



Ecole « Méthodologie et outils d'optimisation
en développement logiciel »
Fréjus, 6-10 Février 2012



Architect of an Open World™

Benchmarking

A first step toward code optimization.

Ludovic Saugé, PhD

Ingénieur Applications Senior

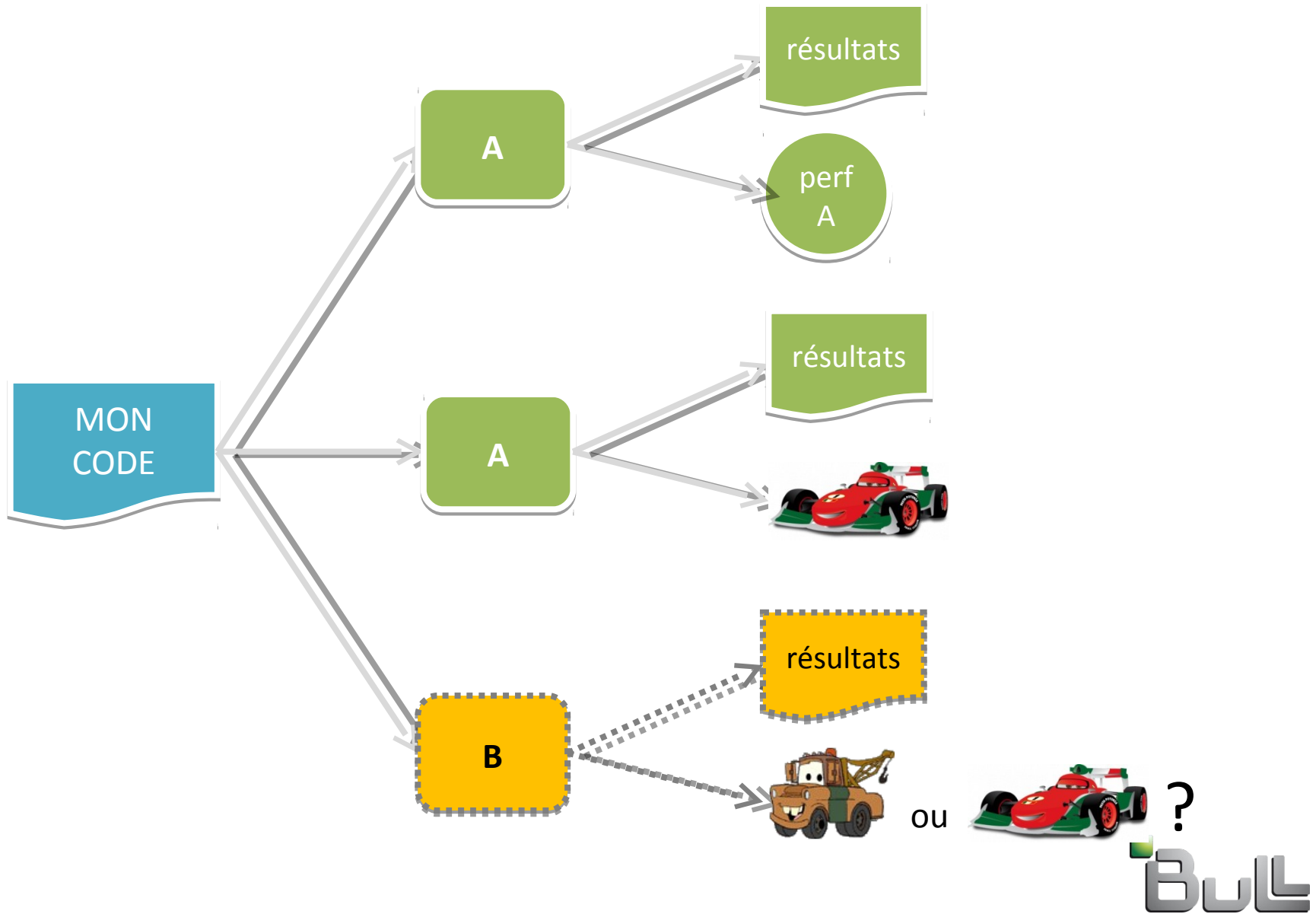
Equipe Applications et Performances

