

Xeon Phi au CRIHAN

Patrick BOUSQUET-MÉLOU (pbm@crihan.fr)

16 Juin 2015

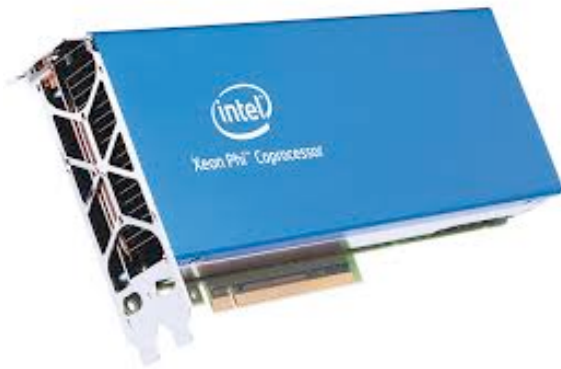
Objectifs

Veille technologique architecture many-core

- Se former pour préparer les utilisateurs au Xeon Phi «Knights Landing» (2016)
 - Administration
 - Programmation sur cette architecture many-core
- Ouvrir ce type d'architecture à des chercheurs désireux de la tester
- Acquérir en 2016 plusieurs nœuds de calcul dotés du (co -) processeur Phi «Knights Landing» (> 60 cœurs, 4 threads/cœur)
 - Préparer les chercheurs à l'apparition de clusters de ce type dans les centres nationaux ou internationaux

Xeon Phi au CRIHAN

Configuration disponible

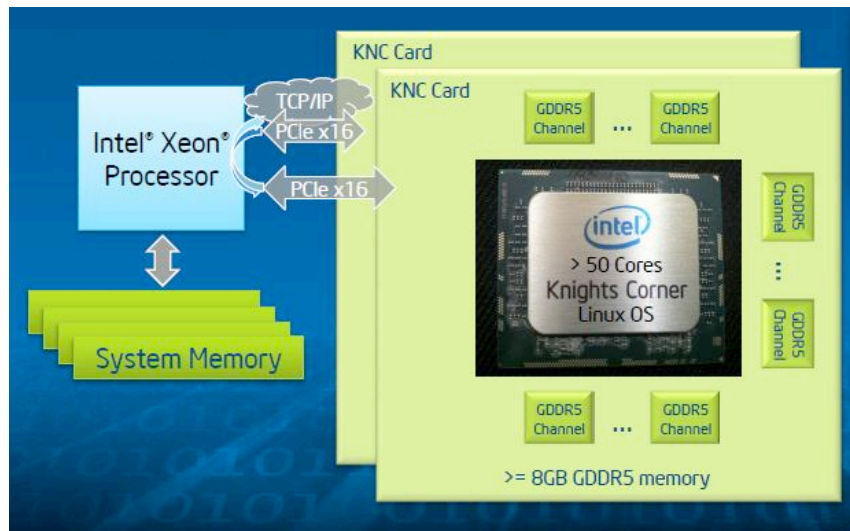


- 1 serveur IBM dx360-M4
 - 2 x (CPU Ivy Bridge) @ 2,5 GHz
 - 20 cœurs
 - 2 cartes Xeon Phi «Knights Corner» (KNC)
 - 128 Go RAM

Xeon Phi au CRIHAN (2)

Knights Corner (KNC)

- Co-processeur
 - 1 TFlop/s double précision
 - 60 cœurs (x 4 threads) @ 1,052 GHz
 - 8 GB de mémoire (7 GB utiles)
 - Connexion CPU / KNC
 - PCIe Gen2



Xeon Phi au CRIHAN (3)

Environnement logiciel (sur Antares)

- Compilateur Intel 13.1.1 et 14.0.2, librairie Intel **MPI** 4.1.0 et 5.0.2
- **BLAS / LAPACK / FFTW, ScaLAPACK** avec MKL
 - modules « mkl/11.0.3/mic-native/* »
- **Scotch** (partition de maillage)
 - module « scotch/6.0.0_mic-native »
- **HDF5**
 - module « hdf5/1.8.11/mic-native »
- Usage général de modules :
 - <http://www-tech.crihan.fr/calcul/tech/documentation-ibm-cluster-idataplex-antares/modules/>

Modèles de programmation

Trois approches d'utilisation du Xeon Phi

- Une application parallèle (MPI et/ou OpenMP) écrite dans un langage standard (FORTRAN, C, C++) peut utiliser un Phi de trois manières

