

Tapestry:

**Un réseau de recouvrement
résilient à échelle mondiale
pour le déploiement de service**



Sen Liu

25-01-2005

Le plan



I. Contexte

- i. Les auteurs
- ii. Qu'est que l'on a avant?

II. Système

- i. Objectif
- ii. Les algorithmes
 - a. DOLR
 - b. Le routage et la location d'objet
 - c. Le noeud dynamique
- iii. L'architecture

III. Expérimentations

IV. Conclusion

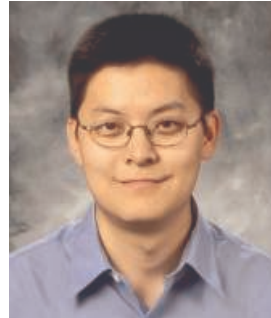
V. Bibliographie



Les auteurs



Auteurs



Ben Y. Zhao



Ling Huang



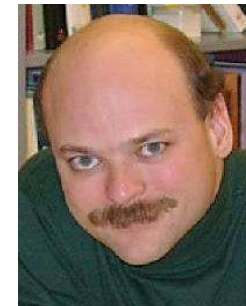
Jeremy Stribling



Sean C. Rhea



Anthony D. Joseph



John D. Kubiatowicz



Contexte

Système

Expérimentation

Conclusion

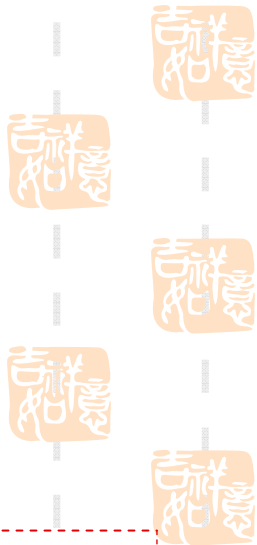
Bibliographie

Qu'est que l'on a avant?



Avant

- La première génération de système de paire à paire
 - Des applications de partage de fichier et de stockage
Napster, Gnutella, Freenet, etc.
- La deuxième génération de système de paire à paire
 - Des reseaux de recouvrement structurés
CAN, Chord, Pastry, Tapestry



Contexte

Système

Expérimentation

Conclusion

Bibliographie

Objectif



Objectif

Un réseau de recouvrement de paire à paire qui provide le routage de messages efficace, de location-indépendante et qui peut passer à grande échelle vers des répliquions d'un objet ou un service en utilisant seulement des ressources localisées.



Contexte

Système

Expérimentation

Conclusion

Bibliographie

DOLR



Algorithmes

DOLR

Routage
et location

Noeud
dynamique

- Le routage et la location d'objet décentralisés: DOLR
 - Router des messages aux pairs
 - Virtualiser des ressources
- Espace d'ID pour des pairs : valeurs de 160-bit avec une base globale définie
- Chaque noeud et chaque objet sont uniformément et par hasard assignés par un algorithme de hachage (SHA-1) du même espace d'ID, --N, --O_G
- ID d'application-spécifiée --A_{id}
- APIs de DOLR
 - PublishObject (O_G, A_{id}) UnpublishObject (O_G, A_{id})
 - RouteToObject (O_G, A_{id}) RouteToNode (N, A_{id}, Exact)

Contexte

Système

Expérimentation

Conclusion

Bibliographie

l'Etat de noeud



Algorithmes

DOLR

Routage
et location

Noeud
dynamique



- Chaque noeud stocke une table de voisins (la maille de routage)
 - Chaque niveau stocke des voisins qui matchent un préfixe à une certaine position dans l'ID, et il stocke un nombre d'entrées égale la base d'identifiant.
- La sauvegarde des liens de voisins sont stockés(Actuellement 2)
- Chaque noeud stocke aussi des backpointeurs vers des noeuds qui pointent vers lui.