Bull Extreme Computing BU

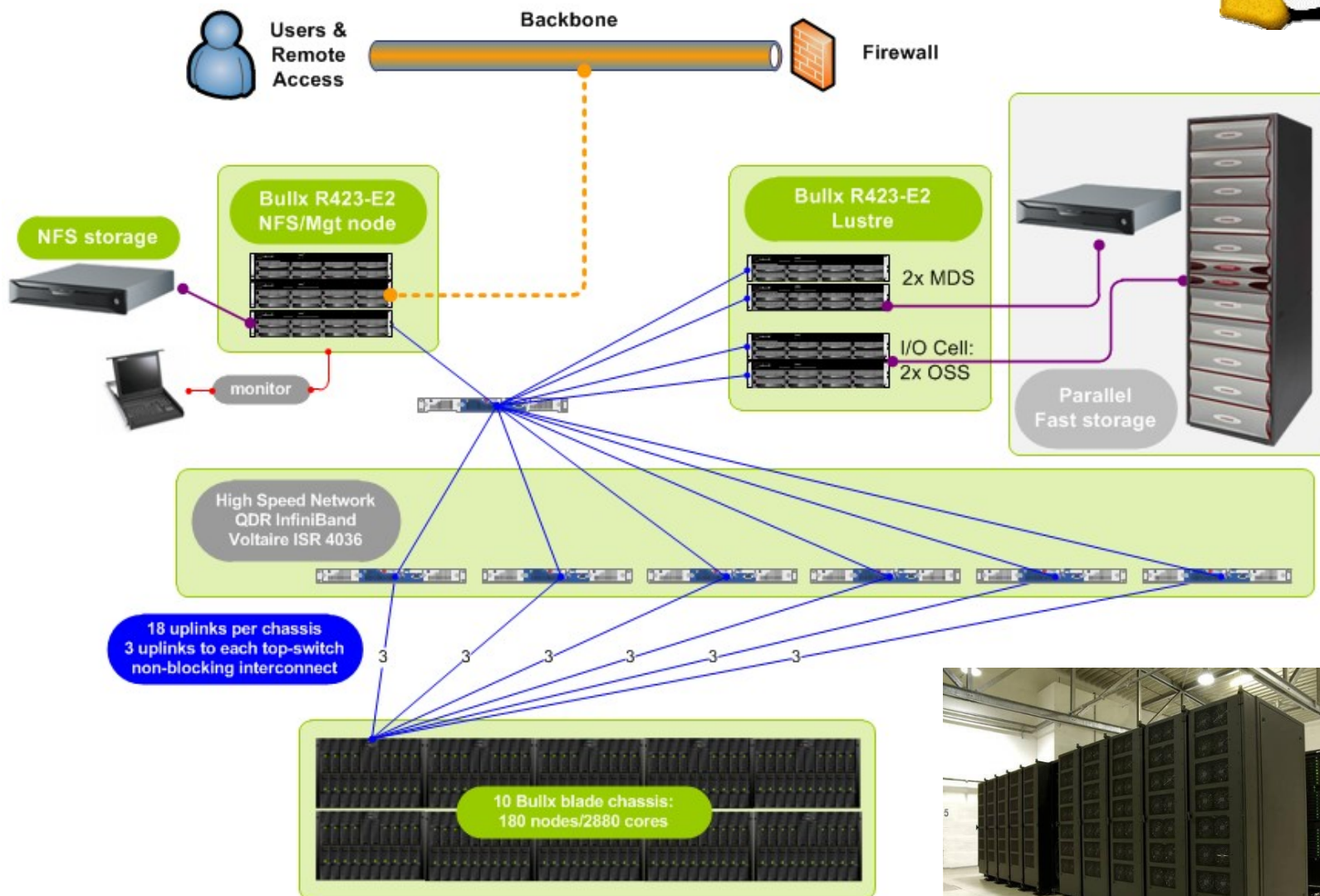
Grenoble

- Introduction
 - Pourquoi benchmarker ?
- Élément centraux
 - Elements de micro-architectures
 - Processeurs
 - Hiérarchie Mémoire
 - Cache, RAM, ...
- Benchmarks synthétiques
 - HPL, HPCC, SPEC CPU2006
- Interconnect et MPI
- Temps de restitution