1. Mode **natif**

- Compilation avec l'option «-mmic» du compilateur Intel

2. Mode «**offload**»

- Exécution de sections de codes sur un co-processeur (les autres sections sur les CPU du serveur hôte)
- <https://software.intel.com/sites/default/files/ee/90/offload-compiler-runtime-for-the-intel-xeon-phi-coprocessor-130315.pdf>
- <http://www.prace-ri.eu/best-practice-guide-intel-xeon-phi-html/>

Modèles de programmation (2)

Trois approches d'utilisation du Xeon Phi (2)

- BLAS/LAPACK et FFT : mode «offload» avec la librairie MKL :
 - https://software.intel.com/sites/default/files/11MIC42_How_to_Use_MKL_Automatic_Offload_0.pdf
 - <https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

3.Mode **symétrique**

- Un calcul dont les processus MPI s'exécutent sur CPU (hôte) et sur un Phi

```
mpirun -host localhost -n 20 ./code.cpu : -host mic0 -n 120 ./code.mic
```

Modèles de programmation (3)

Mode «offload»

- Directives interprétées par compilateur Intel

```
program hello
use omp_lib
implicit none
real*8 sum; integer :: i,n,nt

n=2000000000
sum=0.0d0

!DEC$ offload begin target(mic:0)

!$omp parallel do reduction(+:sum)
do i =1,n
    sum = sum + i
enddo
nt=omp_get_max_threads()
#ifdef __MIC__
    print*, "Hello MIC reduction",sum," threads:",nt
#else
    print*, "Hello CPU reduction",sum," threads:",nt
#endif

!DEC$ end offload

end program
```

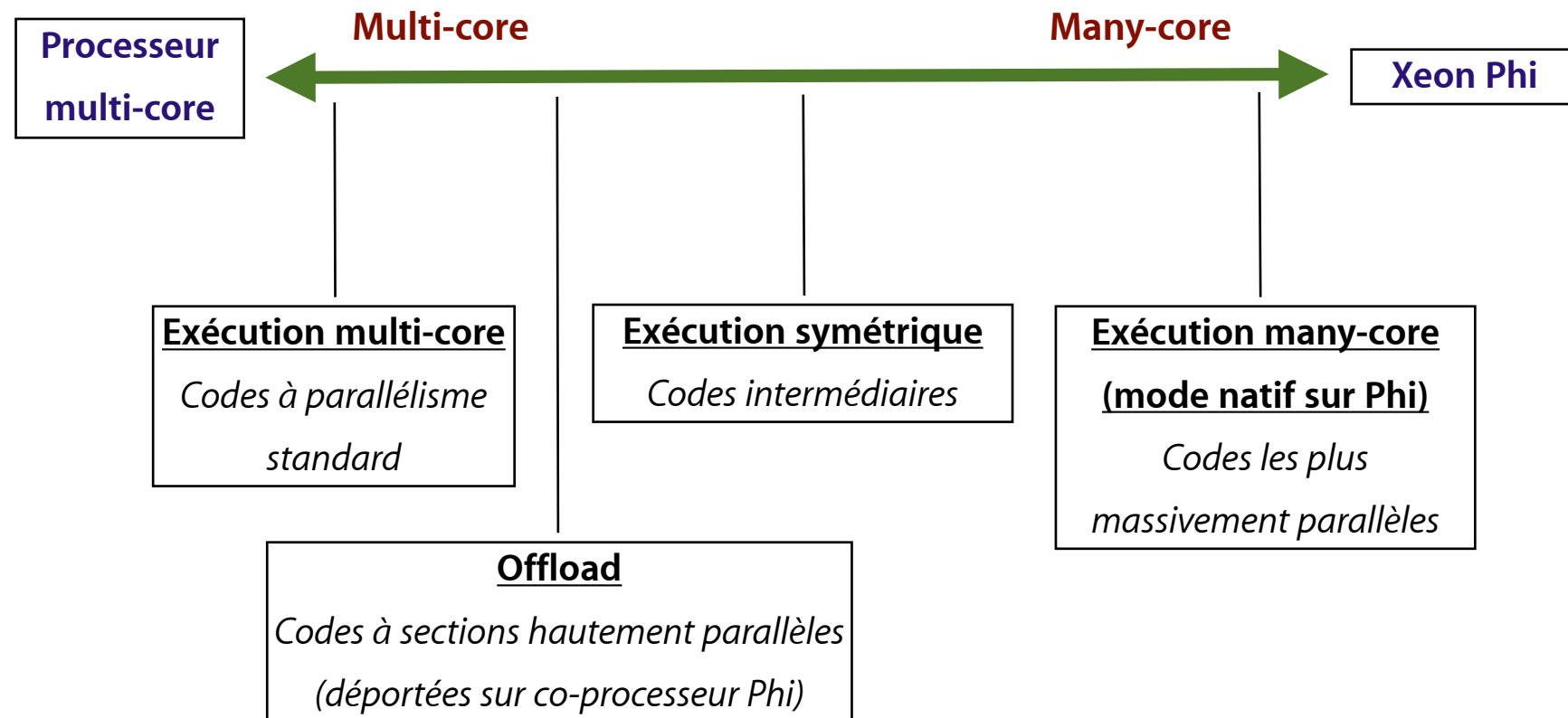
```
#include <omp.h>
int main(){
double sum; int i,n, nt;

n=2000000000;
sum=0.0e0;

#pragma offload target(mic:0)
{
    #pragma omp parallel for reduction(+:sum)
    for(i=1;i<=n;i++){
        sum = sum + i;
    }
    nt = omp_get_max_threads();
    #ifdef __MIC__
        printf("Hello MIC reduction %f threads: %d\n",sum,nt);
    #else
        printf("Hello CPU reduction %f threads: %d\n",sum,nt);
    #endif
}
}
```


