)

→ Quelle distribution des FMU sur les nœuds et les cœurs ?

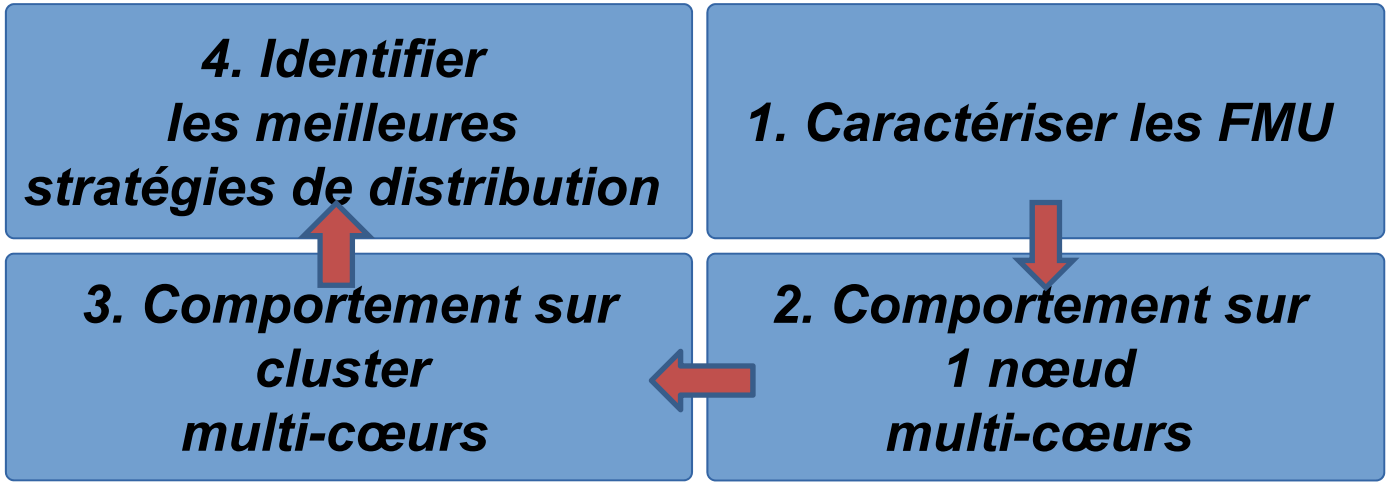
?

Définir une démarche de distribution des FMU pour minimiser le temps global de multi-simulation

- ✓ Équilibrage de charge
- ✓ Réduction du coût de communication
- ✓ Optimisation de l'usage des ressources

Définir les stratégies de distribution
+ tests et benchmarking

Mesurer les temps de calcul *(benchmarking)*

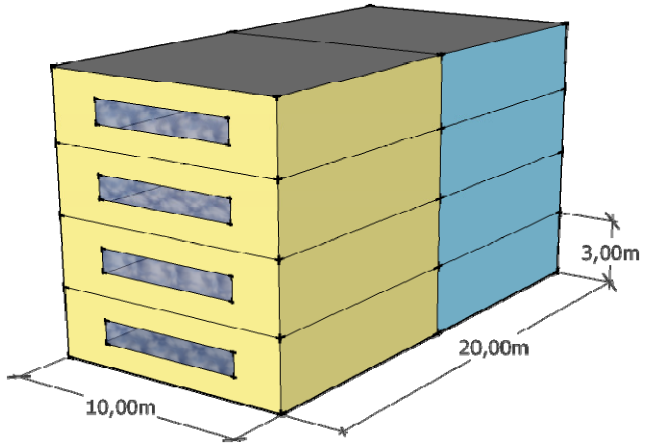


Étudier les performances sur cluster de PC multi-cœurs *(benchmarking)*

Etudier l'impact du nbr de cœurs sur le temps de calcul *(benchmarking)*

- Fourni par l'équipe ENERBAT
- Basé sur BuildSysPro (EDF Modelica library)
- Transfert de chaleur dans un immeuble à 3 étages
- Phénomènes physiques *:*