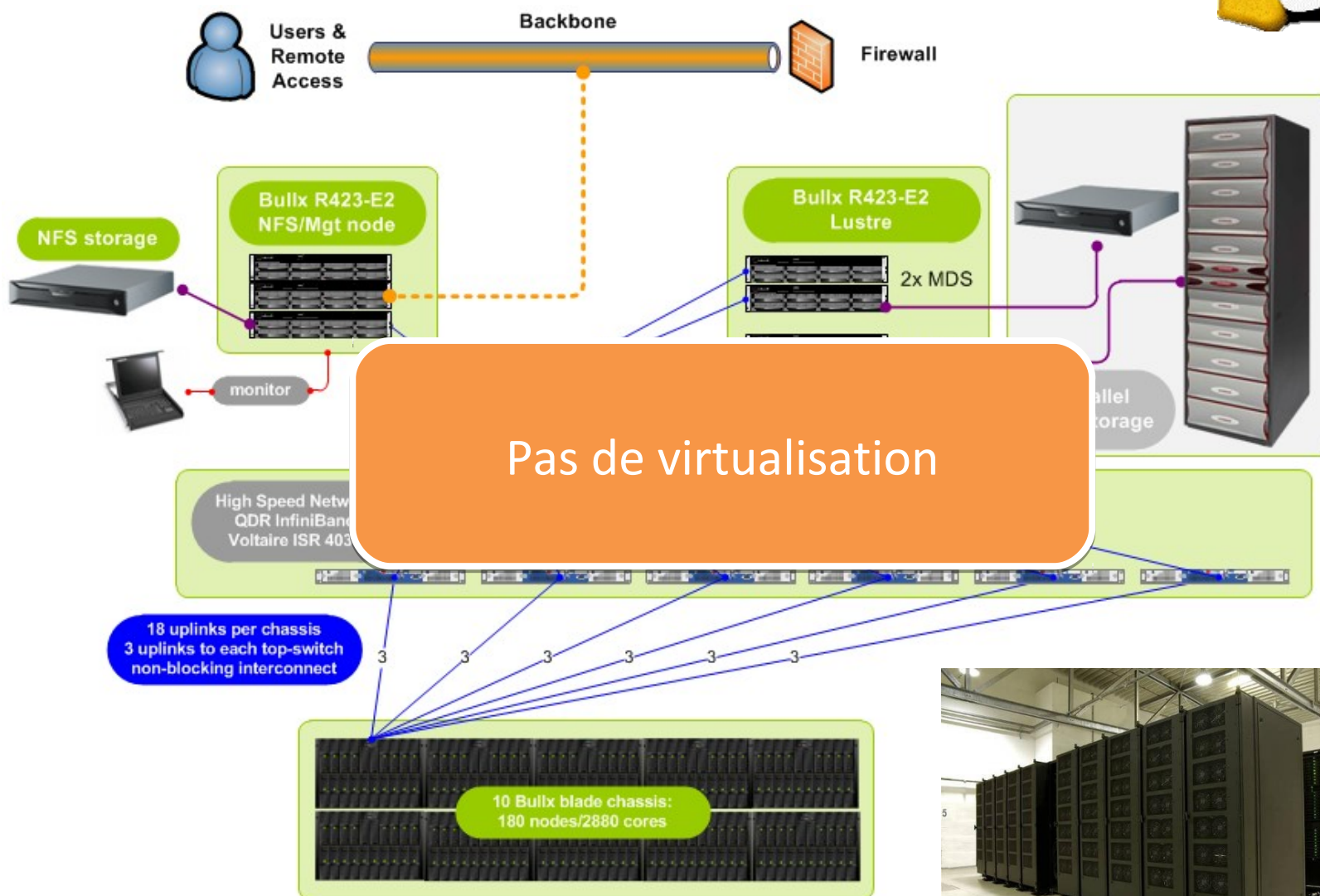
Introduction



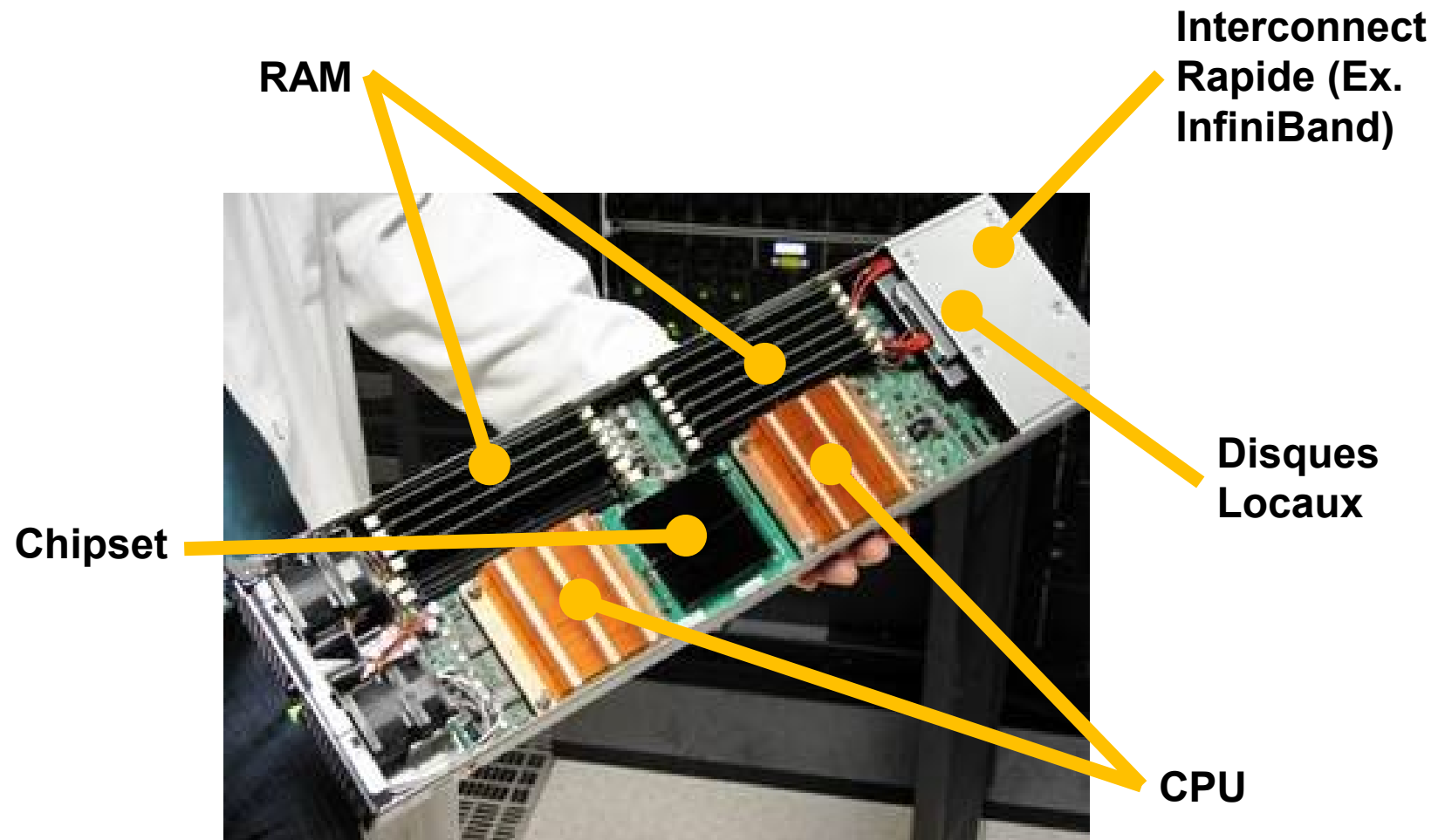
Une solution HPC



Une solution HPC



Elements matériels



Elements matériels

RAM

Interconnect
Rapide (Ex.
InfiniBand)



Quelle est l'influence de tous ces éléments sur les performances et en particulier sur les performances de MON code ?



