



LABORATOIRE
D'AÉROLOGIE



Introduction à Visit

Philippe.Wautelet@aero.obs-mip.fr

CNRS - Laboratoire d'Aérodynamique

Atelier Visualisation in Situ / IMFT / 11 au 13 décembre 2017

1 **Contexte**

2 **Chaîne de traitement**

3 **Visit**

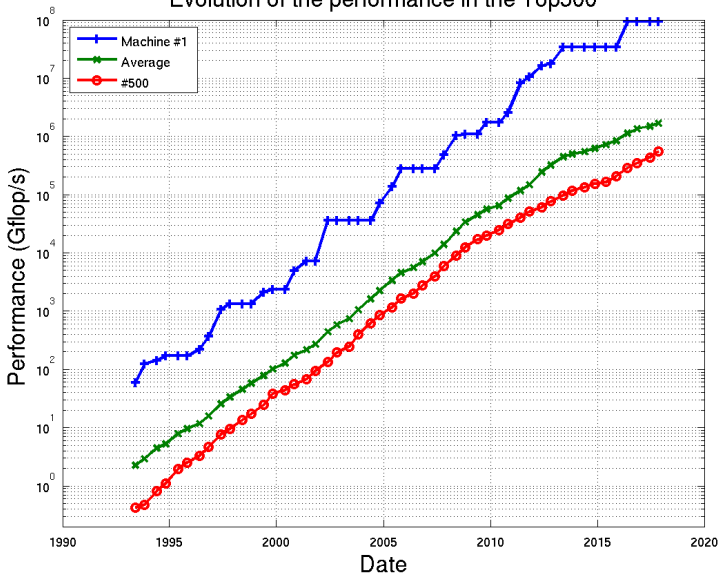
4 **Démonstrations**

Contexte

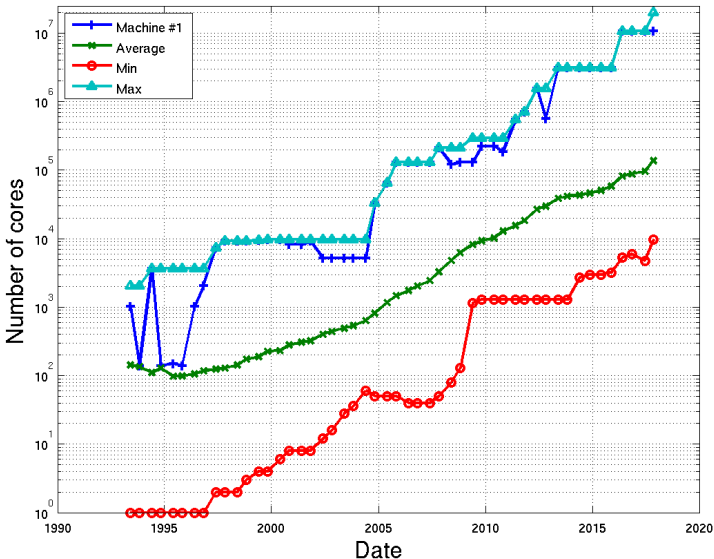
Evolution des calculateurs

- Parallélisme toujours plus massif
- Puissance par cœur augmentant lentement
- Mémoire par cœur limitée (même si possibilité d'avoir des nœuds "grosse mémoire")
- Complexification et augmentation du nombre de couches à traverser pour accéder aux données (transparent pour l'utilisateur mais impactant les performances)
- Débit vers les disques augmentant lentement
- Utilisation de plus en plus fréquente des SSDs (souvent comme niveau intermédiaire / cache dans les gros systèmes)

