

Théorie des applications réparties

Généralités sur les systèmes répartis

Plan du cours

- 1 Quelques exemples
- 2 Définitions
- 3 Pourquoi des systèmes répartis ?
- 4 Communications et horloges
- 5 Horloges logiques
- 6 Quelques problèmes importants d'algorithmique répartie

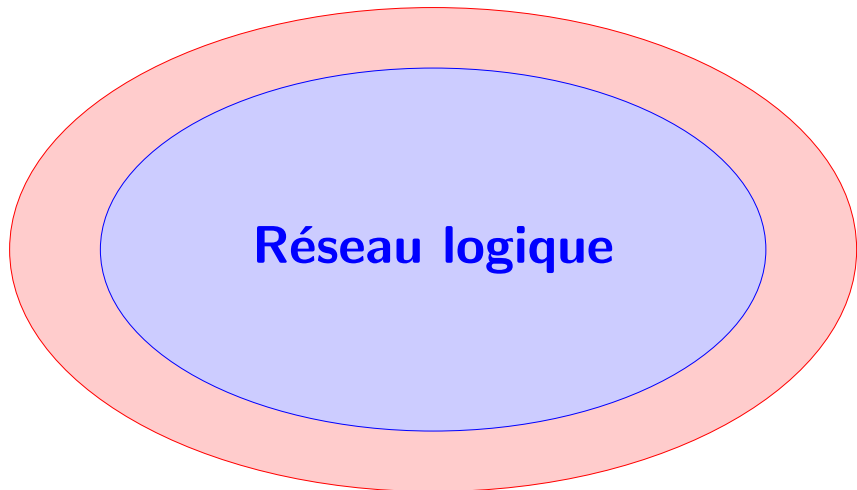
Internet et le Web

Un système réparti dans un système réparti dans un système réparti

Réseau physique

Internet et le Web

Un système réparti dans un système réparti dans un système réparti



Internet et le Web

Un système réparti dans un système réparti dans un système réparti



Web

Les systèmes Unix, rappels de M1101 et M2101

Il y a bien longtemps. . .

Les systèmes Unix « étaient » multi-utilisateurs (d'après un témoin direct de cette époque révolue, jusqu'en 2000 environ)

Les systèmes Unix, rappels de M1101 et M2101

Il y a bien longtemps. . .

Les systèmes Unix « étaient » multi-utilisateurs (d'après un témoin direct de cette époque révolue, jusqu'en 2000 environ)

Processus (du point de vue de l'utilisateur)

P1 ————

P2 ————

P1 ————

P2 ————

P1 ————

P2 ————

Le processeur est chargé de simuler l'exécution en parallèle des processus (car on apprécie généralement de pouvoir travailler en écoutant de la musique)

Mais aussi...

- Un processeur multi-cœur
- Un super calculateur
- Un réseau de capteurs sans fil
- L'informatique ubiquitaire
- Une voiture, un avion
- Le cloud

Et même...

- Une équipe de foot
- Un MMORPG
- Skynet



Plan du cours

- 1 Quelques exemples
- 2 Définitions**
- 3 Pourquoi des systèmes répartis ?
- 4 Communications et horloges
- 5 Horloges logiques
- 6 Quelques problèmes importants d'algorithmique répartie

Des définitions fameuses

« A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable. »

L. Lamport, dans un mail (1987)

Des définitions fameuses

« A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable. »

L. Lamport, dans un mail (1987)

« A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. » *Distributed Systems – Concepts and Design (90's)*

Des définitions fameuses

« A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable. »

L. Lamport, dans un mail (1987)

« A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. » *Distributed Systems – Concepts and Design (90's)*

« A distributed system is a collection of independent computers that appears to its users as a single coherent system. » *Distributed Systems – Principles and Paradigms (2000's)*

La définition que nous utiliserons

On appelle système réparti un ensemble de composants associé à un protocole de communication entre ces composants

La définition que nous utiliserons

On appelle système réparti un ensemble de composants associé à un protocole de communication entre ces composants

Composants

- Ordinateurs
- Processus
- Joueurs
- ...

Communications

- Passage de messages
- Mémoire partagée
- Rendez-vous
- ...