CPU

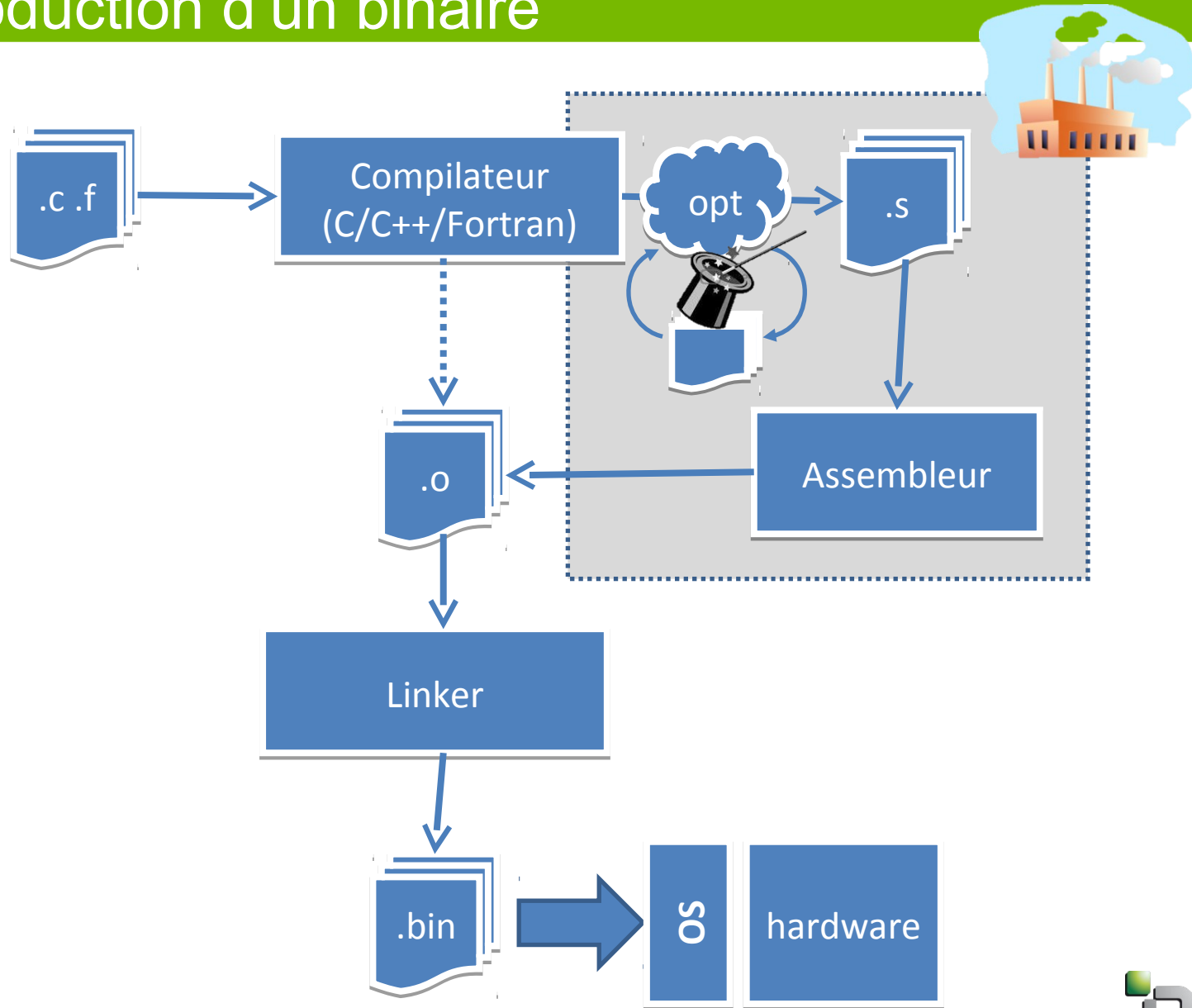
Element d'un cluster

- Réseau rapide d'interconnexion (interconnect)
 - Topologie
 - Routage
- Ensemble de nœuds de calculs
 - CPU
 - Unité de calculs (coeur) ; largeur, fréquence, instructions/cycle ...
 - Hiérarchie de caches ; nombre, quantité, organisation ...
 - Mémoire
 - Organisation/Topologie : UMA, NUMA ...
 - Technologie, fréquence (latence, bande passante,...)
 - Chipset (north/south bridge)
 - Une communication entre tout ça (bus, liens)
- Stockage
 - Les baies (type de disques, bande passante,...)
 - Systèmes de fichiers