Architectures et programmations parallèles

Multi-core / many-core (futur proche)



Applications scientifiques de référence

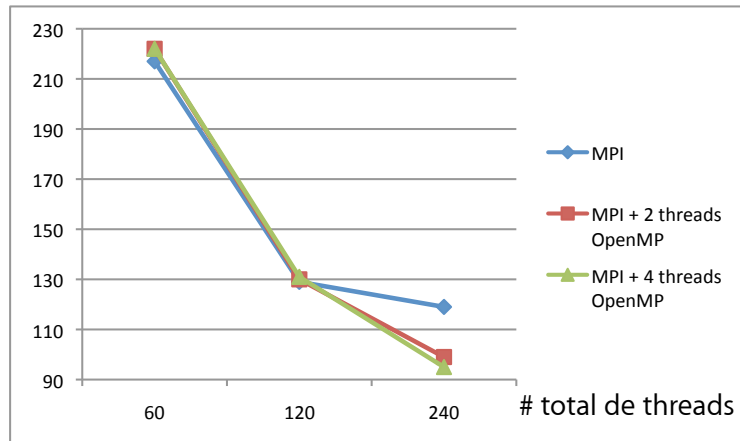
Portées ou en cours de portage sur Xeon Phi

- <https://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-applications-and-solutions-catalog>
 - Climat : WRF
 - Chimie : GAMESS (en cours), NWChem
 - Dynamique moléculaire : AMBER, DL-poly (en cours), LAMMPS, NAMD
 - Mécanique : ANSYS Mechanical, ABAQUS (en cours)
 - CFD : OpenFOAM (en cours)
 - Etc.

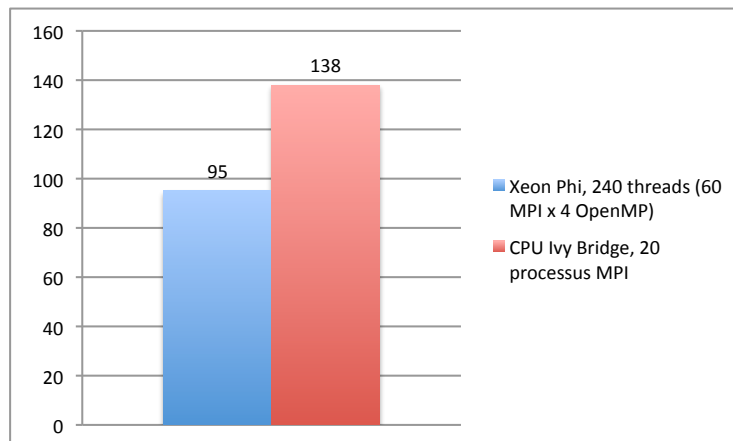
Test de noyaux

Poisson 2D (en natif sur Phi)

