

« Évolution des intergiciels & outils informatiques » Rencontres de Branville

Vincent Garonne

Laboratoire de l'Accélérateur Linéaire

22 mai 2006

Présentation du service informatique/Groupe Devdu



Sommaire

1. Les tendances observées
2. Les générations passées et actuelles de grilles
3. Les prochaines générations de grilles
4. Les nouveaux modèles de calcul et outils



Les tendances et futures problématiques

Résumé des tendances (voir présentation M. Jouvin)

- ▶ Vers des infrastructures massivement parallèles/distribuées
 - Ressources + petites, + puissantes, e.g Palm, multi-cœur
 - ↗ et omniprésence du réseau, e.g. WiFi Max
 - Données/informations massives et largement distribuées

Les possibles questions dans 10 ans...

- 1 Comment utiliser ces infrastructures pour les futures expériences de physique des particules, e.g. ILC ?
- 2 Quelles seront les nouveaux modèles de travail, techniques et outils dans ces environnements ?
- 3 **But de cette présentation : Essayer d'amener des éléments de réponses, poser des questions et esquisser le rôle futur du LAL**



Les tendances et futures problématiques

Résumé des tendances (voir présentation M. Jouvin)

- ▶ Vers des infrastructures massivement parallèles/distribuées
 - Ressources + petites, + puissantes, e.g Palm, multi-cœur
 - ↗ et omniprésence du réseau, e.g. WiFi Max
 - Données/informations massives et largement distribuées

Les possibles questions dans 10 ans...

- 1 Comment utiliser ces infrastructures pour les futures expériences de physique des particules, e.g. ILC ?
- 2 Quelles seront les nouveaux modèles de travail, techniques et outils dans ces environnements ?
- 3 **Méthode : « Pour voir le futur, il faut regarder derrière soi »**



Les différentes générations de grilles pour la HEP

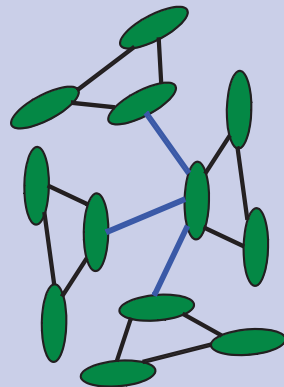
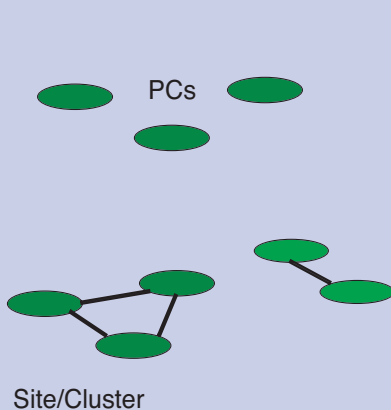
Intérêt « world wide grid » dès le début

- ▶ Les grilles de 1^{ère} génération
 - Approche client/serveur, basée sur le toolkit Globus 2
 - Identification des principales fonctionnalités
 - Exemples : DataGrid, Nordugrid et GRID3
- ▶ Les grilles de 2^{ème} génération
 - Architecture Orientée Services, basée sur OGSI/OGSA
 - Grilles de production
 - Exemples : EGEE, ARC et OSG
- ▶ Les grilles communautaires
 - Développées par les expériences de physique pour leurs besoins
 - Grille légère, en surcouche des grilles de production
 - Exemples : Alien, DIRAC et PANDA