Remarque : réparti vs parallèle

Parallèle

Plusieurs composants travaillant ensemble, chacun ayant une vision globale du système, mémoire partagée

Réparti

Plusieurs composants travaillant ensemble, chacun ayant une vision locale du système, passage de messages

Remarque : réparti vs parallèle

Parallèle

Plusieurs composants travaillant ensemble, chacun ayant une vision globale du système, mémoire partagée

Réparti

Plusieurs composants travaillant ensemble, chacun ayant une vision locale du système, passage de messages

Pas vraiment de consensus

Dans ce cours on se contente de distinguer centralisé et réparti

Caractéristiques souhaitables en pratique

Varié beaucoup selon les systèmes, mais certaines se retrouvent souvent

Caractéristiques souhaitables en pratique

Variant beaucoup selon les systèmes, mais certaines se retrouvent souvent

Transparence(s)

L'utilisateur ne doit/souhaite pas savoir que le système est réparti, on cherche à masquer certains aspects :

- Localisation des composants
- Accès au système
- Hétérogénéité entre les composants
- Pannes locales

Caractéristiques souhaitables en pratique

Variant beaucoup selon les systèmes, mais certaines se retrouvent souvent

Transparence(s)

L'utilisateur ne doit/souhaite pas savoir que le système est réparti, on cherche à masquer certains aspects :

- Localisation des composants
- Accès au système
- Hétérogénéité entre les composants
- Pannes locales

Passage à l'échelle et/ou dynamisme

Ajout (ou retrait) de composants dans le système

Caractéristiques souhaitables en pratique (suite)

Tolérance aux pannes

- Différents types de pannes (voir cours 3)
- Redondance

Sécurité

- Authentification et authenticité
- Confidentialité
- Disponibilité des ressources, équité

Plus difficile qu'en centralisé ? cf *man in the middle*...

Plan du cours

- 1 Quelques exemples
- 2 Définitions
- 3 Pourquoi des systèmes répartis ?**
- 4 Communications et horloges
- 5 Horloges logiques
- 6 Quelques problèmes importants d'algorithmique répartie

Coût, efficacité

Exemple

MareNostrum (Barcelone) est constitué de près de 50000 processeurs « grand public », **bien moins coûteux** que de faire construire un (ou quelques) processeur spécifique qui permettrait d'atteindre **les mêmes capacités de calcul**

Coût, efficacité

Exemple

MareNostrum (Barcelone) est constitué de près de 50000 processeurs « grand public », **bien moins coûteux** que de faire construire un (ou quelques) processeur spécifique qui permettrait d'atteindre **les mêmes capacités de calcul**

Et même en ne tenant pas compte des coûts

On ne peut à priori pas augmenter infiniment la fréquence des processeurs (cf M1101) pour augmenter la vitesse de calcul, mais si on en fait travailler deux en parallèle on « double » cette vitesse

Flexibilité, adaptabilité

Extensions

Il est relativement simple d'ajouter ou de retirer des composants d'un système réparti bien conçu, c'est même une caractéristique attendue de nombre d'entre eux

- Réseaux (internet, pair-à-pair, ...)
- Cloud

Flexibilité, adaptabilité

Extensions

Il est relativement simple d'ajouter ou de retirer des composants d'un système réparti bien conçu, c'est même une caractéristique attendue de nombre d'entre eux

- Réseaux (internet, pair-à-pair, ...)
- Cloud

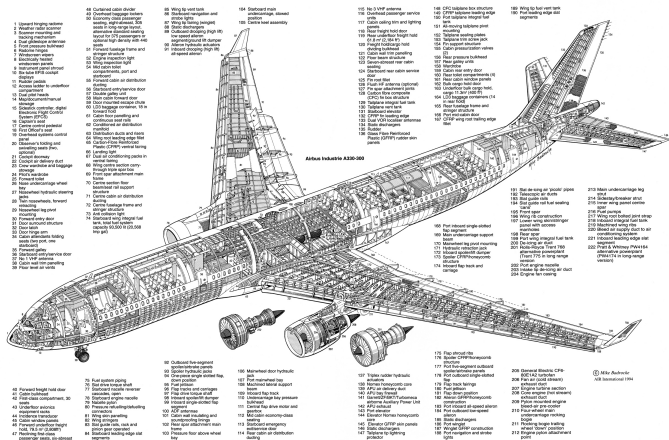
Mises à jour

Cette facilité d'extension peut permettre simplement de mettre à jour un système par morceaux, autorisant ainsi à garder ce système en fonctionnement durant sa mise à jour

- OSPF
- Graceful restart

Contrainte

Certains systèmes sont naturellement répartis en raisons de contraintes, par exemple, physiques



