

# Vers l'orchestration des grilles de PCs par les mécanismes de publication-souscription issus des technologies du WEB

Présenté par  
**Leila Abidi**  
sous la direction de  
**Christophe Cerin & Mohamed Jemni**

Université de Paris 13, LIPN UMR CNRS 7030, France

Université de Tunis, LaTICE ENSIT, Tunisie

Vichy, 3 juin 2014



# Outline

- 1 Contexte
- 2 Problématiques
- 3 Contributions
- 4 Travaux en cours & travaux futurs

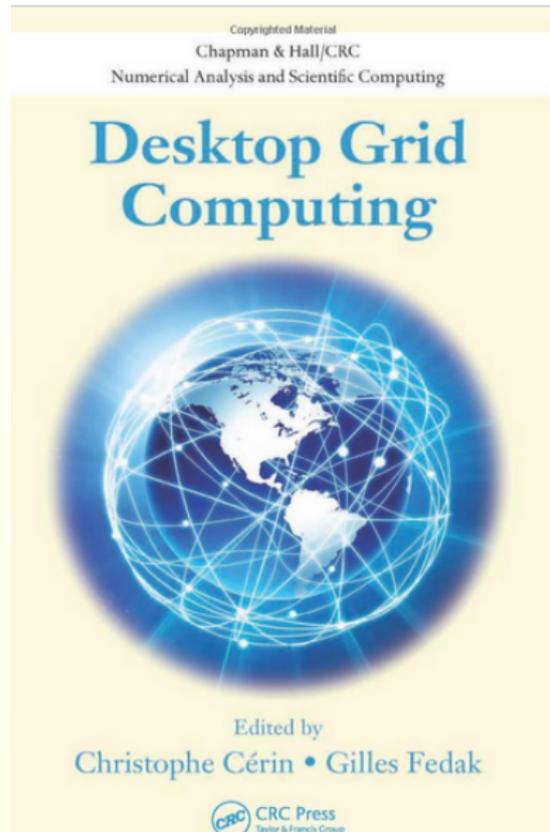


# Outline

- 1 Contexte
- 2 Problématiques
- 3 Contributions
- 4 Travaux en cours & travaux futurs



- Nous étudions les **Grilles de PC**, le calcul volontaire... pour aller vers les organisations de type Clouds ;
- Nous sommes partis d'une certaine situation (plus de 15 ans de travaux dans le domaine)
- Nous cherchons à repenser les interactions entre les composants traditionnels d'une grille de PCs :
  - en terme de **technologies** liées au **WEB**
  - en terme **formel** (à l'aide des outils de CPN) pour acquérir de la confiance dans les protocoles d'interaction



# Outline

- 1 Contexte
- 2 Problématiques**
- 3 Contributions
- 4 Travaux en cours & travaux futurs



# Faits et questions clés

- Les systèmes deviennent assez complexes qu'il faut les vérifier formellement : Besoin de **consolider** le protocole existant (BonjourGrid dans notre cas)
- Le paradigme de communication (pour la coordination, non pas pour l'échange de données) doit avoir un haut niveau d'asynchronisme afin d'assurer la **scalabilité**
- Les **technologies Web** sont l'avenir de la conception des systèmes distribués. Comment les technologies du Web et les technologies des grilles peuvent-elles fusionner ?
- Le nombre croissant d'appareils **mobile intelligents** pousse les applications de grille existantes à s'intégrer dans ces nouvelles plateformes à base de technologies du Web



# Faits et questions clés

- Les systèmes deviennent assez complexes qu'il faut les vérifier formellement : Besoin de **consolider** le protocole existant (BonjourGrid dans notre cas)
- Le paradigme de communication (pour la coordination, non pas pour l'échange de données) doit avoir un haut niveau d'asynchronisme afin d'assurer la **scalabilité**
- Les **technologies Web** sont l'avenir de la conception des systèmes distribués. Comment les technologies du Web et les technologies des grilles peuvent-elles fusionner ?
- Le nombre croissant d'appareils **mobile intelligents** pousse les applications de grille existantes à s'intégrer dans ces nouvelles plateformes à base de technologies du Web