Contexte

Système

Expérimentation

Conclusion

Bibliographie

Le routage



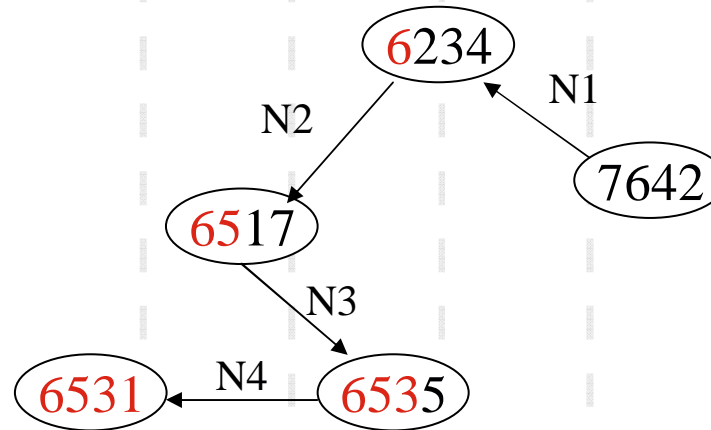
Algorithmes

DOLR

Routage
et location

Noeud
dynamique

- Utiliser le routage de préfixe (un exemple: 7642 -- 6531)
- S'il y a une entrée de voisin est nulle, alors utilise le routage substitutif.



Contexte

Système

Expérimentation

Conclusion

Bibliographie

La publication d'objet



Algorithmes

DOLR

Routage
et location

Noeud
dynamique

- Chaque ID est mappé à un noeud racine
- Le noeud racine d'un ID est le noeud où $\text{noeudID} = \text{ID}$ ou le plus proche noeud
- Un noeud envoie un message de publication vers le noeud racine de l'objet
- A chaque saut, noeuds stockent des pointeurs vers le noeud de source
 - Données restent dans la source
 - Si avec des répliquions, les pointeurs sont stockés en étant classifié en fonction de latence de réseau
- L'objet doit être périodiquement publié(Soft-State)

Contexte

Système

Expérimentation

Conclusion

Bibliographie

La location d'objet



Algorithmes

DOLR

Routage et location

Noeud dynamique

- Un client envoie un message vers le noeud racine de l'objet
- Chaque saut examine leur table de pointeurs
 - S'il y a un match, le message est avancé directement à la location d'objet
 - Sinon, le message est routé vers le noeud racine de l'objet
- Car des pointeurs sont classifiés, chaque recherche d'objet est dirigé à la réplique d'objet la plus proche