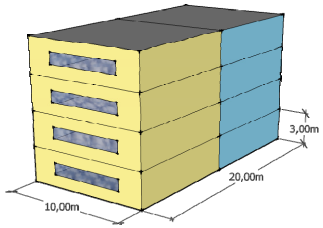
  - Conduction
  - Convection
  - Radiation longue d'onde
  - Radiation solaire



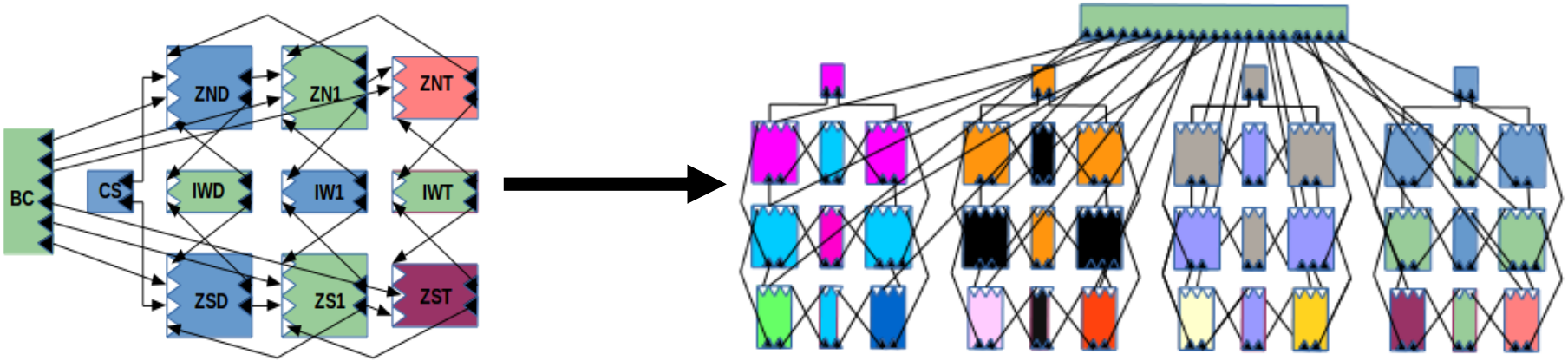
- *Boundary conditions :*

  - Données météorologiques
  - Condition spécifique pour le vide sanitaire et le dernier étage

Benchmark fourni par Enerbat : transfert de chaleur dans des immeubles (3 étages)
basé sur BuildSysPro (EDF Modelica library)



- ✓ un immeuble de 11 FMU
- ✓ niveaux de complexité 3 et 5
- ✓ réplication des immeubles jusqu'à 8



- ✓ Environnement : deux clusters d'exécution
  - 16 nœuds x 4 cœurs (Nehalem), 1Gbit/s
  - 16 nœuds x 6 cœurs (Sandy Bridge), 10Gbit/s

1. Caractérisation des FMU

Chaque FMU < tps_calcul, tps_control, tps_comm>

Exécution sous des conditions réelles

Résultats :

- Enerbat3_05 : 2 grosses FMU, 4 moyennes (50% d'une grosse) et 5 négligeables
- Enerbat3_03 : 2 grosses FMU, 4 moyennes (75% d'une grosse), 3 petites (20% d'une grosse) et 2 négligeables