# Faits et questions clés

- Les systèmes deviennent assez complexes qu'il faut les vérifier formellement : Besoin de **consolider** le protocole existant (BonjourGrid dans notre cas)
- Le paradigme de communication (pour la coordination, non pas pour l'échange de données) doit avoir un haut niveau d'asynchronisme afin d'assurer la **scalabilité**
- Les **technologies Web** sont l'avenir de la conception des systèmes distribués. Comment les technologies du Web et les technologies des grilles peuvent-elles fusionner ?
- Le nombre croissant d'appareils **mobile intelligents** pousse les applications de grille existantes à s'intégrer dans ces nouvelles plateformes à base de technologies du Web



# Faits et questions clés

- Les systèmes deviennent assez complexes qu'il faut les vérifier formellement : Besoin de **consolider** le protocole existant (BonjourGrid dans notre cas)
- Le paradigme de communication (pour la coordination, non pas pour l'échange de données) doit avoir un haut niveau d'asynchronisme afin d'assurer la **scalabilité**
- Les **technologies Web** sont l'avenir de la conception des systèmes distribués. Comment les technologies du Web et les technologies des grilles peuvent-elles fusionner ?
- Le nombre croissant d'appareils **mobile intelligents** pousse les applications de grille existantes à s'intégrer dans ces nouvelles plateformes à base de technologies du Web



# Champs disciplinaires et communautés

- Spécification formelle : les CPN sont un outil qui guident le développement et qui permettent de vérifier de "bonnes propriétés"
- La communauté qui développe des intergiciels de grilles (HPC... évoluant vers les organisations de type Clouds)
- Architecture des systèmes : la structure générale inhérente à un système informatique (grille de PCs), l'organisation des différents éléments et des relations entre eux.
- Service computing : discipline transversale qui couvre les sciences et les technologies de l'information afin de combler le fossé entre les services aux entreprises et les services informatiques plus classiques (serveur Web...).