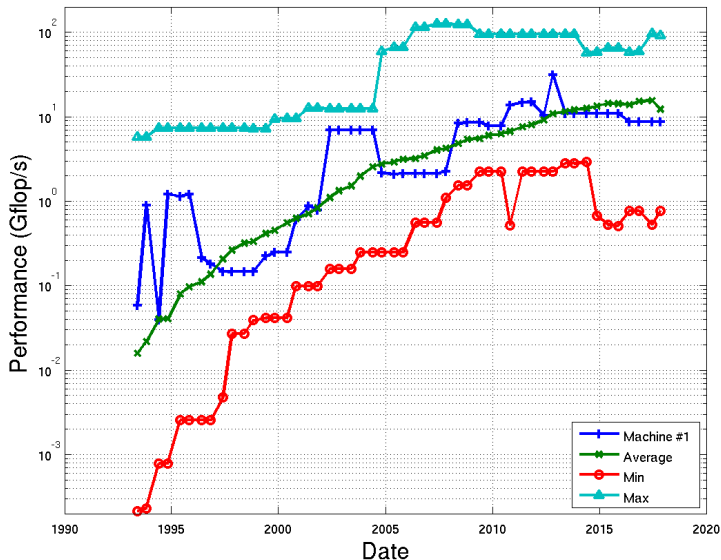
Evolution of the performance in the Top500



Evolution of the number of cores in the Top500



Evolution of the performance per core in the Top500



Masse de données (« big data »)

- Les données brutes ne peuvent plus être déplacées (raisonnablement)
- A consommer / transformer sur place (proximité entre lieux de production et transformation / consommation)
- Pertinence des données stockées
- Conservation long terme
- Pérennité des données (formats de fichiers...)
- Manque fréquent de parallélisme pour les phases de pré/post-traitement

Chaîne de traitement

Approches

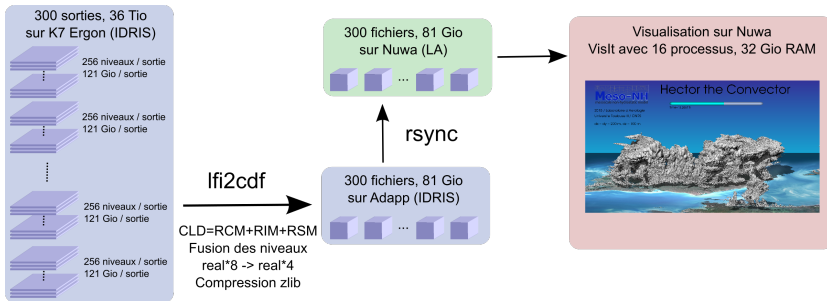
Pour les grosses simulations (plusieurs tera-octets de données à traiter) : 3 grandes approches.

- Visualisation en parallèle sur le lieu de production (mode client/serveur(s) ou déportée avec VNC/X2Go)
- Extraction (réduction) données -> transfert -> visualisation (en parallèle) sur station ou centre local
- Visualisation des données in situ, c'ad en interagissant directement avec la simulation et sans (nécessairement) conserver les données

Difficultés

- Extraction des données
- Formats de fichiers
- Choix outils de visualisation
- Performance réseau (débit mais aussi latence particulièrement pour la visualisation déportée)

Chaîne de traitement : exemple



Visit

Généralités

- VisIt est un outil interactif de visualisation et d'analyse de jeux de données
- VisIt a été originellement développé dans le cadre du projet ASCI du département de l'énergie (DOE) américain pour visualiser et analyser les résultats des simulations téraflopiques
- Première version distribuée en 2002 (développement/prototypage commencé en 2000)
- Distribué sous licence BSD (*Open Source*)
- Exécutables fournis pour Windows, Mac OS X, Linux (sources disponibles mais compilation pas toujours triviale)

Caractéristiques techniques

- Outil parallèle (peut être utilisé en séquentiel)
- Supporte le mode client(s)/serveur(s)
- Tire parti des accélérateurs graphiques (GPU) si disponibles
- Peut créer des animations
- Architecture à base de *plugins* permettant de créer des lecteurs de données, des opérateurs sur les données et de nouveaux tracés
- Actuellement, environ 120 lecteurs de données, 60 opérateurs et 20 tracés
- Peut être contrôlé par une interface Python (au lieu de l'interface graphique classique)
- Bibliothèque *libsims* pour l'instrumentation in situ

Démonstrations

Caractéristiques techniques

- Démo exemple Visit
- Démo Hector
- Vidéo Hector