- 1 Upper leading section
- 2 Weather vane support
- 3 Reverse mounting and loading mechanism
- 4 Fuel distribution manifold
- 5 Fuel pressure indicator
- 6 Hydraulic line
- 7 Electrical system
- 8 Air conditioning
- 9 Insurance panel structure
- 10 Air line filter
- 11 Fuel filter
- 12 Access ladder
- 13 Door
- 14 Door lock
- 15 Door handle
- 16 Door latch
- 17 Door seal
- 18 Door hinge
- 19 Door hinge
- 20 Door hinge
- 21 Door hinge
- 22 Door hinge
- 23 Door hinge
- 24 Door hinge
- 25 Door hinge
- 26 Door hinge
- 27 Door hinge
- 28 Door hinge
- 29 Door hinge
- 30 Door hinge
- 31 Door hinge
- 32 Door hinge
- 33 Door hinge
- 34 Door hinge
- 35 Door hinge
- 36 Door hinge
- 37 Door hinge
- 38 Door hinge
- 39 Door hinge
- 40 Door hinge

- 41 Overhead cabinet door
- 42 Overhead storage locker
- 43 Overhead storage locker
- 44 Overhead storage locker
- 45 Overhead storage locker
- 46 Overhead storage locker
- 47 Overhead storage locker
- 48 Overhead storage locker
- 49 Overhead storage locker
- 50 Overhead storage locker
- 51 Overhead storage locker
- 52 Overhead storage locker
- 53 Overhead storage locker
- 54 Overhead storage locker
- 55 Overhead storage locker
- 56 Overhead storage locker
- 57 Overhead storage locker
- 58 Overhead storage locker
- 59 Overhead storage locker
- 60 Overhead storage locker
- 61 Overhead storage locker
- 62 Overhead storage locker
- 63 Overhead storage locker
- 64 Overhead storage locker
- 65 Overhead storage locker
- 66 Overhead storage locker
- 67 Overhead storage locker
- 68 Overhead storage locker
- 69 Overhead storage locker
- 70 Overhead storage locker
- 71 Overhead storage locker
- 72 Overhead storage locker
- 73 Overhead storage locker
- 74 Overhead storage locker
- 75 Overhead storage locker
- 76 Overhead storage locker
- 77 Overhead storage locker
- 78 Overhead storage locker
- 79 Overhead storage locker
- 80 Overhead storage locker

- 81 Wing to wing beam
- 82 Wing to wing beam
- 83 Wing to wing beam
- 84 Wing to wing beam
- 85 Wing to wing beam
- 86 Wing to wing beam
- 87 Wing to wing beam
- 88 Wing to wing beam
- 89 Wing to wing beam
- 90 Wing to wing beam
- 91 Wing to wing beam
- 92 Wing to wing beam
- 93 Wing to wing beam
- 94 Wing to wing beam
- 95 Wing to wing beam
- 96 Wing to wing beam
- 97 Wing to wing beam
- 98 Wing to wing beam
- 99 Wing to wing beam
- 100 Wing to wing beam
- 101 Wing to wing beam
- 102 Wing to wing beam
- 103 Wing to wing beam
- 104 Wing to wing beam
- 105 Wing to wing beam
- 106 Wing to wing beam
- 107 Wing to wing beam
- 108 Wing to wing beam
- 109 Wing to wing beam
- 110 Wing to wing beam
- 111 Wing to wing beam
- 112 Wing to wing beam
- 113 Wing to wing beam
- 114 Wing to wing beam
- 115 Wing to wing beam
- 116 Wing to wing beam
- 117 Wing to wing beam
- 118 Wing to wing beam
- 119 Wing to wing beam
- 120 Wing to wing beam

- 121 Outboard air frame
- 122 Outboard air frame
- 123 Outboard air frame
- 124 Outboard air frame
- 125 Outboard air frame
- 126 Outboard air frame
- 127 Outboard air frame
- 128 Outboard air frame
- 129 Outboard air frame
- 130 Outboard air frame
- 131 Outboard air frame
- 132 Outboard air frame
- 133 Outboard air frame
- 134 Outboard air frame
- 135 Outboard air frame
- 136 Outboard air frame
- 137 Outboard air frame
- 138 Outboard air frame
- 139 Outboard air frame
- 140 Outboard air frame
- 141 Outboard air frame
- 142 Outboard air frame
- 143 Outboard air frame
- 144 Outboard air frame
- 145 Outboard air frame
- 146 Outboard air frame
- 147 Outboard air frame
- 148 Outboard air frame
- 149 Outboard air frame
- 150 Outboard air frame
- 151 Outboard air frame
- 152 Outboard air frame
- 153 Outboard air frame
- 154 Outboard air frame
- 155 Outboard air frame
- 156 Outboard air frame
- 157 Outboard air frame
- 158 Outboard air frame
- 159 Outboard air frame
- 160 Outboard air frame