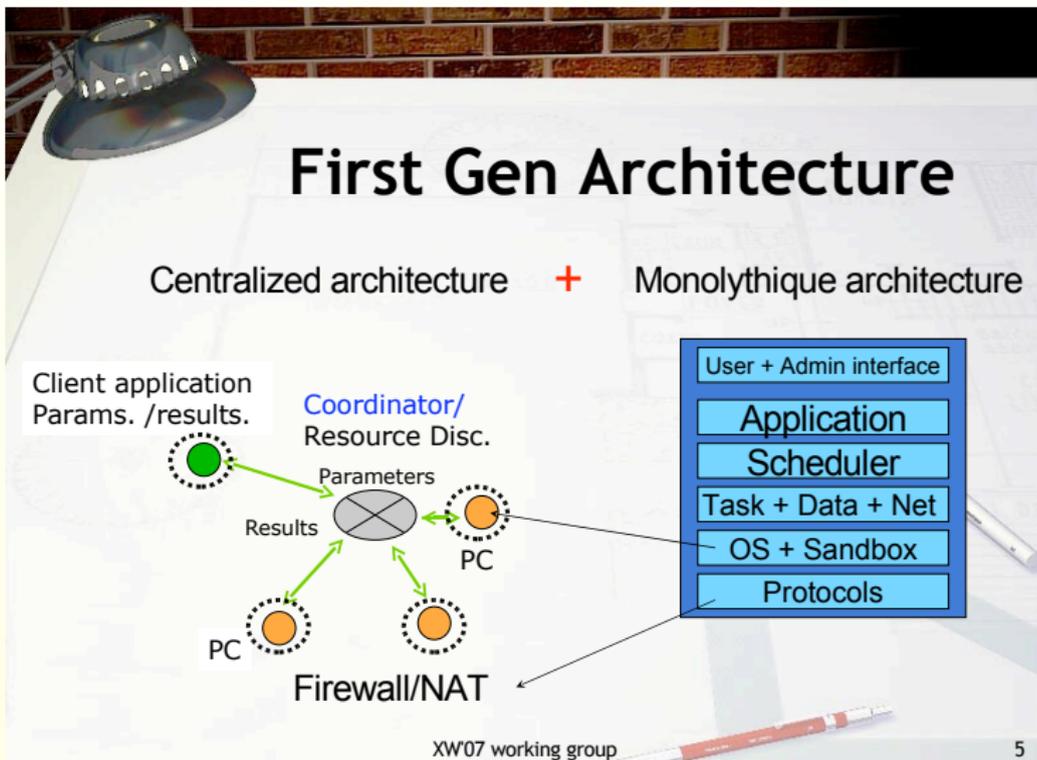


# Difficultés majeures

- Comment reconsidérer les intergiciels de grille de PC pour les cloudifier ?
- L'utilisateur du cloud est au centre des préoccupations : il doit pouvoir déployer une application sans avoir besoin d'un administrateur système
- Les interactions entre :
  - les composants "traditionnels" (master/worker, certification des résultats, monitoring...)
  - et "nouveaux" : la facturation (en n'oubliant pas que les technologies pour l'implémentation sont issues du Web - on ne fait plus du ad-hoc comme par le passé)



# Histoire des grilles de PCs : première génération



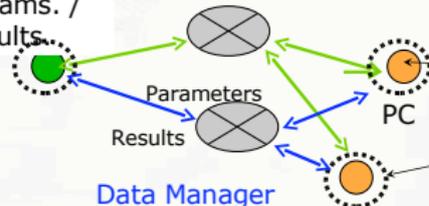
# Histoire des grilles de PCs : deuxième génération

## Second Gen Architecture

Centralized architecture (split tasks/data mgnt, Inter node coms) + Monolithic architecture

Client application  
Params. /  
results

Coordinator/  
Scheduler (Tasks)



Data Manager  
Scheduler (Tasks)

User + Admin interface

Application

Scheduler

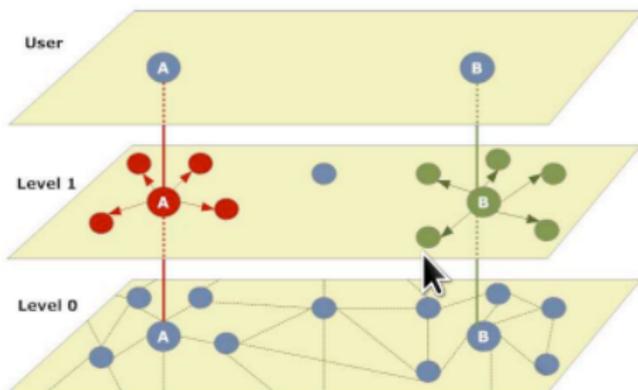
Task + Data + Net

OS + Sandbox

Protocols

Firewall/NAT

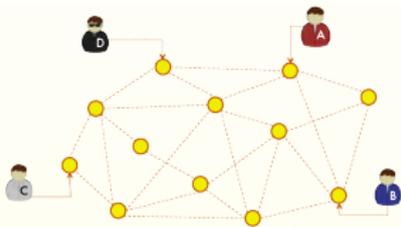
# Histoire des grilles de PCs : BonjourGrid



Déploiement de l'intergiciel de calcul		
XtremWeb	Boinc	Condor
Construction de réseau des éléments de calcul		
Découverte des ressources	Caractéristiques des ressources	
Etablissement de la connexion à BonjourGrid		
Protocole ZeroConf (Bonjour)		



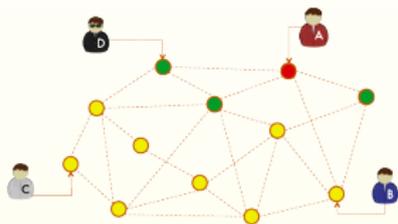
# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
- A la fin, le coordinateur retourne à l'état "idle"
- Les machines esclaves sont libérées et
- Le **protocole de coordination** :
  - gère et contrôle les ressources, les services et les éléments de calcul ;
  - ne dépend d'aucune machine en particulier ou d'un élément central.



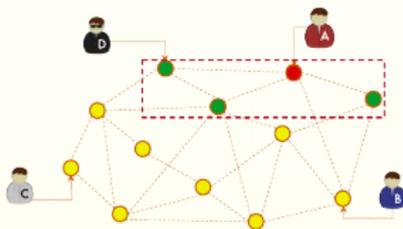
# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
- A la fin, le coordinateur retourne à l'état "idle"
- Les machines esclaves sont libérées et
- Le **protocole de coordination** :
  - gère et contrôle les ressources, les services et les éléments de calcul ;
  - ne dépend d'aucune machine en particulier ou d'un élément central.