2. Comportement sur un nœud multi-cœurs

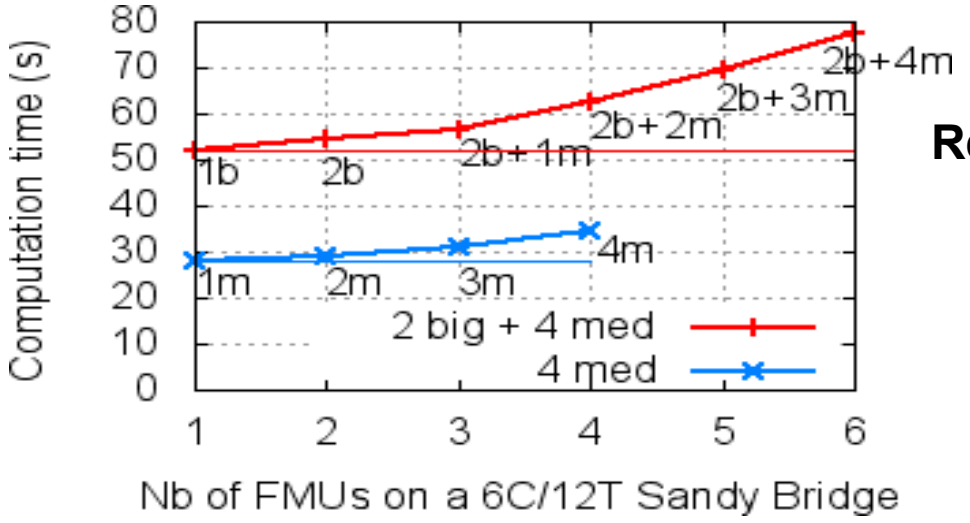


Fig 1: computation time on a Sandy Bridge node

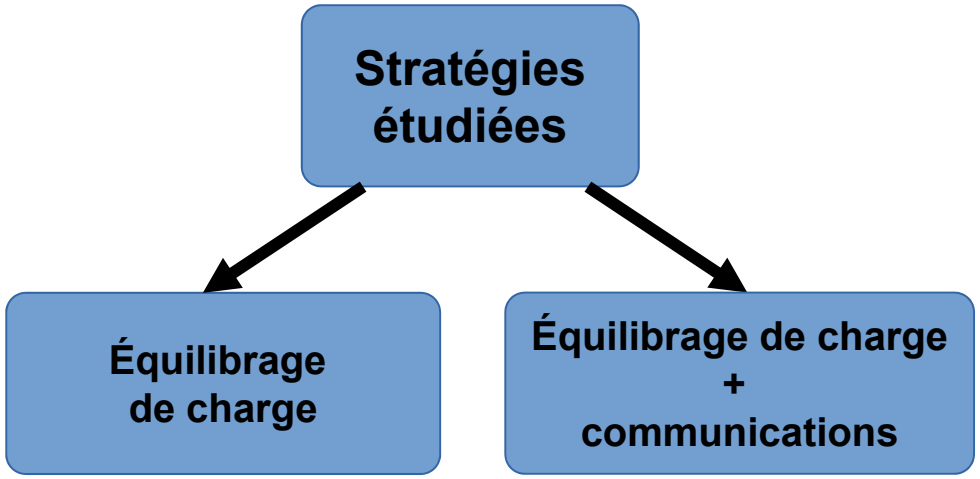
Résultats sur 1 nœud à 6 cœurs :

tps(6 FMU, 1 nœud)

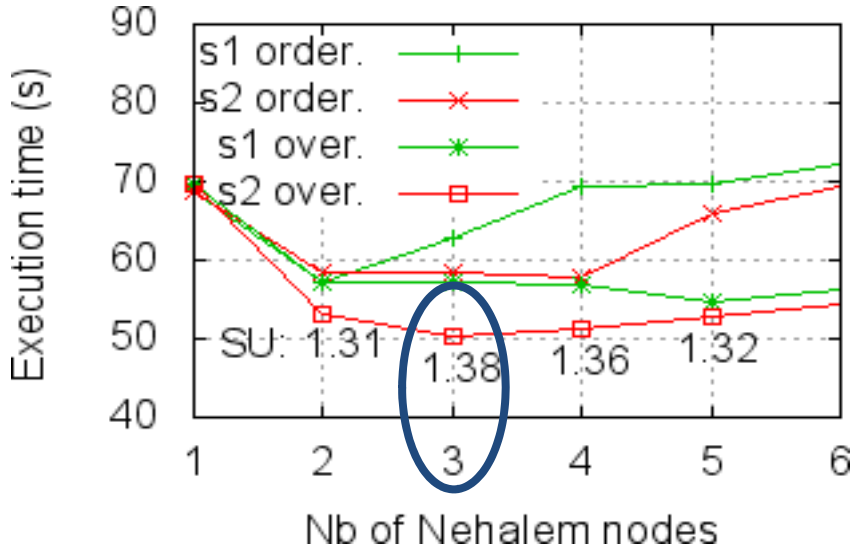
>

tps (1 FMU, 1 nœud)

3 & 4. Comportement et stratégies sur cluster de multi-cœurs :



- Minimiser les communications inter-noeuds
- Équilibrer la charge des calcul entre les nœuds



- S1 : équilibrage de charge
- S2 : maximiser les comms. intra-noeud + équilibrage de charge

