

Wooki : Un éditeur Wiki P2P

Maître de Conférence, Nancy-Université

LORIA - Universités de Nancy

29 Septembre 2006

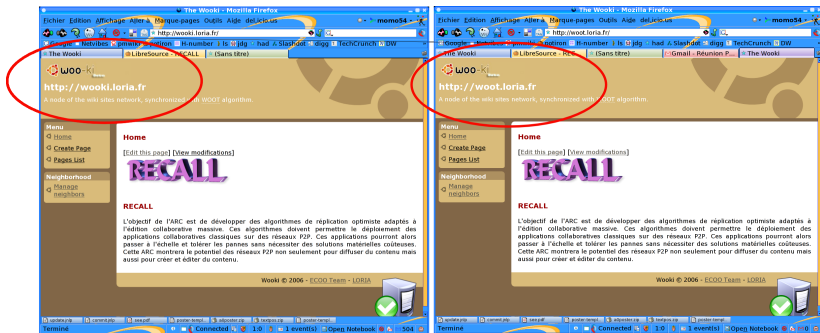
Outline

- 1 WOOT2
- 2 Unordered Pbcast
- 3 Anti-entropy
- 4 Purging and state transfer
- 5 Conclusions et Perspectives

Wooki

- Un éditeur Wiki :
 - adapté aux réseaux P2P
 - supportant le travail déconnecté

WOOT2
Unordered Pbcast
Anti-entropy
Purging and state transfer
Conclusion



Wooki

- WOOT2 +
- Unordered pbcast +
- Anti-entropy +
- interface Wiki

WOOT2

- Dédié aux structures linéaires...
- Assure la convergence qq's l'ordre de reception des opérations
 - sans site central,
 - sans consensus distribué,
 - sans ordre total.
 - sans vecteurs d'états

Algorithme de diffusion générique

Une opération est:

- 1 générée,
- 2 intégrée localement,
- 3 diffusée aux autres sites,
- 4 mise en attente tant que ses préconditions sont fausses,
- 5 ré-intégrée.

Modèle de donnée d'une page Wooki

- séquence de (idl,ligne,degré,visibilité)
- Le degré et l'id d'une ligne sont fixés à la création de cette ligne et sont invariant.
- La visibilité est un booléen
- Overhead en stockage = 1 id + 1 entier + 1 booléen par ligne

example

- riri
- fifi
- loulou

- ((0,1),riri,0,true)
- ((0,2),fifi,1,true)
- ((0,3),loulou,2,true)

Profil des Opérations du Wooki

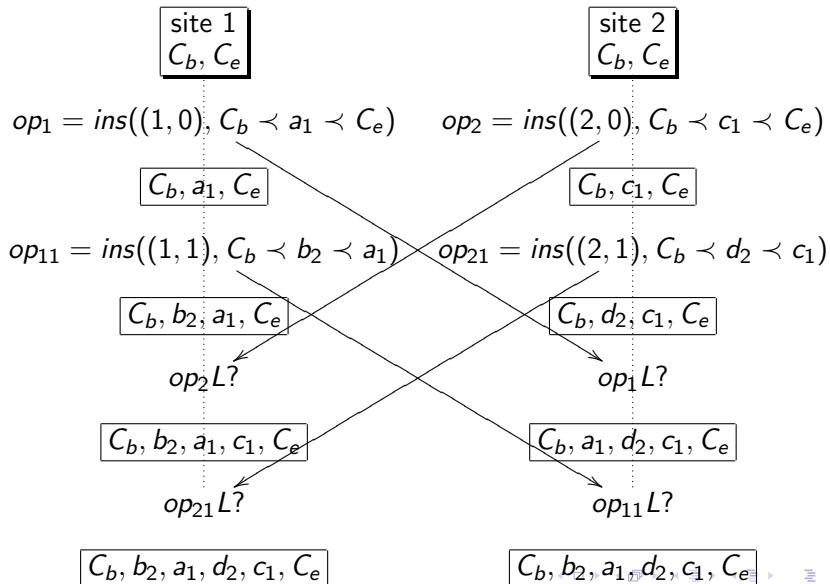
- 1 Insert(idl, C_P , C_N , degré, data):
 - idl=(siteid,(Lclock++)),
 - C_P =idl ligne précédente (C_B si début)
 - C_N =idl ligne suivante (C_E si fin)
 - $\text{degree}(c) = \max(\text{degree}(C_P), \text{degree}(C_N)) + 1$
- 2 del(idl):
 - Détruit la ligne identifiée par idl

Algorithme d'intégration de Woot

- $\text{Insert}(\text{idl}, C_P, C_N, \text{degré}, \text{data})$ n'est intégrée que si C_P, C_N existent localement...