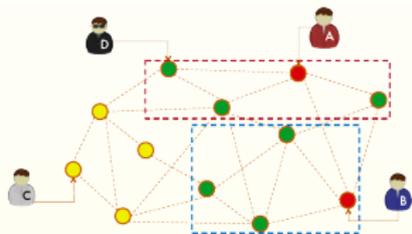
# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
- A la fin, le coordinateur retourne à l'état "idle"
- Les machines esclaves sont libérées et
- Le **protocole de coordination** :
  - gère et contrôle les ressources, les services et les éléments de calcul ;
  - ne dépend d'aucune machine en particulier ou d'un élément central.



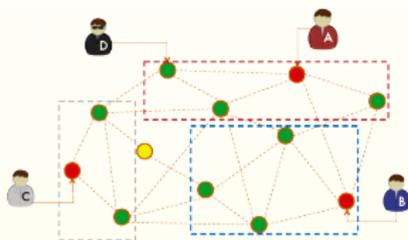
# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
  - A la fin, le coordinateur retourne à l'état "idle"
  - Les machines esclaves sont libérées et
  - Le **protocole de coordination** :
    - gère et contrôle les ressources, les services et les éléments de calcul ;
    - ne dépend d'aucune machine en particulier ou d'un élément central.



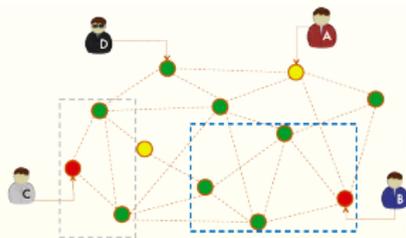
# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
  - A la fin, le coordinateur retourne à l'état "idle"
  - Les machines esclaves sont libérées et
  - Le **protocole de coordination** :
    - gère et contrôle les ressources, les services et les éléments de calcul ;
    - ne dépend d'aucune machine en particulier ou d'un élément central.



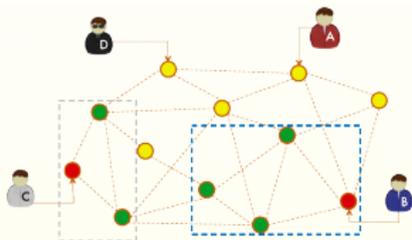
# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
- A la fin, le coordinateur retourne à l'état "idle"
- Les machines esclaves sont libérées et
- Le **protocole de coordination** :
  - gère et contrôle les ressources, les services et les éléments de calcul ;
  - ne dépend d'aucune machine en particulier ou d'un élément central.



# BonjourGrid : comment ça marche ?



- L'utilisateur lance une requête pour le calcul ;
- L'utilisateur fournit le graphe du contrôle du flux, les binaires et les données en entrée ;
- L'utilisateur déploie localement un coordinateur et les requêtes pour les participants ; **Nous assurons le support de XtremWeb, Condor et Boinc.**
- Le coordinateur sélectionne un ensemble de machines
- A la fin, le coordinateur retourne à l'état "idle"
- Les machines esclaves sont libérées et
- Le **protocole de coordination** :
  - gère et contrôle les ressources, les services et les éléments de calcul ;
  - ne dépend d'aucune machine en particulier ou d'un élément central.



# Outline

- 1 Contexte
- 2 Problématiques
- 3 Contributions**
- 4 Travaux en cours & travaux futurs



## Plan théorique et méthodologique

- Assurer une **confiance** dans le protocole grâce aux méthodes de vérification formelle
- Modéliser et vérifier un protocole pour les grilles de PCs basé sur le paradigme de **publication-souscription**
- Repenser les **interactions** entre les composants d'une grille de PCs par rapport aux technologies modernes du Web
- Assurer un degré de **sécurité** en implémentant des mécanismes pour la vérification/certification des résultats

## Développement logiciel

- Développer un **nouveau middleware** de grille pour les PCs, mais aussi pour smartphones et tablettes
- Intégration du nouveau middleware dans le cloud