- 161 Outboard air frame
- 162 Outboard air frame
- 163 Outboard air frame
- 164 Outboard air frame
- 165 Outboard air frame
- 166 Outboard air frame
- 167 Outboard air frame
- 168 Outboard air frame
- 169 Outboard air frame
- 170 Outboard air frame
- 171 Outboard air frame
- 172 Outboard air frame
- 173 Outboard air frame
- 174 Outboard air frame
- 175 Outboard air frame
- 176 Outboard air frame
- 177 Outboard air frame
- 178 Outboard air frame
- 179 Outboard air frame
- 180 Outboard air frame
- 181 Outboard air frame
- 182 Outboard air frame
- 183 Outboard air frame
- 184 Outboard air frame
- 185 Outboard air frame
- 186 Outboard air frame
- 187 Outboard air frame
- 188 Outboard air frame
- 189 Outboard air frame
- 190 Outboard air frame
- 191 Outboard air frame
- 192 Outboard air frame
- 193 Outboard air frame
- 194 Outboard air frame
- 195 Outboard air frame
- 196 Outboard air frame
- 197 Outboard air frame
- 198 Outboard air frame
- 199 Outboard air frame
- 200 Outboard air frame

- 201 Outboard air frame
- 202 Outboard air frame
- 203 Outboard air frame
- 204 Outboard air frame
- 205 Outboard air frame
- 206 Outboard air frame
- 207 Outboard air frame
- 208 Outboard air frame
- 209 Outboard air frame
- 210 Outboard air frame
- 211 Outboard air frame
- 212 Outboard air frame
- 213 Outboard air frame
- 214 Outboard air frame
- 215 Outboard air frame
- 216 Outboard air frame
- 217 Outboard air frame
- 218 Outboard air frame
- 219 Outboard air frame
- 220 Outboard air frame
- 221 Outboard air frame
- 222 Outboard air frame
- 223 Outboard air frame
- 224 Outboard air frame

- 225 Outboard air frame
- 226 Outboard air frame
- 227 Outboard air frame
- 228 Outboard air frame
- 229 Outboard air frame
- 230 Outboard air frame
- 231 Outboard air frame
- 232 Outboard air frame
- 233 Outboard air frame
- 234 Outboard air frame
- 235 Outboard air frame
- 236 Outboard air frame
- 237 Outboard air frame
- 238 Outboard air frame
- 239 Outboard air frame
- 240 Outboard air frame
- 241 Outboard air frame
- 242 Outboard air frame
- 243 Outboard air frame
- 244 Outboard air frame
- 245 Outboard air frame
- 246 Outboard air frame
- 247 Outboard air frame
- 248 Outboard air frame
- 249 Outboard air frame
- 250 Outboard air frame
- 251 Outboard air frame
- 252 Outboard air frame
- 253 Outboard air frame
- 254 Outboard air frame
- 255 Outboard air frame
- 256 Outboard air frame
- 257 Outboard air frame
- 258 Outboard air frame
- 259 Outboard air frame
- 260 Outboard air frame

- 261 Outboard air frame
- 262 Outboard air frame
- 263 Outboard air frame
- 264 Outboard air frame
- 265 Outboard air frame
- 266 Outboard air frame
- 267 Outboard air frame
- 268 Outboard air frame
- 269 Outboard air frame
- 270 Outboard air frame
- 271 Outboard air frame
- 272 Outboard air frame
- 273 Outboard air frame
- 274 Outboard air frame
- 275 Outboard air frame
- 276 Outboard air frame
- 277 Outboard air frame
- 278 Outboard air frame
- 279 Outboard air frame
- 280 Outboard air frame
- 281 Outboard air frame
- 282 Outboard air frame
- 283 Outboard air frame
- 284 Outboard air frame
- 285 Outboard air frame
- 286 Outboard air frame
- 287 Outboard air frame
- 288 Outboard air frame
- 289 Outboard air frame
- 290 Outboard air frame
- 291 Outboard air frame
- 292 Outboard air frame
- 293 Outboard air frame
- 294 Outboard air frame
- 295 Outboard air frame
- 296 Outboard air frame
- 297 Outboard air frame
- 298 Outboard air frame
- 299 Outboard air frame
- 300 Outboard air frame

- 301 Outboard air frame
- 302 Outboard air frame
- 303 Outboard air frame
- 304 Outboard air frame
- 305 Outboard air frame
- 306 Outboard air frame
- 307 Outboard air frame
- 308 Outboard air frame
- 309 Outboard air frame
- 310 Outboard air frame
- 311 Outboard air frame
- 312 Outboard air frame
- 313 Outboard air frame
- 314 Outboard air frame
- 315 Outboard air frame
- 316 Outboard air frame
- 317 Outboard air frame
- 318 Outboard air frame
- 319 Outboard air frame
- 320 Outboard air frame
- 321 Outboard air frame
- 322 Outboard air frame
- 323 Outboard air frame
- 324 Outboard air frame
- 325 Outboard air frame
- 326 Outboard air frame
- 327 Outboard air frame
- 328 Outboard air frame
- 329 Outboard air frame
- 330 Outboard air frame
- 331 Outboard air frame
- 332 Outboard air frame
- 333 Outboard air frame
- 334 Outboard air frame
- 335 Outboard air frame
- 336 Outboard air frame
- 337 Outboard air frame
- 338 Outboard air frame
- 339 Outboard air frame
- 340 Outboard air frame

