The background is a dark blue field with several large, semi-transparent gears of various sizes scattered across it. On the left side, there is a vertical strip with a colorful, abstract, and somewhat pixelated texture in shades of orange, yellow, and brown.

RPC-V: Toward Fault-Tolerance for Internet Connected Desktop Grids with Volatiles Nodes

IEEE, 2004

Samir Djilali, Thomas Hérault, Oleg Lodygensky, Tanguy Morlier, Gilles Fedak et Franck Capello

INRIA, LRI, Université de Paris, Orsay



Plan de la présentation

Introduction

1. But de l'article

2. Pré-critique de l'article

3. RPC-V

3.1 Protocole

3.2 Implémentation

4. Expérimentation et Performance

4.1 Simulation

4.2 Test au travers de l'Internet

Conclusion

Futurs travaux de recherche

Discussion

Introduction

Etude d'un mécanisme de **Fautes tolérance** sur une base **RPC** (Architecture 3 tiers) **dans un environnement de grille** (Système distribué) formé de nœuds Desktop volatiles connectés à internet.

Article qui se veut d'offrir une perspective de FT RPC dans les grilles.

Idée de jamais-vu auparavant

1. But de l'article

Présenter un nouveau protocole RPC dans un environnement grille intégrant un mécanisme de Fautes Tolérance

*Insistons bien sur **l'aspect nouveauté de la démarche** des auteurs pour RPC dans les grilles (Internet Connected Desktop Grid). Toutes les études et démarches précédentes étaient Objets-Oriented (Middleware Corba....)*

2. Pré-critique de l'article

Exposition trop prononcée des problèmes relatifs aux middlewares dans les grilles en général, au challenge du passage à l'échelle, et aux types de services.

Avis personnel:

- Trop de détails inutiles par rapport au sujet
- + Offre un bon complément au chapitre suivant «Related Work»

3. RPC-V

RPC-V: Un protocole FT de RPC
environnement transparent et automatique.

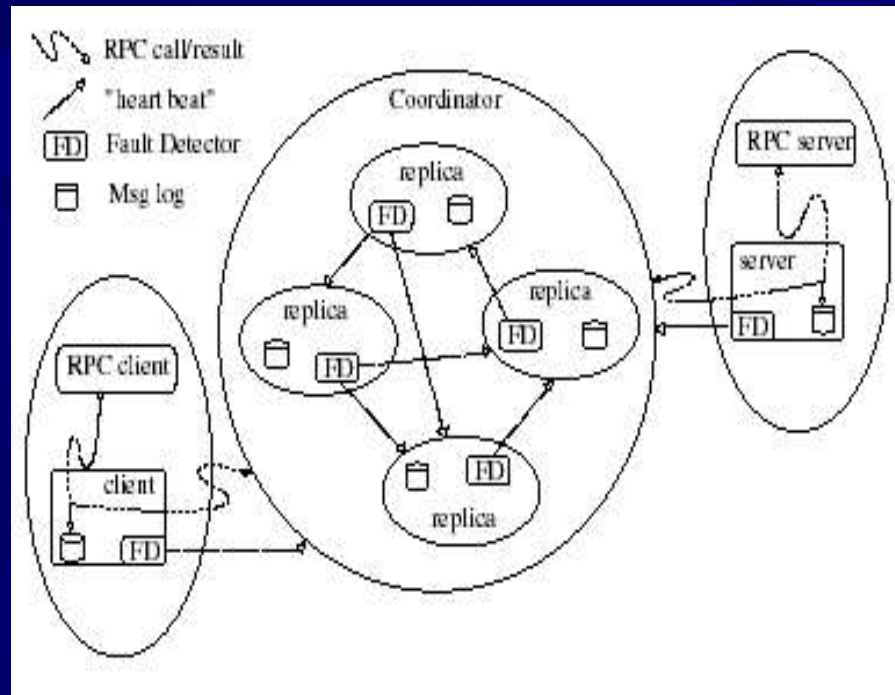
RPC-V en 4 points:

- Architecture 3 tiers: **Client-Coordinateurs-Serveur**.
- **Message logging** dans un but de **recovery**
- **Détection de fautes** sur tous les composants
- **Réplication passive** des Coordinateurs

3.1 RPC-V: Protocole

Points fondamentaux:

- Détection des fautes: suspicion de fautes (Msgs Heart beat)
- Recovery et duplication: principe de check point des RPC Msgs dans les coordinateurs puis duplication passive dans les coordinateurs



3.2 implémentations

API: API de **GridRPC** sans la fonction Remote Handle Management.

Middleware: Intégration de RPC-V dans **XtremWeb** pour des raisons de stabilité et sécurité

Détecteur de fautes: Message périodiques « **Heartbeat** »

Communication: **Connection-less**

Synchronisation: **Time Stamp** sur requête RPC

Checkpointing: Management des **Msgs Logging**

Communication P2P(Principe de **réplication**) entre Coordinateurs.

4. Expérimentation et Performance

But: Evaluer les performances du système « tolérance aux fautes dans RPC appliqué aux grilles (Internet Connected Desktop Grids) ».

Expérimentation en 2 phases:

- Simulation (conditions laboratoire)
- Test au travers de l'Internet

4.1 Simulation

Evaluation des 3 principes de base:

- Msgs Logging
- Réplication des coordinateurs
- Synchronisation

Visualisation de la fréquence des fautes (dégradation de la performance coté serveur et coté coordinateur)

But: Vérifier le fonctionnement correct et l'exactitude du protocole RPC-V de manière théorique

Problème: Les graphiques ne sont pas très représentatifs de la Tolérance aux fautes!

Test au travers de Internet

Coopération 3 universités, Wisconsin (Serveurs), Paris sud et Lille (Serveurs et Coordinateurs).

Test précédents réalisés en condition réelle

Test sur la réplication et le «Switching» des coordinateurs, montrent bien le comportement souhaités en cas de panne de coordinateur.

Bon points: Les graphes supplémentaires montrent mieux la tolérance aux fautes du système

Conclusion

Le Protocole RPC-V est efficace car en conditions réelles: mise en évidence la tolérance aux pannes.

L'étude de RPC-V est un succès

Les tests ont mis en évidence le peu d'impact de la volatilité des noeuds pour RPC-V.

Futurs travaux de recherche

- Evaluer les capacités du protocoles RPC-V à une plus large échelle.
- Etudier l'impact des taches «Check pointées» du serveur sur la performance et la FT.
- Possibilité d'implémenter une FT sur un RPC classique ou des services de grilles?

Discussion

- Article assez complet, mais un peu trop détaillé. Ceci fait que l'article sort un peu de son sujet de temps en temps.
- + Bonne démonstration du protocole RPC-V et de son efficacité en terme de tolérance aux fautes.