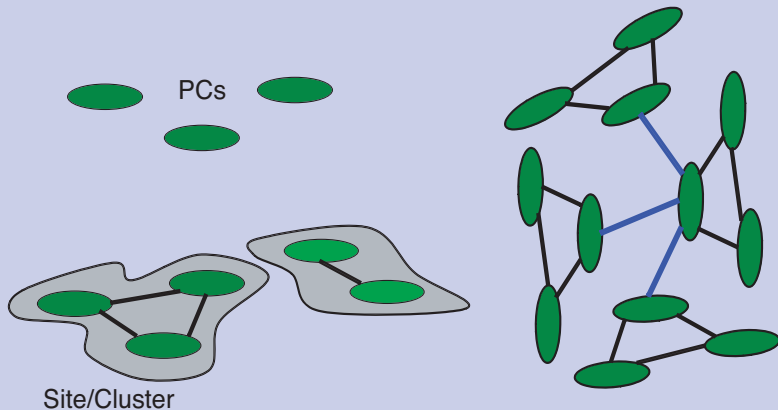
Les ressources physiques

Ressources distribuées



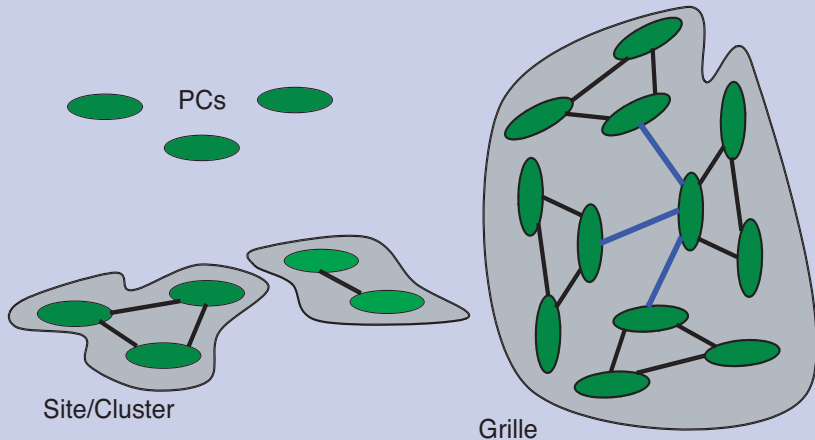
La couche logicielle locale

Ressources distribuées



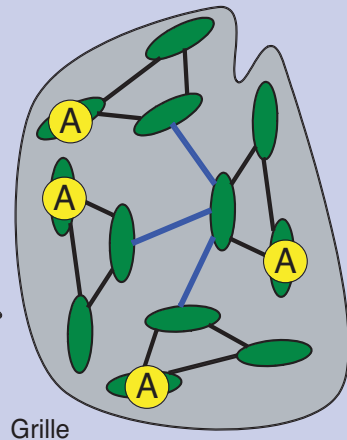
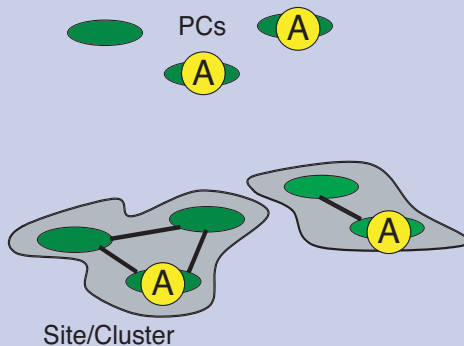
La couche logicielle globale : les grilles