Résultats :

✓ Stratégie S2 meilleure que S1

✓ **Un *speedup* de 1,38** (sur 3 nœuds)

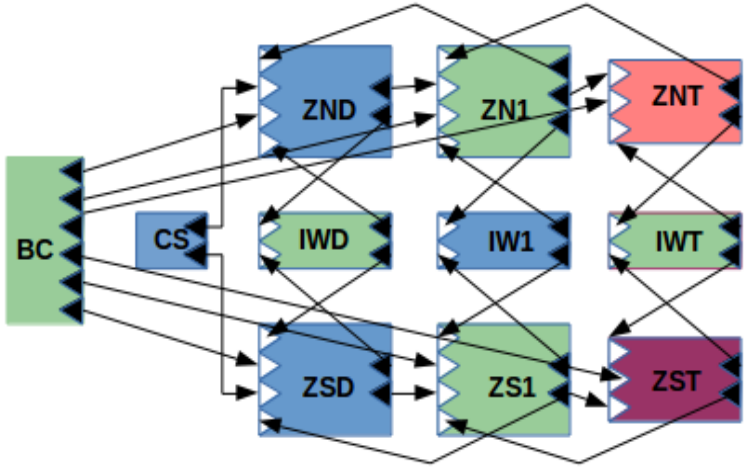
✓ Résultats identiques à [0.3% - 1%] près …. **en cours d'analyse.**

Fig 2: co-simulation time on a Nehalem PC Cluster

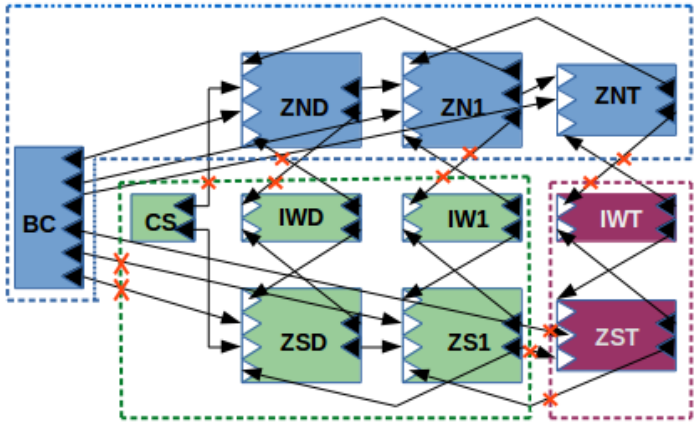
Réplication : 1 building sur n1 nœuds → b buildings sur b × n1 nœuds

Notation : T(1 building, n1 nœuds) : répartition optimale sur n1 nœuds
T(b buildings, b × n1 nœuds)

On espère : $T(b, b \times n1) = T(1, 1 \times n\ 1) = C^{st}$ sur nos clusters

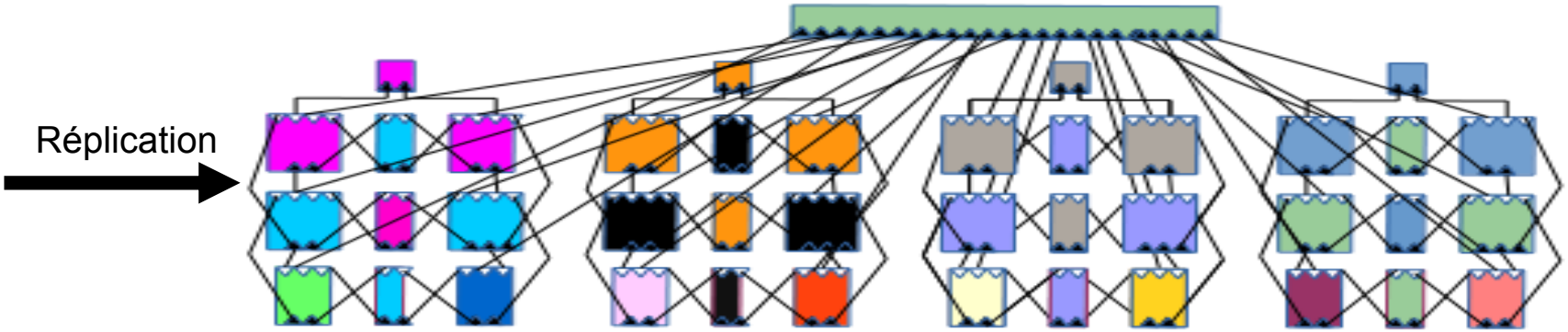


Enerbat3_0**5** sur 4 nœuds (s1 : charge)