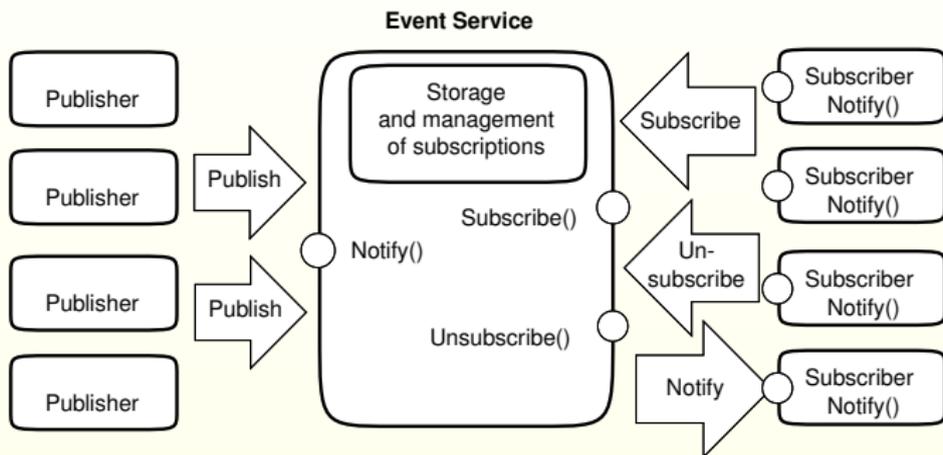
## Plan théorique et méthodologique

- Assurer une **confiance** dans le protocole grâce aux méthodes de vérification formelle
- Modéliser et vérifier un protocole pour les grilles de PCs basé sur le paradigme de **publication-souscription**
- Repenser les **interactions** entre les composants d'une grille de PCs par rapport aux technologies modernes du Web
- Assurer un degré de **sécurité** en implémentant des mécanismes pour la vérification/certification des résultats

## Développement logiciel

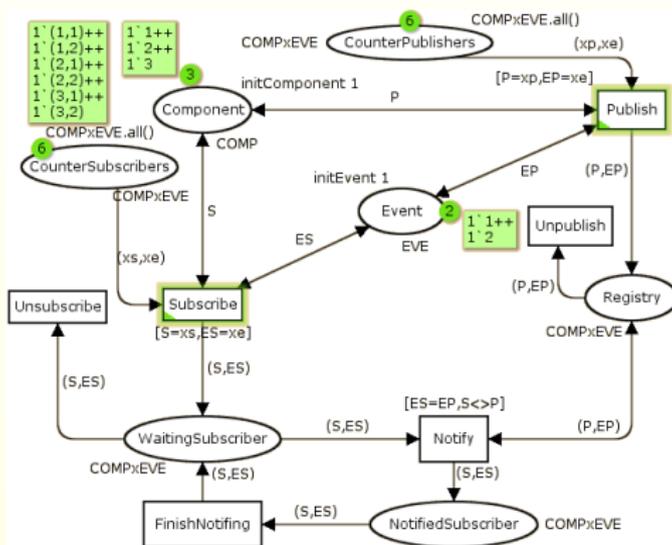
- Développer un **nouveau middleware** de grille pour les PCs, mais aussi pour smartphones et tablettes
- Intégration du nouveau middleware dans le cloud