$\text{IntegrateIns}(c, c_p, c_n)$

- **let** $S' := \text{subseq}(S, c_p, c_n)$
- **if** $S' := \emptyset$ **then**
 - $\text{insert}(S, c, i_n)$
- **else**
 - **let** $d_{\min} := \min(\text{degree}, S'), i := 0$
 - **let** $L := c_p d_0 d_1 \dots d_m c_n$ **where** $\forall i. \text{degree}(d_i) = d_{\min}$
 - **while** $(i < |L| - 1)$ **and** $(L[i] <_{id} c)$ **do** $i := i + 1$
 - $\text{IntegrateIns}(c, L[i - 1], L[i])$
- **endif**



WOOT/WOOT2

- Complexité en temps $O(n^2)$ au lieu de $O(n^3)$ (coût de $C_P(d_i) \leq_S c_p$)
- Gain en espace (plus de relation C_P et C_N)

Evaluation

- Nous avons démontré :
 - que notre algorithme est bien fondé
 - la terminaison
 - la convergence (*model-checking*) (spécification TLA)
- Reste à faire
 - Preuve de convergence
 - Annulation (undo/redo)
 - “Ramasse-miettes” réparti
 - d’autres structures de données (arborescente)

Unordered pbcast

- Chaque site a une vue partielle des autres sites (membership)
- Un message est envoyé à sous ensemble des sites connus avec un nombre de rebonds maximum (inondations)

Unordered Pbcast Protocol

```

(* State kept per pbcast: have I
 * received a message regarding
 * this pbcast yet? *)
let received\_already = false

(* Initiate a pbcast. *)
to pbcast(msg):
  deliver\_and\_gossip(msg,\nrounds)

(* Handle message receipt. *)
on receive Gossip(msg,round):
  deliver\_and\_gossip(msg,round)

(* Auxiliary function. *)
to deliver\_and\_gossip(msg,round):
  (* Do nothing if already received
   * it. *)
  if received\_already then return

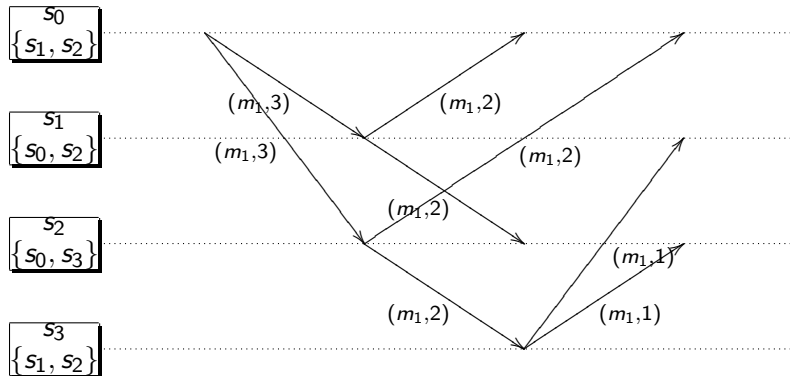
  (* Mark the message as being seen
   * and deliver. *)
  received\_already := true
  deliver(msg)

  (* If last round, don't gossip. *)
  if round = 0 then return

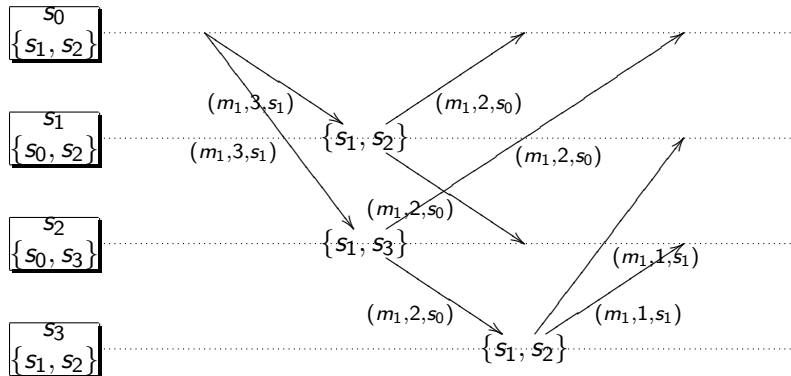
  foreach p in \processes:
    do with probability \rate:
      sendto p Gossip(msg,round-1)

```


Example : Inondation pbcast...



Example: Gestion des tables des sites



Anti-entropy

- Un message peut être perdu, un site peut être deconnecté
- Le site détectant une défaillance, prend un site dans sa table de site et démarre une anti-entropy avec lui.
- Il envoie son log à l'autre site, qui lui renvoie les messages manquant.
- Opération couteuse !

Bimodal multi-cast

Eichier Édition Affichage Aller à Aide

Précédente Suivante 12 sur 48 400%

52 • K. P. Birman et al.

Fig. 3. Illustration of the pbcast anti-entropy protocol.

Purging logs

- Dans Wooki : Age based purging
- Sinon Random purging ou Semantic Purging (neem)
- Pb site non réconciable
 - Transfert d'état...

Conclusion

- WOOKI permet la collaboration de masse
- D'après les premiers tests : 10000 ops concurrentes 10s