Temps (s) pour 4000 itérations SOR

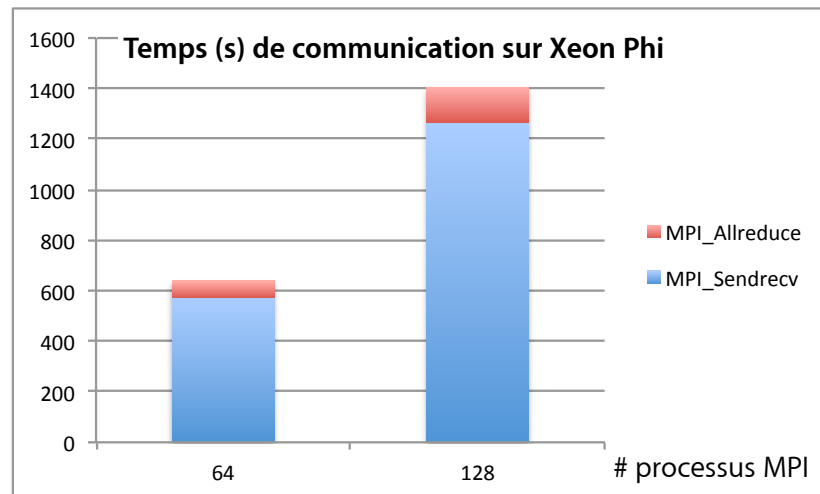
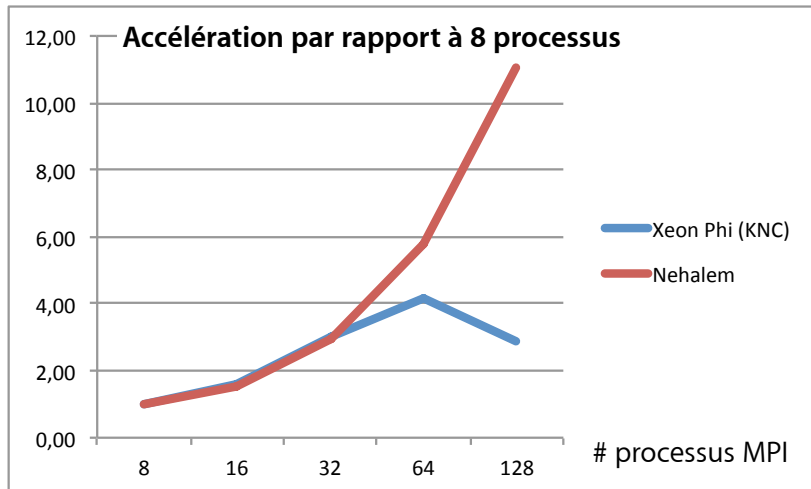


Meilleure performance (temps (s)) sur un Phi et son serveur hôte (Ivy Bridge)



- Différences finies, solveur SOR, maillage 8192 x 8192
- MPI + 2 threads OpenMP
 - 17% plus rapide que MPI
- MPI + 4 threads OpenMP
 - 20% plus rapide que MPI
- Pour ce maillage (8192 x 8192, ~ 500 Mo de mémoire)
 - un Xeon Phi (KNC) est 45% plus rapide que le serveur hôte (20 cœurs Ivy Bridge)

Maillage 128x128x128, séparation de phases, 0,5 s



Test d'applications

ARCHER (en natif sur Phi)