Enerbat3_0**3** sur 3 nœuds (s2 : charge + comms)

Réplication



Expérience de « size up » :

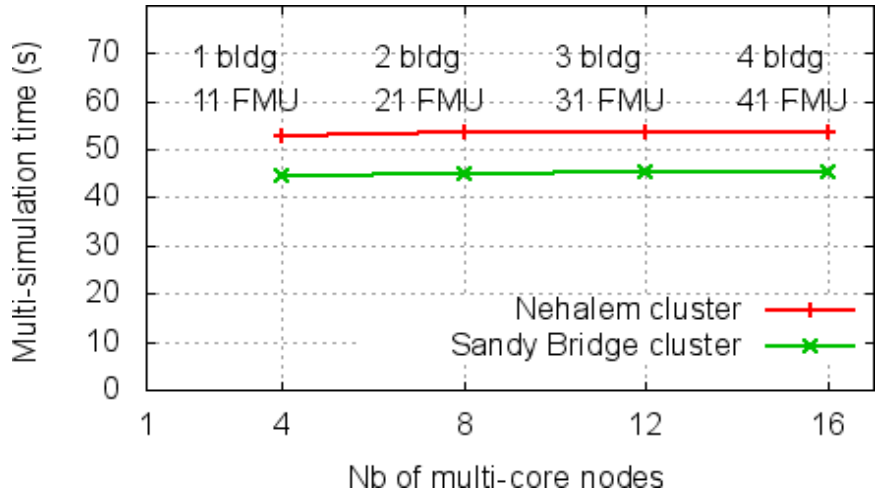


Fig 3: Size up experiments on cl5 benchmark

Use case avec une complexité 5 :

- ✓ Les communications sont négligeables vis-à-vis la charge de calcul
- ✓ Stratégie : équilibrage de charge
- ✓ Réplication de la bonne répartition (n × 4 noeuds)

→ **Size up (presque) idéal**

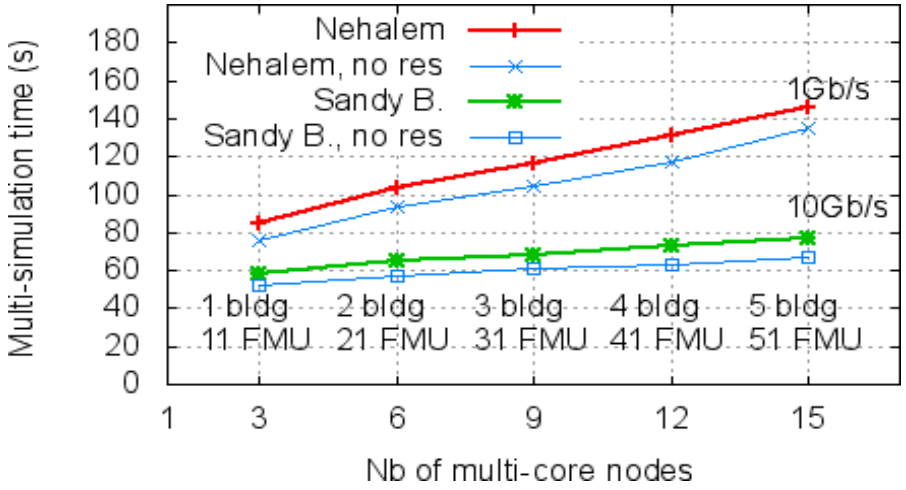


Fig 4: Size up experiments on cl3 benchmark

Use case avec une complexité 3 :

- ✓ Communications non négligeables
- ✓ Stratégie : éq. de charge + comms
- ✓ Réplication de la bonne répartition (n × 3 noeuds)