Contexte

Système

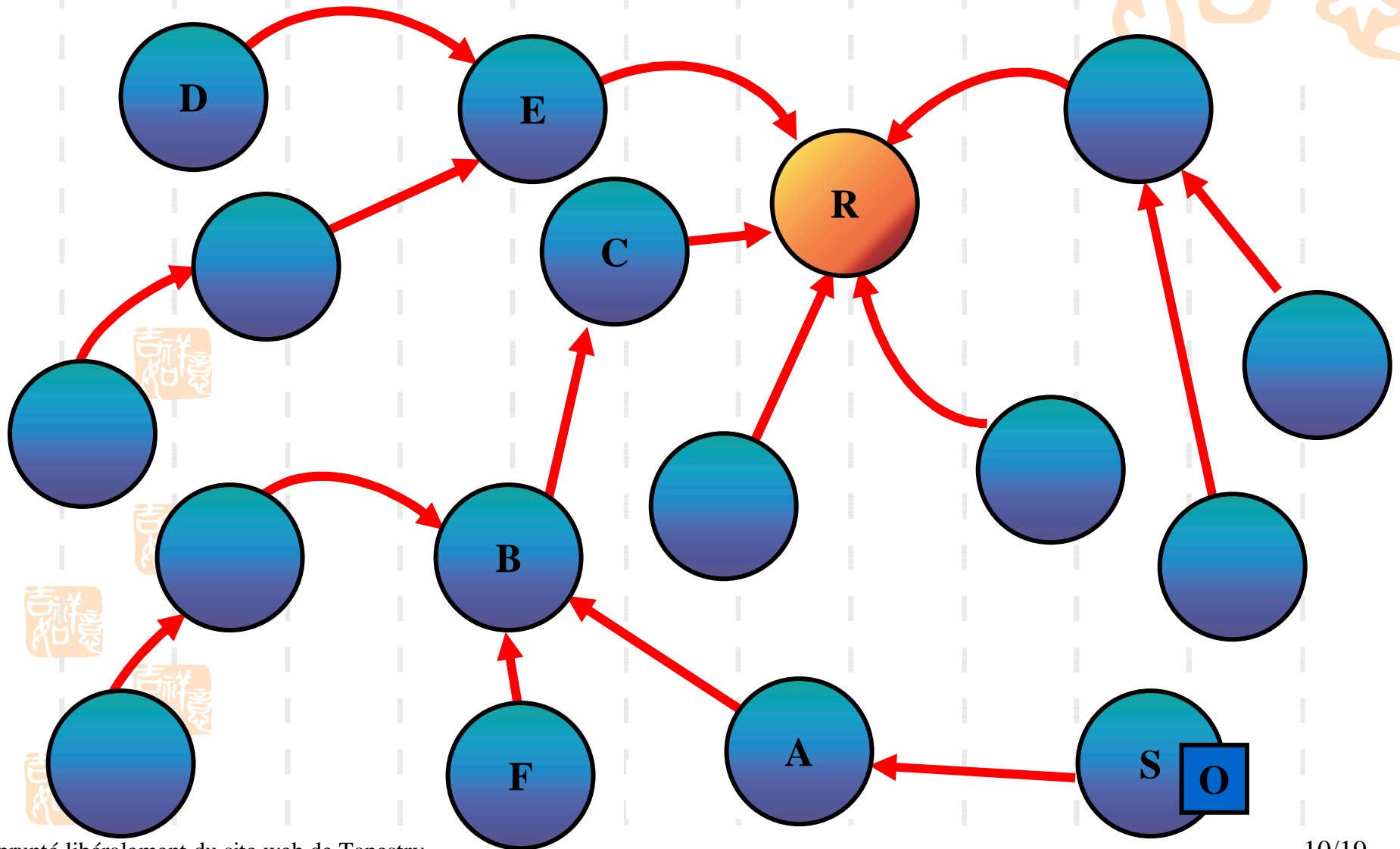
Expérimentation

Conclusion

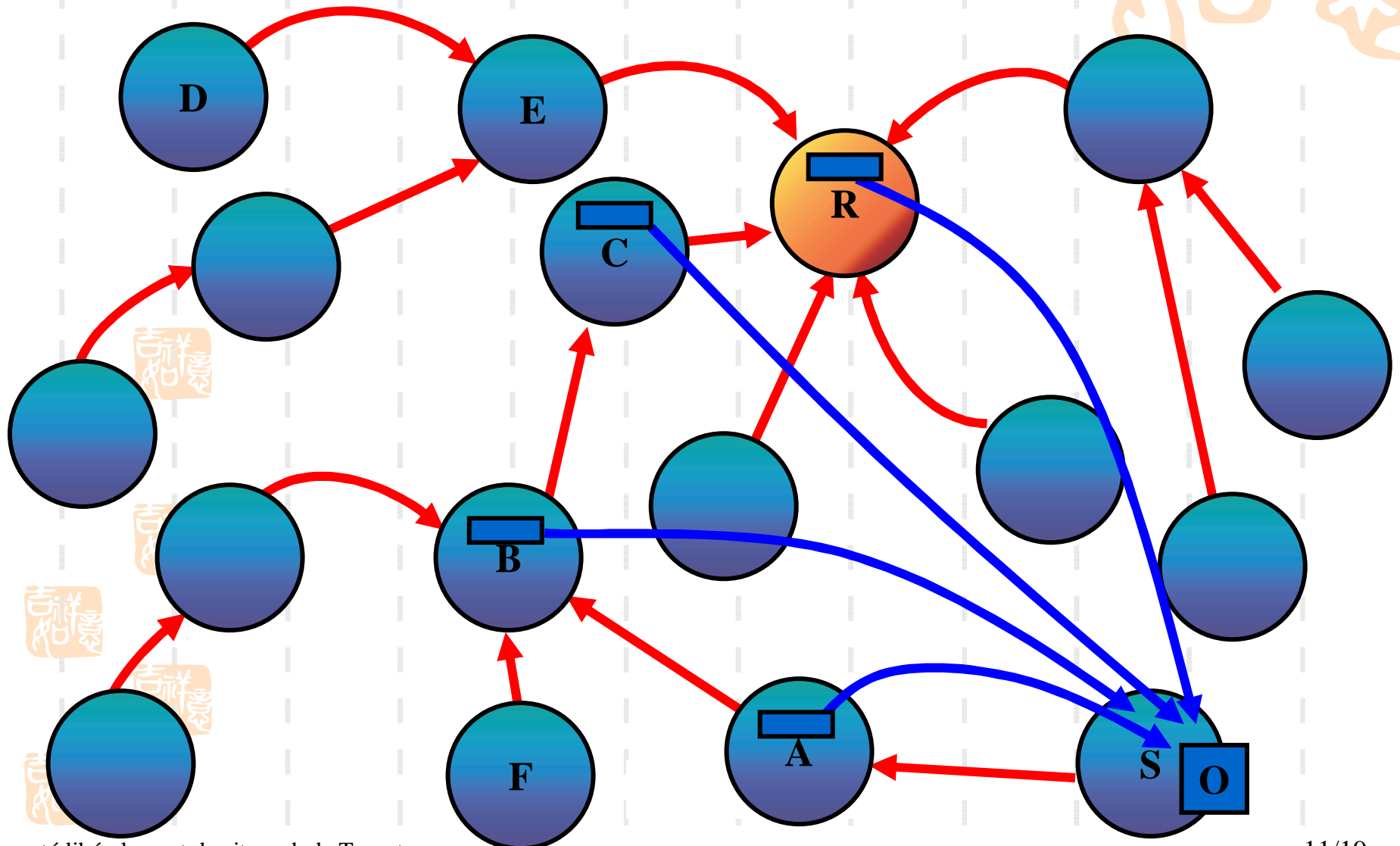
Bibliographie

La location d'objet

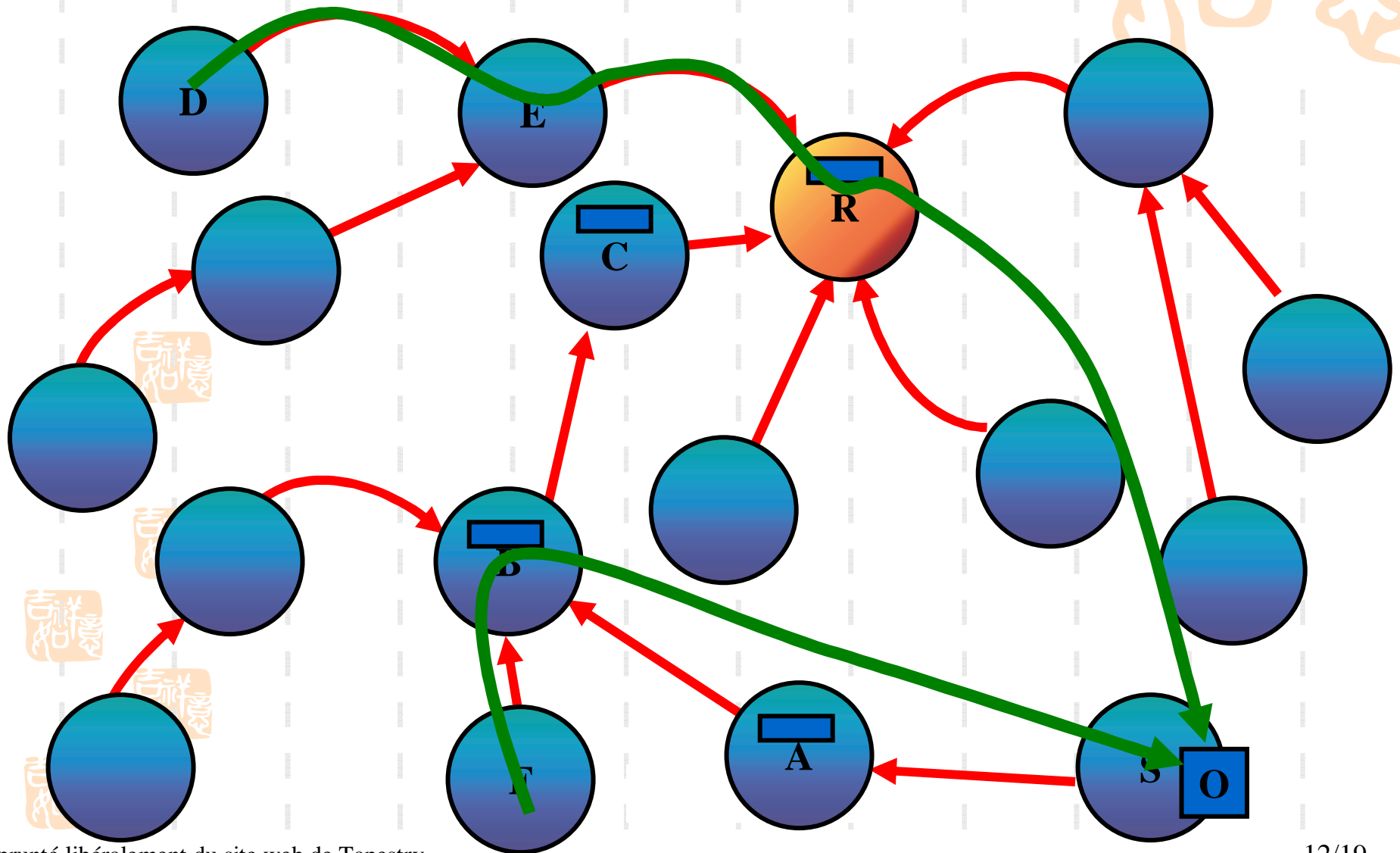
吉祥



La location d'objet



La location d'objet



L'insertion de noeud



Algorithmes

DOLR

Routage
et location

Noeud
dynamique

- L'insertion pour un nouveau noeud N doit accomplir :
 - Les noeuds qui ont besoin d'être alertés doivent être notifiés
 - N peut devenir la nouvelle racine pour des objets.
 - N doit construire sa table de routage
 - Noeuds près de N peuvent mettre N dans leurs tables de routage comme une optimisation

Contexte

Système

Expérimentation

Conclusion

Bibliographie

La suppression de noeud



Algorithmes

DOLR

- Volontaire
 - Noeuds backpointés sont notifiés, qui fixent leurs tables de routage et républient les objets
- Non volontaire
 - Palpitations: la détection des liens ratés initialise la réparation de maille de routage
 - Publication de Soft-state: pointers d'objet partent si ne sont pas républiés