Ressources distribuées



Les agents

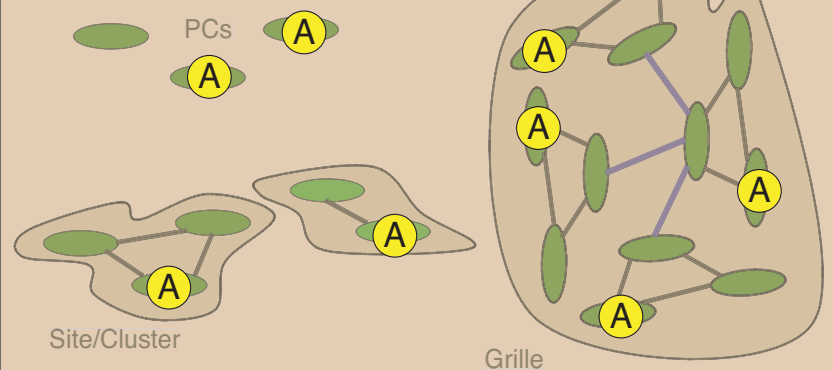
Ressources distribuées



La virtualisation communautaire

Ressources distribuées

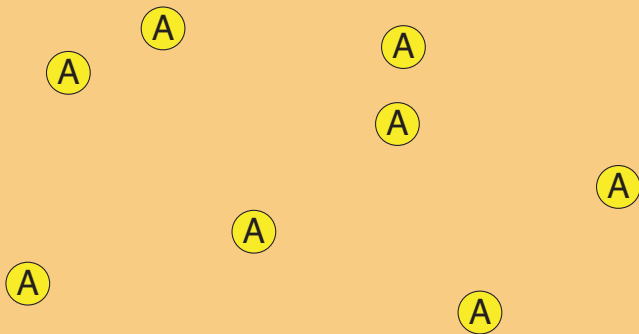
Agents



La virtualisation communautaire

Ressources distribuées

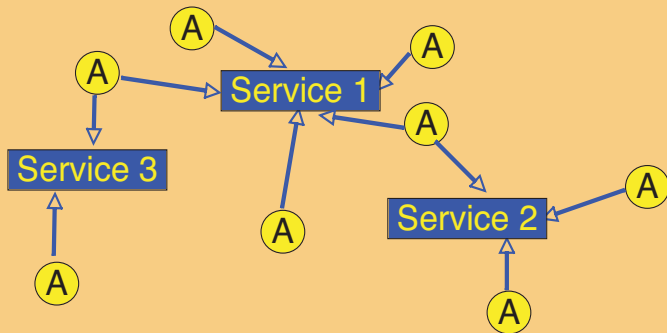
Agents



Les services

Ressources distribuées

Agents



Ressources distribuées

Agents

Services

Service 1

Service 3

Service 2



Les prochaines générations de grilles

Les services actuels

- Allocations des ressources
- Surveillance des tâches
- Gestion des données
- ...

Les services du court et long terme :

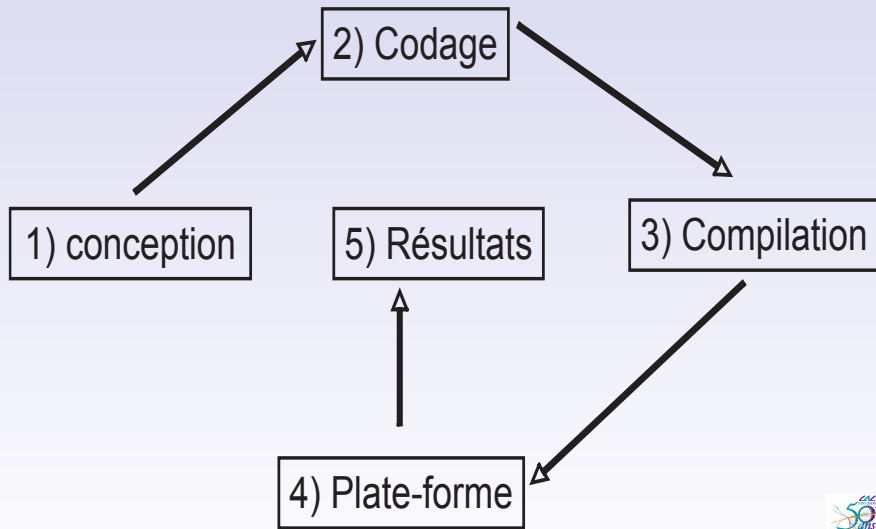
- Gestion des flots de travail
- Équité entre utilisateurs
- Visualisation des résultats
- Inter-opérabilité grilles
- Inter-activité
- Persistance des données

Quel software pour 2010-2015 en HEP ?

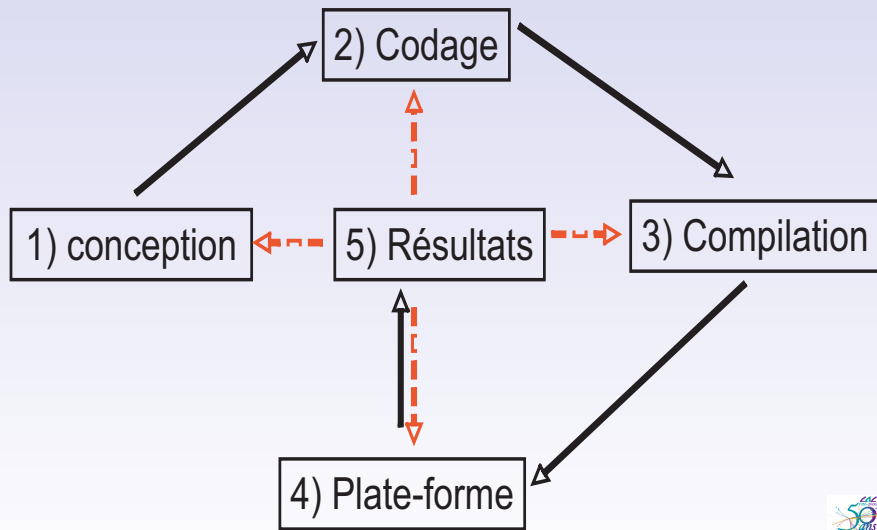
- ▶ Les applications identiques :
 - production, reconstruction et analyse de données
- ▶ Comment coupler/utiliser/supporter les codes applicatifs ?