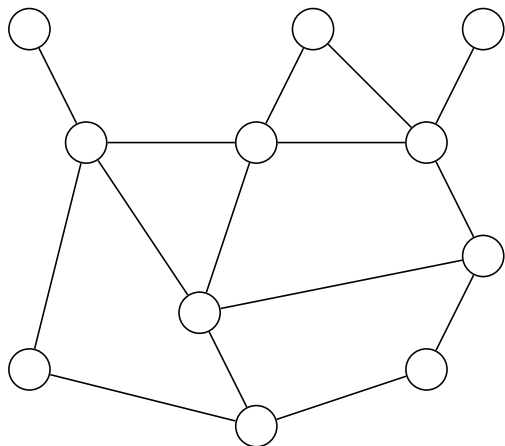
Plan du cours

- 1 Quelques exemples
- 2 Définitions
- 3 Pourquoi des systèmes répartis ?
- 4 Communications et horloges**
- 5 Horloges logiques
- 6 Quelques problèmes importants d'algorithmique répartie

Différents paradigmes de communication

Quelques exemples

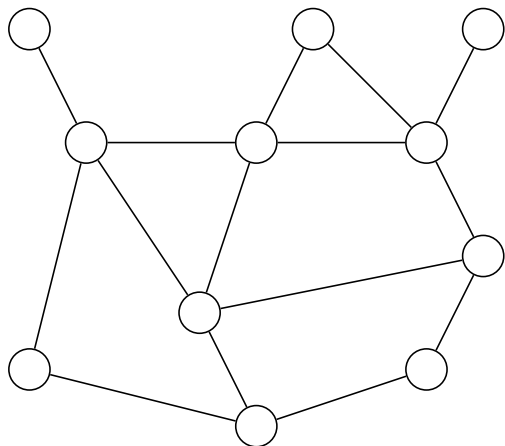
- Passage de messages



Différents paradigmes de communication

Quelques exemples

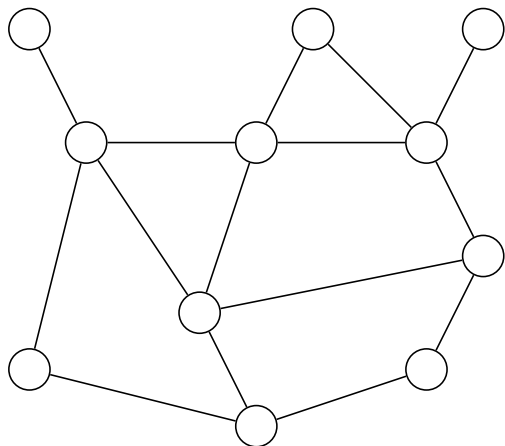
- Passage de messages
- Mémoire partagée



Différents paradigmes de communication

Quelques exemples

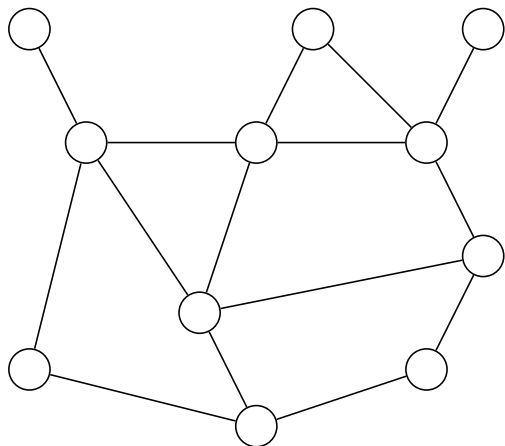
- Passage de messages
- Mémoire partagée
- Tableau noir



Différents paradigmes de communication

Quelques exemples

- Passage de messages
- Mémoire partagée
- Tableau noir
- Rendez-vous



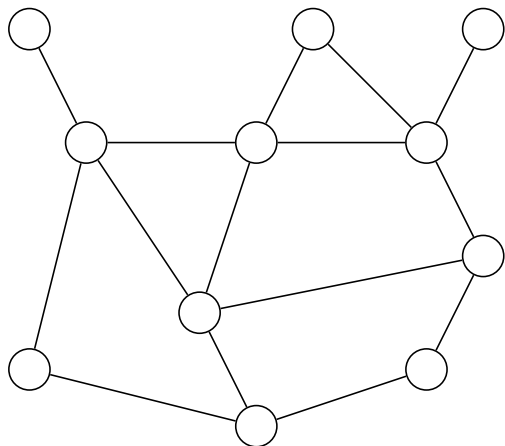
Différents paradigmes de communication

Quelques exemples

- Passage de messages
- Mémoire partagée
- Tableau noir
- Rendez-vous
- ...