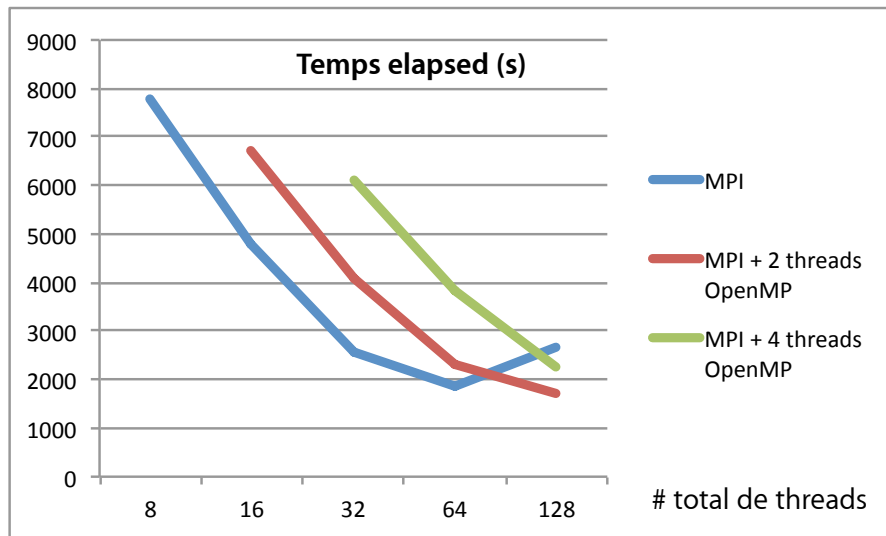
- Auteurs : A. Berlemont, T. Ménard, CORIA UMR 6614
- Code MPI de mécanique des fluides diphasique (suivi d'interface en méthode level set)
- Testé tel quel sur Xeon Phi
 - Chute de performance au-delà de 64 processus
 - Profilage avec Intel MPI

```
export I_MPI_STATS=ipm
export I_MPI_STATS_FILE=mpistats.out
```

Test d'applications (2)

ARCHER (en natif sur Phi) (2)

Maillage 128x128x128,
séparation de phases, 0,5 s



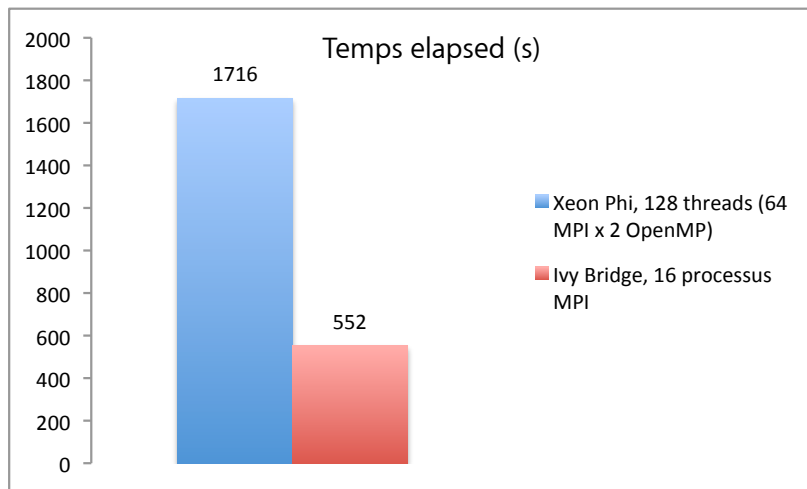
- Ajout d'OpenMP
 - Solveur MGCG (~ 50% d'ARCHER)
 - 64 MPI x 2 threads OpenMP
 - 8% de gain par rapport à la meilleure performance MPI (64)

Test d'applications (3)

ARCHER (en natif sur Phi) (3)

Maillage 128x128x128 (4,5 Go de mémoire)

Séparation de phases, 0,5 s



- Pour ARCHER avec ce cas d'utilisation et ce maillage (~ 5 Go de mémoire)
 - Le serveur hôte est 3 fois plus rapide qu'un Xeon Phi (KNC)

Xeon Phi de 2016

Knights Landing (KNL)

- Informations publiques
 - <https://software.intel.com/en-us/articles/what-disclosures-has-intel-made-about-knights-landing>
- Caractéristiques
 - > 3 TFlop/s double précision
 - \geq 60 cœurs (x 4 threads)
 - Jeu d'instructions AVX-512 (32 instructions par cycle)
 - Jusqu'à 16 GB de mémoire (MCDRAM) embarquée
 - haut débit (400 GB/s), mémoire cache ou allouable par code

Xeon Phi de 2016 (2)

Knights Landing (KNL)

- **Version processeur** (1 KNL par nœud de calcul)
 - support de 384 GB de mémoire (DDR4) avec 6 canaux
 - Bus d'entrées-sorties : jusqu'à 36 voies PCIe Gen3
 - Contrôleur réseau intégré Intel® Omni-Path
- **Version co-processeur**
 - Connexion PCIe entre hôte et co-processeur

Perspectives

Xeon Phi KNL

- Plate-forme d'expérimentation KNL en 2016 pour le Pôle Régional de Modélisation Numérique (PRMN)
 - Version processeur
 - Version co-processeur
- Formations utilisateurs



Centre de Ressources Informatique de Haute-Normandie