→ **Les performances en 10Gb/s sont meilleures qu'en 1 Gb/s**

→ **Size up acceptable en 10Gb/s**

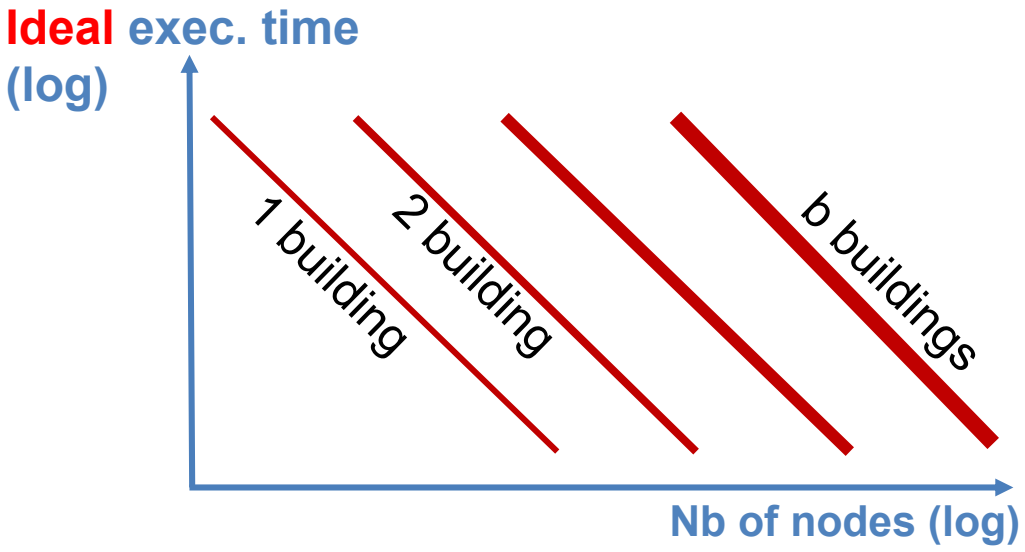
Expérience de « passage à l'échelle » : profils d'accélération de pbs croissants

- ✓ Use case de complexité 3 (calcul & communication)
- ✓ Distribution élémentaire : 1 building sur 3 nœuds (stratégie « charge & comms »)

→ Quelles accélérations pour des pbs plus gros (plus de buildings) ?
→ Comportements similaires quand le problème grossit ?

T_{ideal} (b buildings, n nœuds) = T(b buildings, 1 nœud) / n

Mais exécution sur 1 seul nœud pas toujours possible



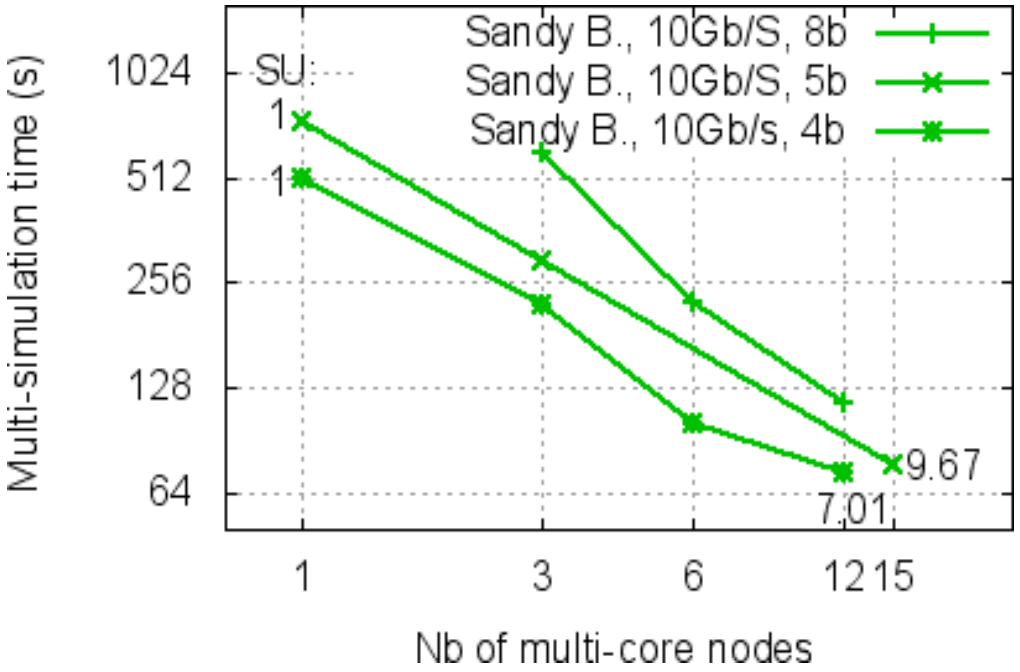
Expérience de « passage à l'échelle » : profils d'accélération de pbs croissants