Routage
et location

Noeud
dynamique

Contexte

Système

Expérimentation

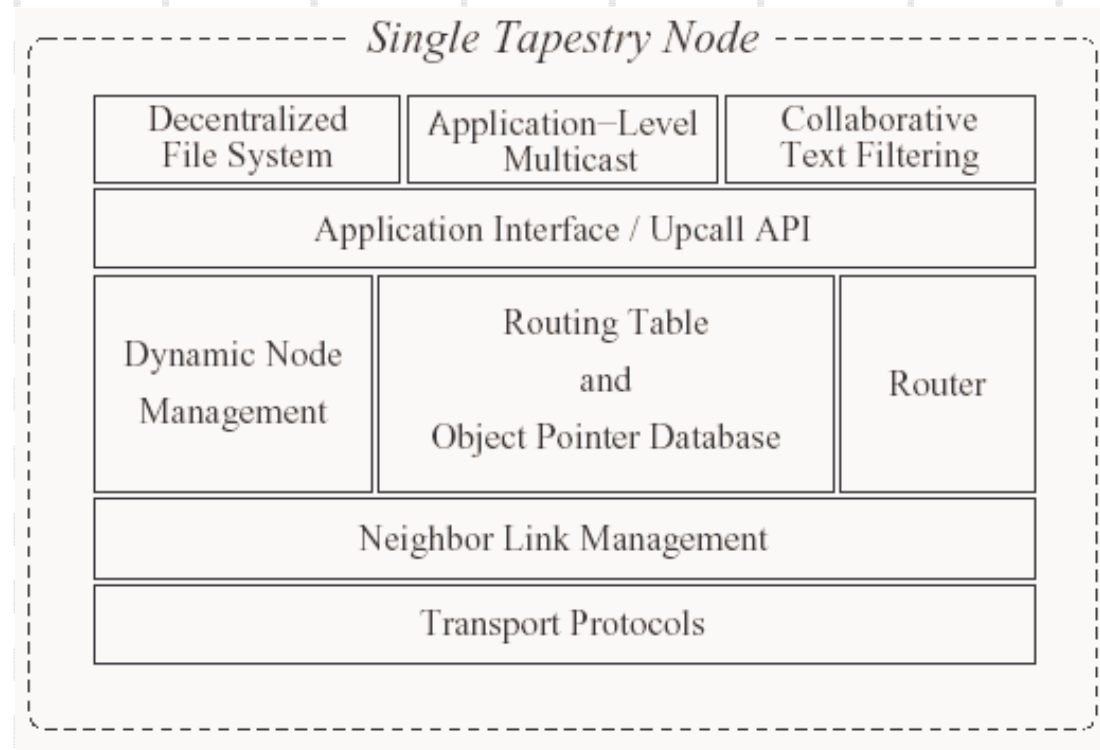
Conclusion

Bibliographie

L'architecture



Architecture



Le prototype implémenté avec Java

Contexte

Systeme

Expérimentation

Conclusion

Bibliographie

Expérimentations



Expérimentation

- Environnements
 - Cluster local, PlanetLab

Résultats:

- Tests de Routage/Location d'objet
 - La route passée dans le recouvrement est 2 fois plus longue que celle dan IP
 - Pointers d'objet aident efficacement la location d'objet
- Dynamiques de réseaux
 - La latence escalade similairement avec la taille de réseau (stabilité)
 - Convergence des fautes, une immersion brève dans le taux de succès de la recherche qui est suivie par se retourner tout de suite à presque 100% de taux de succès



Contexte

Système

Expérimentation

Conclusion

Bibliographie

Conclusion



Conclusion

- Points forts
 - Pouvoir passer à grande échelle et peut garantir la location d'objet
 - Construire localement des tables optimisées d'initialisation et les maintenir pour minimiser le « stretch »
 - Les poiteurs sont utilisée pour faciliter le routage
 - Provider des interfaces d'application puissantes
- Points faibles
 - Des expérimentations sont déployées avec entre 500 et 1000 noeuds, ce n'est pas suffisant de démontrer la capacité de passer à échelle mondiale
- Perspectives
 - Minimiser le « stretch » avec des optimisations

Stretch: le taux entre la distance passé par un message à un pair et la distance minimale de la source à ce pair

Contexte

Système

Expérimentation

Conclusion

Bibliographie

Bibliographie



Bibliographie

- [1] B.Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph J. D. Kubiatoicz, "Tapestry: A resilient Global-Scale Overlay for Service Deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41-53, 2004.
- [2] B. Y. Zhao, J. D. Kubiatoicz, and A. D. Joseph, "Tapestry: An infrastructure for fault tolerant wide area location and routing," Univ. California, Berkeley, CA, Tech. Rep. CSD-01 1141, Apr. 2001.
- [3] K. Hildrum, J. D. Kubiatoicz, S. Rao, and B. Y. Zhao, "Distributed object location in a dynamic network," in *Proc. SPAA*, Winnipeg, Canada, Aug. 2002, pp. 41–52.



Contexte

Système

Expérimentation

Conclusion

Bibliographie

Merci !

吉祥



Et des questions ?