Liens

Passage de messages simulé
par rendez-vous



Mémoire partagée

Primitives

- Lecture (*read*)
- Écriture (*write*)

Threads

TD juste après ce cours

Mémoire partagée

Primitives

- Lecture (*read*)
- Écriture (*write*)

Risques

Écritures concurrentes

Exemple (M3101), rediriger la sortie standard

Pas bien :

```
close(1);  
dup(somefd);
```

Bien :

```
dup2(somefd, 1);
```

Threads

TD juste après ce cours

Mémoire partagée

Primitives

- Lecture (*read*)
- Écriture (*write*)

Risques

Écritures concurrentes

⇒ il faut disposer d'opérations atomiques complexes, réserver les ressources (verrous)

Exemple (M3101), rediriger la sortie standard

Pas bien :

```
close(1);
dup(somefd);
```

Bien :

```
dup2(somefd, 1);
```

Threads

TD juste après ce cours

Passage de messages

Primitives

- Émission (*send*) multiple ou non
- Réception (*receive*)

Sockets

TD la semaine prochaine

Passage de messages

Primitives

- Émission (*send*) multiple ou non
- Réception (*receive*)

Risques

- Pertes
- Délais
- Ordres non-liés entre émissions et réceptions

Sockets

TD la semaine prochaine

Passage de messages

Primitives

- Émission (*send*) multiple ou non
- Réception (*receive*)

Risques

- Pertes
- Délais
- Ordres non-liés entre émissions et réceptions

Synchrone/Asynchrone

Le passage de messages est un paradigme de communication fortement asynchrone

Sockets

TD la semaine prochaine

Différence principale avec les systèmes centralisés

Centralisé

Une seule horloge

Réparti (vue d'un composant)

Horloges multiples et asynchrones

Différence principale avec les systèmes centralisés

Centralisé

Une seule horloge



Facilité à ordonner les évènements

Réparti (vue d'un composant)

Horloges multiples et asynchrones



Difficulté à ordonner les évènements

Différence principale avec les systèmes centralisés

Centralisé

Une seule horloge



Facilité à ordonner les évènements



Notion de causalité simple

Réparti (vue d'un composant)

Horloges multiples et asynchrones



Difficulté à ordonner les évènements



Notion de causalité moins naturelle

Différence principale avec les systèmes centralisés

Centralisé

Une seule horloge



Facilité à ordonner les évènements



Notion de causalité simple



Notion d'état global directe

Réparti (vue d'un composant)

Horloges multiples et asynchrones



Difficulté à ordonner les évènements



Notion de causalité moins naturelle



Pas d'état global évident

Plan du cours

- 1 Quelques exemples
- 2 Définitions
- 3 Pourquoi des systèmes répartis ?
- 4 Communications et horloges
- 5 Horloges logiques**
- 6 Quelques problèmes importants d'algorithmique répartie

La causalité, quand même, c'est important

Le problème de réception causale de messages

On considère plusieurs composants c_1, c_2, \dots, c_n qui communiquent par passage de messages, on considère aussi que tous les messages sont diffusés à tous les composants (*broadcast*)

Comment s'assurer que si un message m_1 est reçu par c_j avant l'émission par c_i d'un message m_2 , alors m_1 sera reçu par tout c_j avant m_2 ?

La causalité, quand même, c'est important

Le problème de réception causale de messages

On considère plusieurs composants c_1, c_2, \dots, c_n qui communiquent par passage de messages, on considère aussi que tous les messages sont diffusés à tous les composants (*broadcast*)

Comment s'assurer que si un message m_1 est reçu par c_j avant l'émission par c_i d'un message m_2 , alors m_1 sera reçu par tout c_j avant m_2 ?