# Publication-Souscription : Adéquat pour la scalabilité

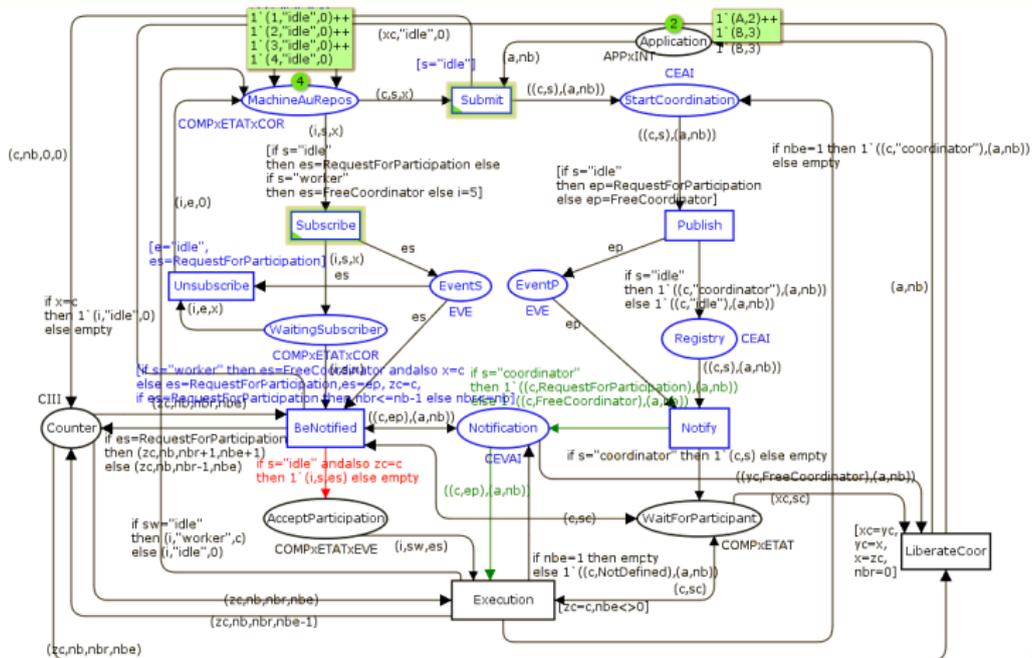


- Mode de communication : **multi-point**, **anonyme**, **implicite**, et **asynchrone**
- Découplage total entre les événements :
  - **Spatial** (les entités en interaction ne se connaissent pas)
  - **Temporel** (pas besoin d'interagir en même temps)

# Modélisation formelle du paradigme de publication-souscription



# Modélisation formelle de BonjourGrid



# Résultats

## Propriétés générales

- Pas de deadlocks : toutes les transitions sont exécutables (tout les événements peuvent se produire)
- L'espace d'états est construit à partir d'une seule composante fortement connexe (la propriété de vivacité est vérifiée)

## Propriétés spécifiques

- Chaque événement publié est reçu par tous les souscripteurs intéressés
- Un coordinateur commence l'exécution de son application s'il existe au moins une machine acceptant de participer
- Si un coordinateur finit l'exécution de son application alors tout les "Workers" associés doivent être libérés
- Un "Worker" ne peut être attaché qu'à un seul coordinateur



# Résultats

## Propriétés générales

- Pas de deadlocks : toutes les transitions sont exécutables (tout les événements peuvent se produire)
- L'espace d'états est construit à partir d'une seule composante fortement connexe (la propriété de vivacité est vérifiée)

## Propriétés spécifiques

- Chaque événement publié est reçu par tous les souscripteurs intéressés
- Un coordinateur commence l'exécution de son application s'il existe au moins une machine acceptant de participer
- Si un coordinateur finit l'exécution de son application alors tout les "Workers" associés doivent être libérés
- Un "Worker" ne peut être attaché qu'à un seul coordinateur





# Proposition d'un nouvel intergiciel (incluant les technologies du Web)

- Basé sur le paradigme de publication-souscription – Co-développement entre code et spécification
- Aspects requis :
  - Stocker le code d'exécution
  - Stocker les données d'entrée et de sortie
  - Offrir un mécanisme de publication-souscription
- État de l'art : Redis peut satisfaire tous les aspects requis