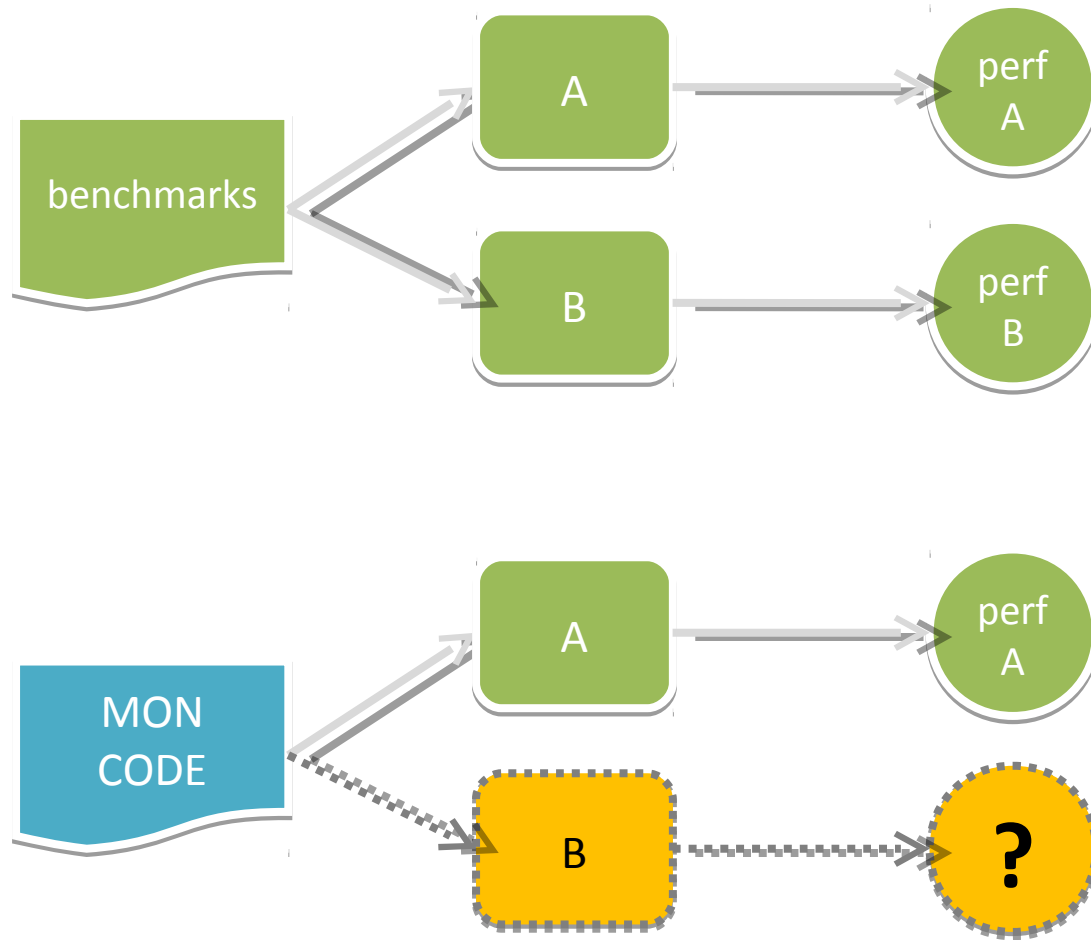
Pourquoi sommes nous là ?