Solution... plus tard

Mais en tout cas il faut savoir ordonner les événements survenant dans différents composants

Rappel de maths

Relation d'ordre

Une relation binaire R sur un ensemble E , telle que :

- R est réflexive : $\forall x \in E, xRx$,
- R est transitive : $\forall x, y, z \in E, (xRy \wedge yRz \Rightarrow xRz)$,
- R est antisymétrique : $\forall x, y \in E, (xRy \wedge yRx \Rightarrow x = y)$

Ordre total

R définit un ordre total sur E quand deux éléments de E sont toujours comparables pour R :

$$\forall x, y \in E, (xRy \vee yRx)$$

Ordre partiel

Si R ne définit pas un ordre total sur E alors R définit un ordre partiel

Notion d'horloge logique

Objectif

Construire un ordre causal (le moins partiel possible) sur les évènements (émissions et réceptions de messages) qui ont lieu au sein d'un système réparti

Notion d'horloge logique

Objectif

Construire un ordre causal (le moins partiel possible) sur les évènements (émissions et réceptions de messages) qui ont lieu au sein d'un système réparti

Un composant

Très simple, on a tout de suite un ordre total

Notion d'horloge logique

Objectif

Construire un ordre causal (le moins partiel possible) sur les événements (émissions et réceptions de messages) qui ont lieu au sein d'un système réparti

Un composant

Très simple, on a tout de suite un ordre total

Deux composants, trois, quatre. . .

Beaucoup moins évident

Notion d'horloge logique

Objectif

Construire un ordre causal (le moins partiel possible) sur les évènements (émissions et réceptions de messages) qui ont lieu au sein d'un système réparti

Un composant

Très simple, on a tout de suite un ordre total

Deux composants, trois, quatre. . .

Beaucoup moins évident

Question

Comment construire efficacement un ordre causal assez précis sur les évènements sans avoir une vision globale du système ?

Horloge de Lamport

Principe

- Chaque composant c_i a une horloge locale h_i
- À l'émission d'un message par c_i , la valeur de h_i est aussi émise

Mises à jour de h_i par c_i

- **Émission d'un message** m : $h_i = h_i + 1$; émettre (m, h_i)
- **Réception d'un message** (m, h) : $h_i = \max(h_i, h) + 1$

Horloge de Lamport

Principe

- Chaque composant c_i a une horloge locale h_i
- À l'émission d'un message par c_i , la valeur de h_i est aussi émise

Mises à jour de h_i par c_i

- **Émission d'un message m** : $h_i = h_i + 1$; émettre (m, h_i)
- **Réception d'un message (m, h)** : $h_i = \max(h_i, h) + 1$

Limitations

Ne capture pas complètement la causalité :

$$e_1 \text{ a lieu avant } e_2 \Rightarrow \text{date}(e_1) < \text{date}(e_2)$$

$$\text{date}(e_1) < \text{date}(e_2) \not\Rightarrow e_1 \text{ a lieu avant } e_2$$

Horloge de Lamport, remarque

Ordre total

On peut utiliser le principe de l'horloge de Lamport pour construire un ordre total sur les évènements d'un système réparti

Idée

Utiliser l'ordre lexicographique sur les couples (h_i, i) correspondant à chaque évènement

Horloge de Lamport, remarque

Ordre total

On peut utiliser le principe de l'horloge de Lamport pour construire un ordre total sur les évènements d'un système réparti

Idée

Utiliser l'ordre lexicographique sur les couples (h_i, i) correspondant à chaque évènement

Attention

Ce n'est pas nécessairement dans cet ordre que ce sont enchaînés les évènements, mais c'est un ordre plausible

Vecteurs d'horloges

Principe

- Chaque composant c_i a un vecteur d'horloge local V_i , sa vision du temps global
- Le i^{eme} élément de V_i représente l'horloge de c_i
- À l'émission d'un message par c_i , la valeur de V_i est aussi émise

Mises à jour de V_i par c_j

- **Émission d'un message m :**
 $V_i[i] = V_i[i] + 1$; émettre (m, V_i)
- **Réception d'un message (m, V) :**
 $V_i[i] = V_i[i] + 1$; $\forall j \neq i, V_i[j] = \max(V_i[j], V[j])$

Vecteurs d'horloges, propriétés

Propriété fondamentale

$\forall i, j, V_i[j]$ ne décroît pas, et $V_i[j] \leq V_j[j]$

Vecteurs d'horloges, propriétés

Propriété fondamentale

$$\forall i, j, V_i[j] \text{ ne décroît pas, et } V_i[j] \leq V_j[j]$$

Comparaison de vecteurs d'horloges

- $V_i \leq V_j$ si et seulement si $\forall k, V_i[k] \leq V_j[k]$
- $V_i < V_j$ si et seulement si $V_i \leq V_j$ et $V_i \neq V_j$
- Dans tous les autres cas les vecteurs ne sont pas comparables

Préservation de la causalité

$$e_1 \text{ a lieu avant } e_2 \Leftrightarrow \text{date}(e_1) < \text{date}(e_2)$$

$$e_1 \text{ concurrent avec } e_2 \Leftrightarrow \text{date}(e_1) \text{ et } \text{date}(e_2) \text{ ne sont pas comparables}$$

Retour à notre exemple

Le problème de réception causale de messages

Comment s'assurer que si un message m_1 est reçu par c_i avant l'émission par c_i d'un message m_2 , alors m_1 sera reçu par tout c_j avant m_2 ?