Cycle de développement actuel



Adéquation dans 10 ans ?



Les nouveaux processus de développement I

Les prédictions

- ▶ Concepts de l'orienté objet acquis
- ▶ D'autres modèles de conception, e.g. approche composants
- ▶ Définition des structures de données et algorithmes
 - Question de la ré-utilisabilité de l'existant
- ▶ Peut-être moins de distinction entre la conception et l'implémentation ? ou pas ...
 - Avènement de la programmation concurrentielle/parrallèle
 - Développement de langage descriptif des modèles et inter-actions
 - Abstraction/Langage de haut niveau
 - Identification de modèle de conception (model pattern)



Les nouveaux processus de développement II

Plate-formes & exécution : Des frameworks virtuels distribués

- ▶ Gestion des flots de travail utilisateurs et processus distribués
- ▶ Développement virtuel naturellement multi-plateforme/OS
 - ▶ De plus grandes contraintes dû à un environnement distribué, parallèle et hétérogène
 - ▶ Importance de la gestion de configuration pour masquer cette hétérogénéité à l'utilisateur
- ▶ La distribution du code favorise une plus grande modularité
 - Quid de *Root* ? - monolithique par philosophie, *Boot* un premier pas vers plus de modularité ?
- ▶ Contraintes intrusives au niveau de l'application
 - Surveillance du déroulement de l'application
 - Traitement des erreurs applicatives



Les outils I

Évolution de l'existant

- ▶ Environnement de développement intégré (Visual, Xcode + Shark, Eclipse)
- ▶ Ajout de fonctionnalité pour le développement multi-threadé, parallèle et nouveaux modèles de programmation
- ▶ Parallélisation automatique, trop compliqué sinon
- ▶ Comment débbugger, profiler et optimiser les programmes dans un environnement non déterministe
 - Problématique de la recherche en parallélisme
 - Évolution gdb, valgrind, oprofile, ...
 - Nouveaux outils d'émulation, de tests
 - Système grandeur nature dédié (Grid5000 ?)
- ▶ Les systèmes de version : fin de vie de Cvs pour Svn, Darcs, ...



Les outils II

Traitements des résultats et informations

- ▶ Constats : Beaucoup de ressources, conséquences : beaucoup de résultats et d'informations
- ▶ Comment les analyser, détecter les erreurs ?
 - Problématique du « result mass checking »
- ▶ Développement d'outils d'aide à l'analyse des résultats et traitements de l'information
 - Trouver des informations pertinentes dans des masses de données considérables, hétérogènes et distribuées
 - Problématique d'apprentissage, de représentation des connaissances, d'intégration de données hétérogènes, de fouilles de données



Conclusions

- ▶ Les futurs environnements apporteront dès le départ des contraintes sur la conception/implémentation du soft
- ▶ Beaucoup de problèmes se résoudre de manière pragmatique et expérimentale
- ▶ Certains concernent la recherche en informatique
- ▶ Importance du rôle des experts
- ▶ Nos convictions pour un meilleur futur :
 - Plus de comités (transversaux) d'experts de standardisation dans plusieurs domaines (services grilles, persistances des données, visualisation, etc.)
 - Besoin de standards « open source », modulaires et inter-opérables



Questions ?

- ▶ Quel rôle pour le LAL ?
- ▶ Nos domaine d'expertise actuels :
 - Bases de données
 - Gestion de projet/Configuration
 - Processus de développement
 - Outils collaboratifs
 - Visualisation des résultats
 - Grille de calculs/Applicatifs
 - Supports services/ressources
 - Apprentissage
 - ...
- ▶ Quasiment l'ensemble des activités « logicielles » et « matérielles » d'une expérience de physique !!!

