



# ADT K'STAR

C2S@Exa

Thierry Gautier  
INRIA MOAIS - Grenoble

# Un ordinateur (aujourd'hui)



processeur

+



mémoire

+



accélérateur

+

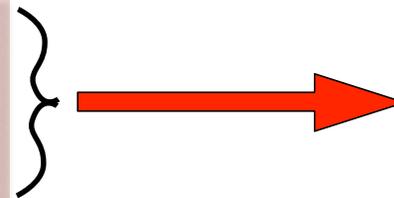


# Programmation de ces architectures

- Réseau haute performance + multi-cœurs + accélérateurs

- Pile logicielle : vers une uniformisation

- Network: infiniband, réseau propriétaire = MPI
- Multicores: OpenMP, Pthread, Ad-Hoc libs
- Accelerators: Cuda, OpenCL, OpenACC
- SIMD Units: Compiler or extension



**Cible d'OpenMP 4.0**  
**ARB : Architecture**  
**Review Board**

- Objet considéré ici = le nœud Multi-CPU + Multi-accélérateurs

# Challenges

- **Comment ?**

- ▶ garantir un haut niveau de performance
- ▶ assurer la portabilité des performances sur des architectures variées

- **2 points clés**

- ▶ **Scheduling**

- garantir les performances, robustesse

- ▶ **Support exécutif**

- surcoût, exploitation des différents niveaux de parallélisme, scalability

# Environnements existants

- **Compilateurs OpenMP**

- ▶ **GCC / libGOMP**
- ▶ **Intel / libiomp5**
- ▶ **CLANG-OMP / libiomp5**
- ▶ **PGI, ...**
- ▶ **Support limité pour les architectures hétérogènes**
  - norme récente 07/2013
  - Intel ICC OpenMP

- **Environnements logiciels**

- ▶ **OMPSS [BSC]**
  - compilateur (mercurium) + support exécutif (nanos)
- ▶ **INRIA**
  - StarPU [EPI RUNTIME/STORM]
  - XKaapi [EPI MOAIS]
- ▶ **.. FLAME**

# ADT K'STAR <http://kstar.gforge.inria.fr>

- **Objectif**

- ▶ **Compilateur « OpenMP » pour l'exploitation des architectures hétérogènes**
  - OpenMP-4.0
  - C, C++ et Fortran
- ▶ **Supports exécutifs cibles : XKaapi, StarPU**

- **Participants**

- ▶ **EPI MOAIS**

- T. Gautier, coordinateur, F. Broquedis
- P. Brunet (ADT K'STAR 10/2013-10/2015),
- P. Virouleau (ADT KAWAH - 10/2014)

- ▶ **EPI RUNTIME/STORM**

- O. Aumage, co-responsable, S. Thibault
- N. Furmento
- S. Pitoiset

# ADT K'STAR <http://kstar.gforge.inria.fr>

Application

OpenMP API + Runtime library

XKA-API

- locality aware work stealing

C C C C

G G G G

StarPU

- HEFT scheduler

C C C C

G G G G

# Technologie

- **Base clang-omp**

- **Support OpenMP par Intel**
- **Compilateur robuste C/C++**
- **Norme OpenMP-3.1 complète**
- **Norme OpenMP-4.0**
  - tâches avec dépendances
  - target (accélérateur)

- **Compilateur source-à-source**

- Ré-écriture après les passes d'analyse syntaxique et sémantique
- Insertion des appels aux supports exécutifs XKaapi et StarPU

- **Compilation finale par un compilateur natif C ou C++**

- **déport de certaines constructions OpenMP ne nécessitant pas de support à l'exécution**
  - #pragma omp atomic
  - #pragma omp simd...

# Status

- <http://kstar.gforge.inria.fr/#!features.md>
- **Travaux en cours sur le traitement des clauses « target »**
  - déport des calculs sur accélérateur
  - exploitation des modèles d'exécution hétérogène de XKaapi et StarPU
  - prototype fonctionnel
    - code académique, factorisation de Cholesky
- **Tests/Applications**
  - **GCC OpenMP testsuite**
  - **Benchmarks : Kastors**
    - <http://kastors.gforge.inria.fr>
  - **ScalFMM**
    - EPI HiePACS

# Extensions à OpenMP

- **Objectifs**

- Tests des fonctionnalités sur des applications réelles
- Force de proposition pour OpenMP