Retour à notre exemple

Le problème de réception causale de messages

Comment s'assurer que si un message m_1 est reçu par c_i avant l'émission par c_i d'un message m_2 , alors m_1 sera reçu par tout c_j avant m_2 ?

Vers une solution...

On peut utiliser les vecteurs d'horloges pour ordonner les messages

Retour à notre exemple

Le problème de réception causale de messages

Comment s'assurer que si un message m_1 est reçu par c_i avant l'émission par c_i d'un message m_2 , alors m_1 sera reçu par tout c_j avant m_2 ?

Vers une solution...

On peut utiliser les vecteurs d'horloges pour ordonner les messages

Solution (point de vue de c_i)

- **Émission de m :**
 $V_i[i] = V_i[i] + 1;$
 $\forall j \neq i$, émettre (m, V_i) pour c_j ;
- **Réception de (m, \mathbf{V}) :**
 attendre que $\forall j, V_i[j] + 1 \geq V[j];$
 prendre en compte m ;

Plan du cours

- 1 Quelques exemples
- 2 Définitions
- 3 Pourquoi des systèmes répartis ?
- 4 Communications et horloges
- 5 Horloges logiques
- 6 Quelques problèmes importants d'algorithmique répartie**

Exclusion mutuelle

Le problème

Plusieurs processus partagent une ressource, que chacun ne touche qu'à un endroit particulier de son code (appelé section critique), on veut s'assurer qu'il n'y ait jamais plus d'un processus à la fois qui soit dans sa section critique

Propriétés attendues

- **Sûreté** : à tout instant il y a au plus un processus en section critique
- **Vivacité** : si un processus veut entrer dans sa section critique, il le fera en temps fini (implique l'équité)

Exclusion mutuelle

Le problème

Plusieurs processus partagent une ressource, que chacun ne touche qu'à un endroit particulier de son code (appelé section critique), on veut s'assurer qu'il n'y ait jamais plus d'un processus à la fois qui soit dans sa section critique

Propriétés attendues

- **Sûreté** : à tout instant il y a au plus un processus en section critique
- **Vivacité** : si un processus veut entrer dans sa section critique, il le fera en temps fini (implique l'équité)

Solution(s)

Des idées ?

Consensus

Le problème

Plusieurs agents proposent chacun une valeur et doivent chacun décider une valeur, l'objectif est que tous les agents décident la même valeur

Propriétés attendues

- **Validité** : la valeur décidée par chaque agent qui décide est l'une des valeurs proposées
- **Accord** : tous les agents qui décident décident la même valeur
- **Terminaison** : tous les agents décident en temps fini

Consensus

Le problème

Plusieurs agents proposent chacun une valeur et doivent chacun décider une valeur, l'objectif est que tous les agents décident la même valeur

Propriétés attendues

- **Validité** : la valeur décidée par chaque agent qui décide est l'une des valeurs proposées
- **Accord** : tous les agents qui décident décident la même valeur
- **Terminaison** : tous les agents décident en temps fini

Solution(s)

Des idées ? Et si certains agents ont des pannes ?

Élection

Le problème

Plusieurs agents doivent élire un chef parmi eux

Propriétés attendues

- **Unicité** : un unique agent se considère comme le chef
- **Accord** : les autres agents qui ont décidé de leur rôle considèrent que l'agent précédent est le chef
- **Terminaison** : tous les agents décident de leur rôle en temps fini

Élection

Le problème

Plusieurs agents doivent élire un chef parmi eux

Propriétés attendues

- **Unicité** : un unique agent se considère comme le chef
- **Accord** : les autres agents qui ont décidé de leur rôle considèrent que l'agent précédent est le chef
- **Terminaison** : tous les agents décident de leur rôle en temps fini

Solution(s)

Des idées ? Liens avec le consensus ?

Snapshot

Le problème

On souhaite que, à la demande d'un composant, un aperçu (*snapshot*) de l'état global du système puisse être construit

Snapshot

Le problème

On souhaite que, à la demande d'un composant, un aperçu (*snapshot*) de l'état global du système puisse être construit

Propriétés attendues

- **Sûreté** : si un aperçu est calculé, il est cohérent
- **Vivacité** : si au moins un composant demande un aperçu, celui-ci est effectivement calculé

Aperçu cohérent

État de chacun des composants du système, tel que :

- il n'y a aucun message orphelin (reçu mais pas émis),
- les messages en transit (émis mais pas reçus) sont pris en compte