# Redis : fait partie de la mouvance NoSQL

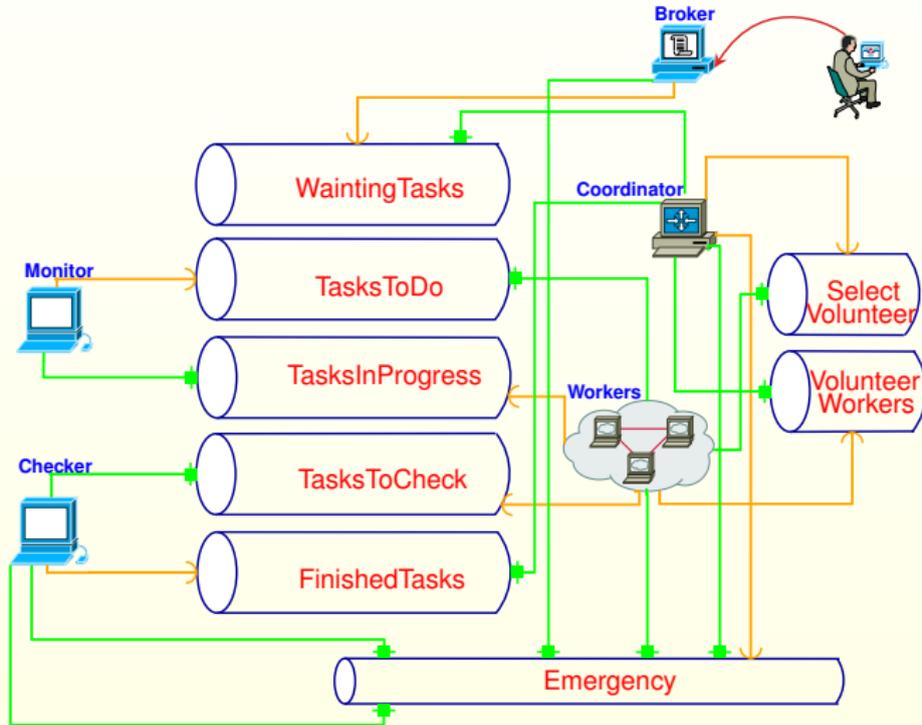
- Un système de gestion de base de données clef-valeur scalable,
- Propose différents types de données : chaînes de caractères, listes, ensembles ordonnés, hashes...
- Offre différentes options pour la persistance de données
- Offre de très hautes performances
- Facile à intégrer dans différentes architectures
- Supporte le paradigme de publication-souscription



redis



# RedisDG : schéma d'interaction



# Feedback

- Notre système est opérationnel : léger en termes de code
- Entièrement spécifié en termes de notification d'événements : **non conventionnel** mais **pertinent** pour un middleware de grille de PCs
- Capable de gérer des graphes séries-parallèles
- Certification des résultats
- Le Monitoring vise à remonter de l'information sur l'état du système (vérifier si tous les workers attachés à une application sont vivants)



# Outline

- 1 Contexte
- 2 Problématiques
- 3 Contributions
- 4 Travaux en cours & travaux futurs**



# Travaux en cours

- RedisDG est en cours d'intégration dans le Cloud SlapOS de l'université
- Remontée des informations de monitoring
- Evaluation des performances sur Grid5000



# Travaux futurs : long terme

- Objectif (contribution au FUI Resilience) : pousser et montrer les limites de la technologie SlapOS en cherchant à intégrer des intergiciels qui ne sont pas architecturés comme les logiciels du fond de commerce de SlapOS (Blogs, serveurs Apache, suite bureautique en ligne...)
- Objectif moins lié à Resilience : montrer que l'architecture SlapOS est ou n'est pas très conforme à ce que propose les standards (NIST...) pour architecturer un Cloud.



## Travaux futurs : moyen terme

- Dépasser les limites en termes de scalabilité pour l'implémentation actuelle du sous-système de **monitoring**
- Mener des **expérimentations** sur des clusters à large échelle
- Rendre **accessible** les technologies des DG pour la communauté d'E-science à travers l'automatisation des déploiements (sans intervention d'un administrateur système)



# Publications

- **2013 : Grid and Pervasive Computing**  
Leila Abidi, Christophe Cérin, Mohamed Jemni : Desktop Grid Computing at the Age of the Web.
- **2013 : ACM Symposium On Applied Computing**  
Leila Abidi, Christophe Cérin, Jean-Christophe Dubacq, Mohamed Jemni : A Publication-Subscription Interaction Schema for Desktop Grid Computing.
- **2012 : Grid and Pervasive Computing**  
Leila Abidi, Christophe Cérin, Kais Klai : Design, Verification and Prototyping the Next Generation of Desktop Grid Middleware.
- **2011 : IEEE Services Computing Conference**  
Leila Abidi, Christophe Cérin, Sami Evangelista : A Petri-Net Model for the Publish-Subscribe Paradigm and Its Application for the Verification of the BonjourGrid Middleware.

