

# Le Grand Chambardement Logiciel

"Realizing the physics programs of the planned and/or upgraded HEP experiments over the next 10 years will require the HEP community to address a number of challenges in the area of software and computing. It is expected that the computing models will need to evolve and **a significant software upgrade is required** to prepare..."

*HEP Software Foundation, Community White Paper*

# Avant-propos

- Conflits d'intérêts
  - un biais « Physique des Particules »
  - un biais « Paris-Saclay »
  - un biais « C++ & Python »
- Remerciements
  - Pierre Kestener, Maison de la Simulation
  - Olivier Aumage, INRIA
  - Hartmut Kaiser, Louisiana State University
  - Thomas Heller, Friedrich-Alexander University

# Le matériel change

- Toujours plus de coeurs : face au mur de fréquence, les fondeurs multiplient les coeurs, beaucoup plus vite que la mémoire.
- Des instructions toujours plus vectorielles.
  - Notre modèle de calcul trivialement distribué devient inapplicable.
- Toujours plus hétérogènes : après les GPUs, introduction des FPGAs aux côtés des CPUs.

# Nos calculs doivent changer

- Aujourd'hui
  - Des cadruciels multi-tâches (basés sur Intel TBB).
  - Les physiciens sommés d'écrire des codes "thread-safe".
  - Des accélérateurs dans le déclenchement en ligne.
- Demain
  - Des instructions de bas-niveau vectorisées.
  - Des flots de données asynchrones.

# Une offre logicielle pléthorique

- Applications clef-en-main : Matlab, Mathematica, ...
- Bibliothèques : CuBLAS, ArrayFire, Thrust, **Kokkos**...
- Directives : OpenACC, OpenMP 4.
- Intégration au langage : **SyCL**, Go, **C++20**.
- Pilotes matériel : CUDA (Nvidia), OpenCL (les autres).
- A l'exécution : StarPU, **HPX**.
- Abstractions de haut niveau : DSLs et **EDSLs**.

# Ce qui les différencie

- Niveau de support des différentes formes de parallélisme : vectorisation, accélérateurs, mémoire partagée ou distribuée, co-routines, multi-thread, multi-processus, multi-noeuds, multi-sites...
- Niveau d'intégration au langage de programmation : bibliothèques, directives, extensions, instructions natives.
- Optimisations statiques (à la compilation) vs dynamique (à l'exécution).

# Ce qui les rassemble

- Des technologies issues du milieu HPC, focalisées sur l'exploitation optimale d'un noeud de calcul.
- A marier avec MPI pour le parallélisme muti-noeuds à mémoire distribuée, à l'échelle d'un cluster.
- A marier avec les technologies HTC et "Big Data" pour un parallélisme multi-site.

# Des concepts qui (ré)émergent

- Repenser les données
  - Du AoS au SoA, voire au AoSoA.
  - Faire la chasse aux pointeurs, favoriser les POD.
  - Nécessité de placer, répliquer, aligner ses données (mémoire inhomogène, caches).
- Repenser les algorithmes
  - Privilégier la force brute, recalculer.
  - Concepts de “futurs” et d’ “exécuteurs”.
- Programmer **fonctionnel** et **thread-safe**
  - Retour à de pures fonctions avec des entrées et des sorties explicitées.
  - Elimination des if, mutables, globales, calculs “cachés” en tout genre.

# Stratégie

- Graal : un code source unique de haut niveau, qui s'adapte en temps réel au matériel disponible, sans sacrifier la performance.
- Un pari possible, 100% C++ : des EDSLs pour des descriptions de haut niveau, et C++20 fera la synthèse des travaux autour de SyCL, HPX, Kokkos.
- Quelle que soient les incertitudes technologiques :
  - toute réécriture du code améliore sa lisibilité et ses performances, même si on continue de l'exécuter en mode séquentiel,
  - beaucoup de concepts très similaires se retrouvent dans toutes les alternatives technologiques,
  - l'essentiel n'est pas dans l'apprentissage des nouvelles syntaxes, mais dans la refonte des structures de données et des algorithmes.

# Propositions d'actions IN2P3

"...les logiciels utilisés majoritairement dans nos disciplines sont constitués de modules séquentiels et mis en exécution à l'aide de techniques de parallélisme embarrassant basées sur la distribution des événements. Ce modèle tend à devenir obsolète et un rapprochement avec les communautés du calcul intensif devient indispensable pour penser la refonte de notre patrimoine logiciel. Ce rapprochement doit être soutenu par l'IN2P3, sous la forme d'actions à discuter notamment avec le réseau des informaticiens..."

*Colloque Calcul IN2P3, juin 2015*

# Master-Projet "Programmation Parallèle et Performance du Calcul"

- Pour que les acteurs IN2P3 soit reconnus et valorisés
- Pour faciliter le lien avec les organismes et laboratoires d'informatique, de calcul intensif, et attirer des stagiaires des M2 de calcul.

# Idées de projets

- Parallélisation du framework Gaudi.
  - Nouvelles approches pour le tracking des expériences LHC.
  - Performance portable et durable.
  - Parallelism carpentry.
- 
- Précision du calcul, nombres aléatoires, portabilité et validation des résultats.
  - Expérimentation des nouveaux matériels de calcul et des outils de debugging et profilage en contexte parallèle.
  - Calcul hybride et utilisation de super-calculateurs.
  - Simulation de systèmes de calcul distribué et intensif.

# Parallelism Carpentry

- Sur le modèle testé à l'occasion des dernières Journées Informatique de l'institut, construire une base de tutoriels d'auto-formation exécutables sur des images Docker.
- Compléter par des tutoriels nécessitant du matériel spécifiques autour des plate-formes adéquats.
- Organiser des “hackathons” en face à face autour de ces tutoriels.

# Trois petits photons

- Une mini-application de simulation mise à disposition par Vincent Lafage (IPNO), et remaniée par Gilles Grasseau (LLR) à l'occasion de l'école informatique "Parallélisme sur matériel hétérogène"
- Peut-elle servir de banc d'essai pour expérimenter de nouveaux langages, comparer des technologies, enseigner ?
- <https://bitbucket.org/bixente/3photons/>

# GridCL : un cluster pour la R&D

- GridCL a été financé par le labex P2IO pour disposer de matériel manycore pour la R&D, notamment de K20 (NVidia) et de Xeon Phi (Intel).
- Sur le modèle du TECHLAB du CERN, l'IN2P3 pourrait reprendre et prolonger l'effort de GridCL pour que les laboratoires de l'institut disposent de matériel de R&D.

# Groupe de discussion RI3

- Veiller et échanger les retours d'expérience
- Recenser, développer et partager les ressources matérielles et logicielles
  - Cluster GPU (Centre de calcul)
  - GridCL (Labex P2IO)
- Avoir une rencontre annuelle, telle que celle d'aujourd'hui.

# Réserve

# Des problèmes spécifiques

- Repenser la génération des nombres aléatoires.
- Difficultés pour la validation logicielle et physique
  - Des affichages qui viennent dans le désordre.
  - Des résultats qui ne sont jamais exactement les mêmes.