- Comment le tel ou tel choix d'architecture affecte les performances des machines ?
 - Ces effets sont-ils mesurables ? Comment les mesurer ?
 - Ex. interconnect
 - Latence, bande passante
 - Performance MPI
 - Code de l'utilisateur ...
- Quelles conséquences sur les performances de mon code ?
 - Mon code sera-t-il pénaliser sur ce type d'architecture ?
 - Si oui, est-ce que je peux faire quelque chose ?
 - Quelle est l'architecture la plus adaptée ?
- Puis-je prédire les performances de mon code sur une machine B connaissant ces performances sur la machine A ?
 - Optimiser sur mon laptop ou machine de bureau a-t-il un sens ?
 - Quel benchmark choisir ?
- Est-ce que seul le hardware à son importance ?
 - Non

Production d'un binaire



Mes grandes étapes de l'optimisation



Mes grandes étapes de l'optimisation

- Portage du code (et des librairies)
- Profiling
 - Y'a-t-il, quels sont les “**hot spots**” ? Quels sont les “**bottlenecks**” ?
 - Outils disponibles : l'incontournable `gprof`
 - À grains plus fins : sur l'étude des **compteurs hardware**.
Limitations:
 - nécessite parfois les droits SU (il faudra jouer de la configuration `sudo` ...)
 - parfois difficile à manier dans un environnement cluster avec un batch
 - plutôt à utiliser sur le kernels



La connaissance des « bottlenecks », permet de se donner des objectifs (atteignables) en terme de performances, tout au moins des limites hautes ou basses . Elle nécessite de connaître les perfs de la machine cibles, et d'évaluer les perfs unitaires intéressantes pour évaluer cette cible en performance.

Mes grandes étapes de l'optimisation

- Extraction de kernels ?

- Noyau de calculs : code séquentiel : écriture de son **propre benchmark** (version standard, optimisées,...)
- Permet de travailler déjà sur son laptop (on fait beaucoup de chose sur son laptop ...)

- Analyse et Optimisations

- Passe d'abord par trouver les bonnes options du **compilateur** ou pragma
- Inspection du code intermédiaire (en assembleur)
 - Ne mord pas !
- Inspection du code en général
 - Re-ordonnancement, nettoyage ...
 - Changement d'algorithme ?

- Attention à l'optimisation à outrance :

- Elle rend le **code adhérent** à une plateforme
- Elle rend sa **maintenabilité** plus difficile (multiplication des architectures)
- Elle peut-être **contre productive** : elle obscurcit le code et trompe/rend inefficace les phases d'optimisation du compilateur
- Je parlerai plus facilement de “**bonne pratiques**” ou plutôt de comment éviter les mauvaises

Benchmarks

Qu'est-ce qu'un benchmark ?

- Wikipedia, the free encyclopedia



- "Benchmark (computing)"

In computing, a benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the **relative performance** of an object, normally by running a number of standard tests and trials against it. The term 'benchmark' is also mostly utilized for the purposes of elaborately-designed benchmarking programs themselves."