- ✓ Use case de complexité 3 (calcul & communication)
- ✓ Distribution élémentaire : 1 building sur 3 nœuds (stratégie « charge & comms »)

→ Quelles accélérations pour des pbs plus gros ?
→ Comportements similaires quand le problème grossit ?

T_{ideal} (b buildings, n nœuds) = T(b buildings, 1 nœud) / n

Mais exécution sur 1 seul nœud pas toujours possible



Résultats :

- Tous les pbs accélèrent
  Courbes de temps « parallèles »
  Comportements similaires
- Un **bon speedup de 9,67** pour 5 buildings sur 15 nœuds hexa-cœurs (*vs* 1 nœud hexa-cœurs)
- Pas assez de nœuds pour pousser le pb avec 8 buildings…

Fig 5: CI3 benchmark time with 4, 5 and 8 buildings on 10Gb/s

Améliorations des communications de DACCOSIM :

- Implantation actuelle avec la lib. « ZeroMQ »
- Beaucoup de petits messages : sensibles à la latence et à la Bw
- Perfs DACCOSIM sur Eth 10Gb/s **>** perfs sur Eth 1Gb/s

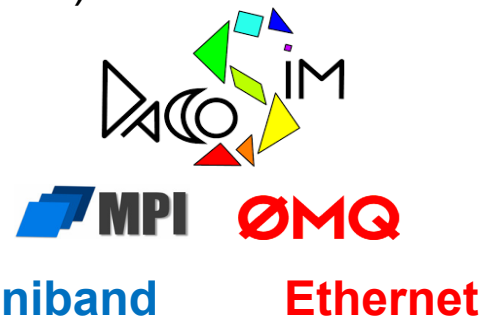
→ Exploiter des clusters HPC « **Infiniband** » (standard HPC)

→ Utiliser la lib. « **MPI »** implantée directement sur Infiniband [1]

CEI 2015-2016 EDF-CentraleSupélec : « portage DACCOSIM sur MPI »

- 50% de DACCOSIM porté sur MPI
- Mais … plus lent que ZeroMQ sur Eth 10Gb/s
  - ✓ petits messages pas adaptés à MPI
  - ✓ archi. logicielle de DACCOSIM conçue pour ZeroMQ
  - ✓ ZeroMQ conçu pour des réseaux std (efficace sur Ethernet)

→ **DACCOSIM sur ZeroMQ et MPI, et sélection au runtime**



[1] « Message passing on InfiniBand RDMA for parallel run-time supports ». University of Torino & University of Pisa.

- Benchmarks d'un *use case ENERBAT* avec deux complexités différentes sur deux clusters différents
- Comportement complexe des cœurs d'un nœud en fonction du nombre de FMU supportées
- Conception et expérimentation d'une méthodologie et de deux premières stratégies de distribution sur cluster

- Diminution significative du temps de multi-simulation
- Exécution jusqu'à 81 FMU sur 12 nœuds

- Première solution au dessus de MPI

Prochaines étapes :

- Automatisation de la distribution des FMU sur les nœuds de calcul
- Optimisation de la taille des messages échangés entre FMU
- Implantation sur MPI (pour version ZeroMQ et MPI)
- Exec. sur clusters Infiniband d'EDF et tests à grande échelle (ENERBAT)

Merci pour votre attention

Cherifa Dad, Stephane Vialle, Mathieu Caujolle, Jean-Philippe Tavella, Michel Ianotto. *"Parallelization, Distribution and Scaling of Multi-Simulations on Multi-Core Clusters, with DACCOSIM Environment".* Technical & Research report, EDF & CentraleSupelec. March 2016.
https://hal-centralesupelec.archives-ouvertes.fr/hal-01289194v3/document