- **Extensions proposées**

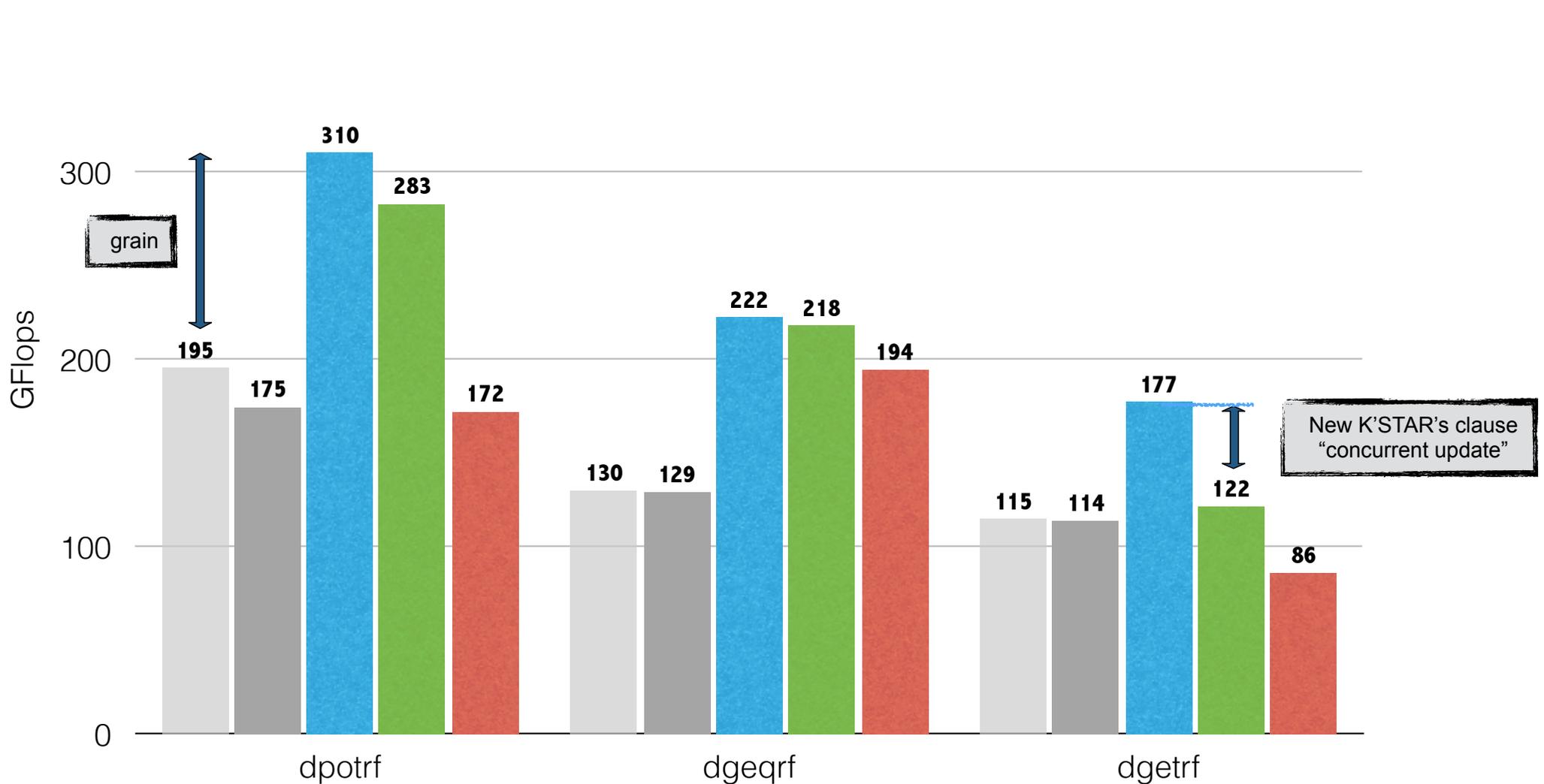
- **Scheduler de boucle**
  - Adaptive [XKaapi, IWOMP 2013]
- **Accès « concurrent »**
  - Plus de parallélisme, [IWOMP2014]

# Kastors <http://kastors.gforge.inria.fr>

- **Ensemble de benchmarks pour OpenMP / tâches avec dépendances**
  - **Diffusion [IWOMP2014]**
    - très bien évalué
  - **Strassen, Jacobi, SparseLU [BOTS]**
  - **Factorisations : LU, QR, Cholesky [PLASMA]**
- **Objectif**
  - **Tests de compilateurs et des supports exécutifs**
  - **Base pour la diffusion d'extensions à la norme**
- **Résultats**
  - **OpenMP est une alternative à des bibliothèques spécialisées**
    - typiquement QUARK pour PLASMA
  - **Des extensions sont nécessaires**
    - écriture concurrente, souplesse sur l'expression des dépendances