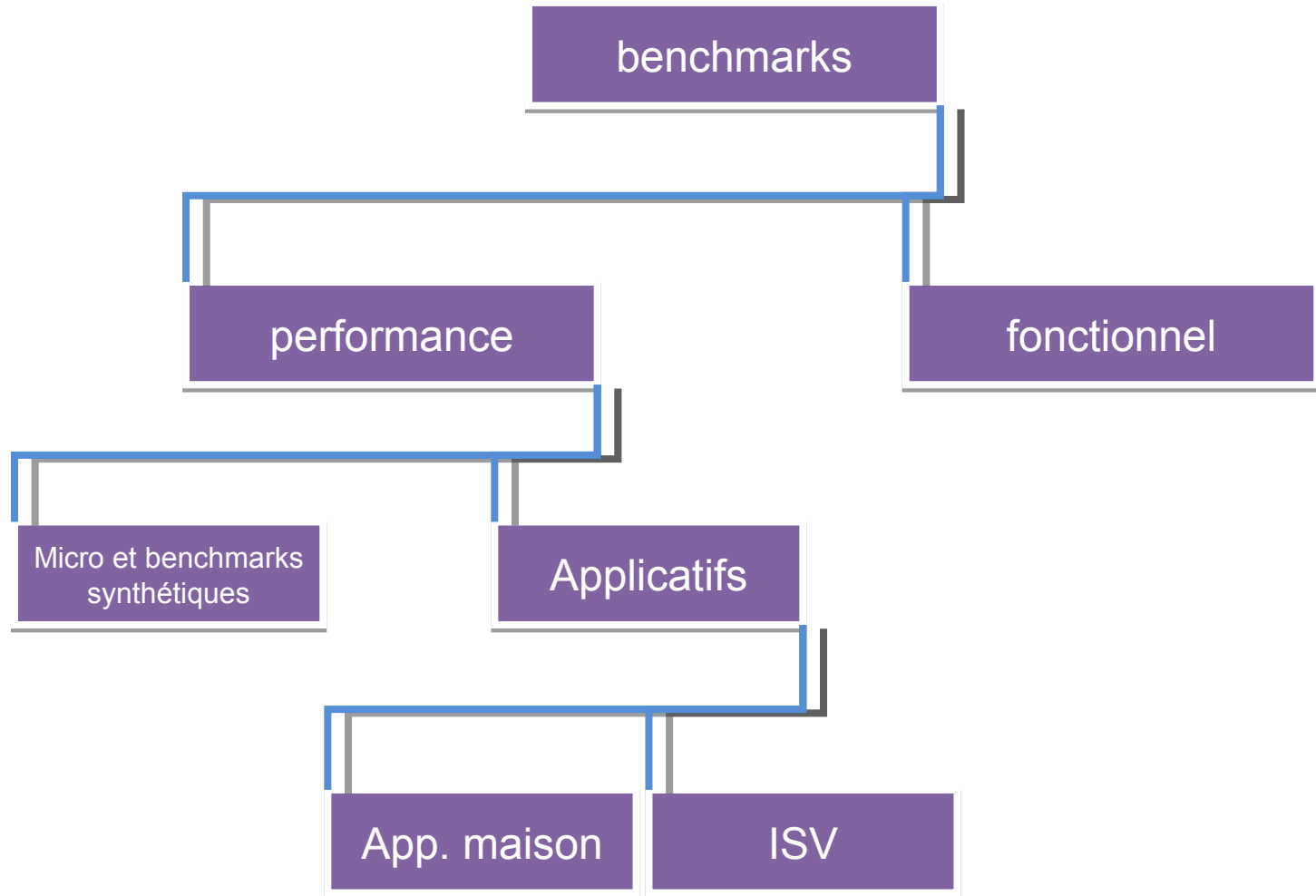
- Spec.org



- **Q2. What is a benchmark?**

A **benchmark** is "**a standard of measurement or evaluation**" (Webster's II Dictionary). A computer benchmark is typically a computer program that performs a strictly defined set of operations - a **workload** - and returns some form of result - a **metric** - describing how the tested computer performed. Computer benchmark metrics usually measure **speed**: how fast was the workload completed; or **throughput**: how many workload units per unit time were completed. Running the same computer benchmark on multiple computers allows a comparison to be made.

Benchmarks | taxonomie (très) simplifiée



Benchmarks

- Les micro-benchmarks et les benchmarks synthétiques fournissent une idée du comportement d'une solution élément à élément. C'est une référence, un point de comparaison entre les technologies
- Les applications sont là pour donner une idée du comportement **GLOBAL** d'une solution
 - La fin des fins d'une machine'est d'effectuer des calculs utilisateurs
 - Doivent être réalistes d'une situation de production.

Le fantasme du benchmarkeur

- A partir d'un certain nombre d'observables et des mesures obtenues à l'aide de μ -benchmarks, c'est prédire sur ce système le comportement d'une application et en particulier le temps de restitution comme une combinaison (linéaire) des mesures w_i .
- $T = a_1 w_1 + a_2 w_2 + a_3 w_3 + \dots + a_n w_n$
- Nécessite de connaître son code en profondeur et ses comportements et dépendances en fonction de ces observables \rightarrow détermination des a_i ...
- Est-ce que ça marche ? des fois ...
 - Nécessaire dans le cas des extrapolations
 - Petit nombre d'observables
ex. HPL

Benchmarks

- Qu'est-il intéressant de mesurer pour être sûr de faire le bon choix ?
- Y'a-t-il un seul indice de performance ?
 - Le temps de restitution d'une application.
- Existe-t-il le benchmark universel ?
 - Celui qui donne une valeur unique, facile à comparer ...
 - Celui qui tourne vite ...

HPL (Top500) ? SPEC CPU 2006 ? HPCC ?