

Tolérance aux fautes dans les SMAs

Guillaume Vauvert

Projet DARX

Objectif

- SMA large échelle
- SMA réparti
- Criticité dynamique
- Choisir la meilleure réplication

Situation actuelle

Obs : [Calcul de la criticité]



(criticité)



Sys : [Calcul du degré de réplication]

- Réduction de l'information
- Interprétation de l'information
- Décision de la réplication (qui & comment)

Qu'est ce que la criticité ?

- Niveau de criticité d'un agent
- = importance pour l'application
- = gravité des conséquences d'un crash

- Il faut donc mesurer/estimer :
 - les conséquences d'un crash d'un agent
 - la probabilité que ce crash se produise.

- Répliquer = élever au carré la proba de crash

Qualité de Fonctionnement

- Trop répliquer => ↓ performance
 - charge processeur, réseau
- Globalement : maintenir Qualité de Fonctionnement
- Multidimensionnel, selon domaine :
 - simulation : vitesse, précision
 - résolution de problème : vitesse, qualité solution
 - fournisseur de services : quantité, qualité
 - ...

QoF non garantie

- Caractère aléatoire du comportement du système :
 - les pannes se produisent de façon aléatoire
 - le comportement n'est pas prédictible de façon exacte
- Connaissances incomplètes :
 - État du système
 - Calcul de la QoF

QoF non garantie

Solutions

- Maintenir une certaine QoF :
 $QoF > q$
- Avec une certaine probabilité :
 $P(QoF > q) > p$

Agir sur la QoF

$$P(QOF > q) > p$$

- Répliquer = \uparrow proba, \downarrow qof
- Déplacer agent :
 - \uparrow proba : site plus sûr
 - \uparrow qof : répartition de charge dans l'espace
- Ordonnancement de tâches
 - \uparrow qof : répartition de charge dans le temps
- Autres actions => autres dimensions qof

Agir sur la QoF

- réplication
- déplacement
- ordonnancement
- détection (heartbeat)
- estimation qof
- déplacement agent
- stratégie de réplication
- ... autres propres à l'application

QoF non satisfiable

Problème

$$P(QoF > q) > p$$

Que faire si ce critère est non satisfiable ?

- Conflit q vs p
 - Baisser QoF, augmenter proba = + répliquer
 - L'inverse = - répliquer
- QoF vs ressources
 - Quelle dimension de la qof sacrifier ?
- Problème : q_i , p ?

QoF non satisfiable

Solution

- Renégocier la qof avec un “agent_qof”
 - garantie, centralisation, info connue ?
- Chercher à maximiser $E(QOF)$
 - pas toujours satisfaisant
- Une fonction de satisfaction décrit les attentes de l'utilisateur. Pour $P(QOF > q) > p$:
sat : $QOF \times [0,1] \rightarrow [0,1]$
 $(q, p) \rightarrow sat(q,p)$
 - plus précis, mais plus difficile à choisir ?

Estimer $P(QoF > q)$

- Sources d'information
 - agent_système : charges, $P(\text{crash})$
 - agent_observateur : $\text{crash} \Rightarrow QoF$
 - utilisateur : $\text{sat}(q,p)$
 - concepteur du système \rightarrow agents
- Problèmes :
 - comprendre et intégrer les sources
 - dynamique, irrégulier, confiance



$\text{sat}(q,p)$
 $P(\text{QoF} > q) > p$


Système
charges
(processeur, réseau)
 $P(\text{crash}(A))$


Observateur
crash \Rightarrow QoF


Répliqueur

- estime $P(\text{QoF} > q) > p$
- adapte réplication (qui/comment) pour satisfaire utilisateur

Distribuer

l'estimation de $P(QoF > q)$

- L'estimation $P(QoF > q)$
 - Estimation locale
 - dépend des mesures de récupération
 - par propagation, les conséquences en cascade sont prises en compte
 - Conséquences au niveau régional / global
 - Comparaison des gravités / proba d'impact
- L'estimation de la probabilité de crash
 - Modèles de faute, statistiques

Distribuer

la décision de réplication

- Agents Observateurs
 - centrée besoins => trop de réplication
- Agents Systèmes
 - centrée ressources = > trop peu de réplication ?
- Agents Mobiles
 - MAJ croyances sur criticités et ressources
 - échange de croyances : contact, traces
 - Pb : relativité des points de vue
 - Pb : itinéraires

Exemple : Simulation de phénomène physique

- Diffusion : agent = noeud du maillage
 - $val(a) = f(\{val(b)/b \in V(a)\})$
- En cas de crash : nouvel agent = moyenne
- Hypothèse : un seul crash
- Crash de a : QOF = (PertePrécision, Durée)
- Csqs a : $PP(a) = val(a) - \sum_{b \in V(a)} val(b) / |V(b)|$
- Csqs b : $PP(b) = g(val(b), f, PP(a), \{val(b)/b \in V(a)\})$

Exemple : Simulation de phénomène physique

- Agt_Obs :
 - estime :
 - csqs crash pour l'agent
 - quantité de calculs, de messages nécessaires
 - transmet au répliqueur
- Agt_Sys :
 - estime :
 - probabilité de crash
 - ressources disponibles
 - transmet au répliqueur

Exemple : Simulation de phénomène physique

- Agt_Réplicateur :
 - estime $P(\text{QOF} > q)$:
 - $\text{QOF}(a) = P(\text{crash}) \times \text{QOF}(\text{crash}) + P(\text{norm}) \times \text{QOF}(\text{norm})$
 - intégrer
 - si non satisfaisant, cherche comment agir
 - diffusion informations (maj en continu) :
 - csqs QOF à des voisins
 - proba de crash
 - d'après ses informations partielles, adapte la réplication

Il reste à ...

- Préciser les algorithmes de diffusion de l'information et de prise de décision distribuée (cas général)
- Définir ce qui est commun et ce qui spécifique à chaque application
- Implémenter des exemples jouets : simulation physique, monde des blocs
- Implémenter de vrais exemples !

FIN

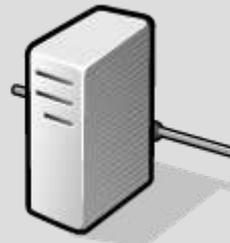


Estimateur
 $P(QOF > q)$



Systeme

Observateur



Réplicateur