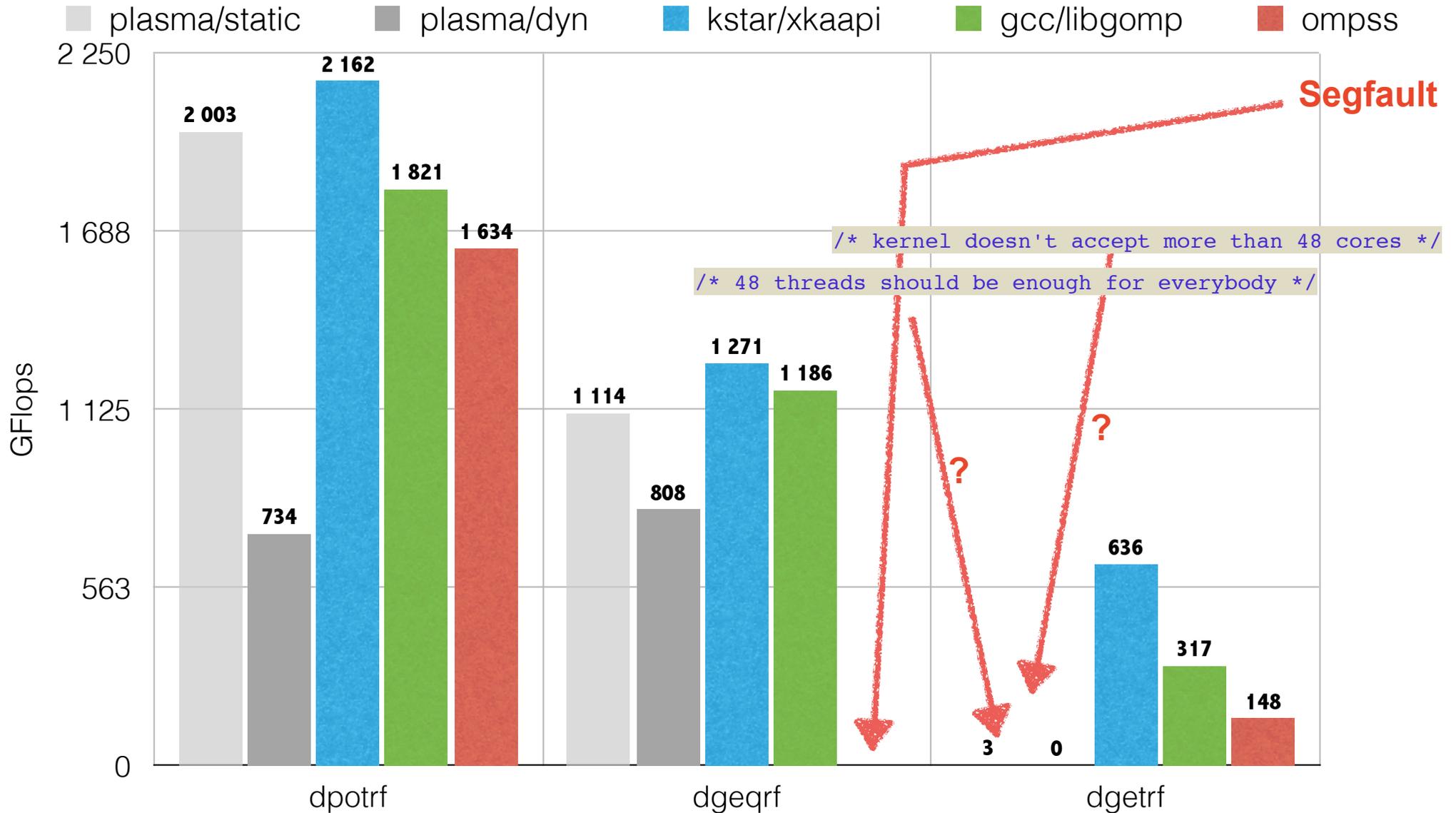
# KASTORS - small matrices

■ plasma/static   ■ plasma/dyn   ■ kstar/xkaapi   ■ gcc/libgomp   ■ ompss



- HPAC 32 cores CPU E5-4620 0 @ 2.20GHz
- matrix 4096x4096, BS=128 (+ IB=64 for QR)

# KASTORS - 192 cores NUMA



• IDCHIRE@JFM matrix 32768 x 32768, BS=512

‣ IB=64 for QR KSTAR/StarPU

‣ IB=128 for QR KSTAR/XKaapi and GCC

# Conclusion

- **1 même API C ou C++ pour 2 supports exécutifs INRIA**
  - XKaapi ou StarPU
  - version multi-CPU OpenMP-4.0
  - ➔ meilleure diffusion de ces logiciels
- **Alternative pérenne pour l'écriture d'algorithmes sur multi-cœurs**
  - algèbre linéaire dense, type PLASMA
  - ➔ extensions nécessaires

# Perspectives

- **Intégration des modèles d'exécution hétérogène de XKaapi / StarPU**
  - prototype fonctionnel
- **Extensions à la norme**
  - aspect NUMA
- **Couplage à des outils d'analyse des performances**
  - Interface en cours de normalisation dans ARB
- **Fortran**
  - portée importante en HPC

