

Modélisation et gestion de ressources dans les services Internet

Jean Arnaud Sara Bouchenak
{*Jean.Arnaud, Sara.Bouchenak*}@inria.fr

Université Grenoble I - INRIA/LIG

Journées EMSoc
23 octobre 2008

Avant de commencer...

Disclaimer

- On ne va pas parler de systèmes embarqués...
- mais des problématiques sont communes :
 - Qualité de service
 - Optimisation
 - ...

Exemples de services Internet...

ebay.fr Acheter Vendre Mon eBay Communauté Aide

Bonjour ! Ouvrir une session ou s'inscrire. Plan du site

Toutes les catégories Rechercher Recherche avancée

Catégories eBay eBay Auto

Catégories

- Art et Antiquités
- Automobiles et Pièces
- Auto - Pièces et accessoires
- Bateau, Voile, Nautisme
- Beauté, Bien-être, Parfum
- Bébé, Puériculture
- Bijoux Montres
- CD, Vinyles et Musique
- Céramiques et Vitrerie
- Collections
- DVD et Cinéma
- Immobilier
- Informatique et PC
- Instrument de Musique
- Jouets et Figurines
- Jouets Vidéo Consoles
- Livre, BD et Revues
- Luxe, Créatif
- Maison, Jardin, Bricolage
- Monnaies
- Motos et Scooters
- Moto - Pièces et accessoires
- Photo et Vidéo
- PNE Artisans, Agriculteurs
- Sports et Vacances
- Téléphone
- Tir à l'arc
- TV, Son, Home-Cinéma
- Vêtements et Accessoires
- Vins et Gastronomie
- Toutes les catégories
- annonces Auto

Inscrivez-vous
C'est facile et GRATUIT !

Déjà inscrit ? Ouvrez une session
Professionnels ? Inscrivez-vous

Faites Des Paires !

High Tech Sport

Manettes Table-tennis Basket Hobbies

Plage Auto

Lunettes de soleil Palmiers Peaux Escabe-glaces

Besoin d'aide ?

Contactez-nous
Envoyez un email à notre service consommateur

Débuter en confiance

eBay expliqué
Tout savoir sur eBay

Les évaluations
Pour connaître la réputation de vos interlocuteurs

Le paiement sur eBay
Payez par carte bancaire avec PayPal

Vendre sur eBay

Décoverez quoi vendre
On a tous quelque chose à vendre sur eBay !

Comment débiter ?
Apprenez à vendre sur eBay : c'est simple !

L'Espace Professionnel
Pour débiter votre activité sur eBay, tout est là !

La contrefaçon est illégale
Ensemble arrêtons-la !

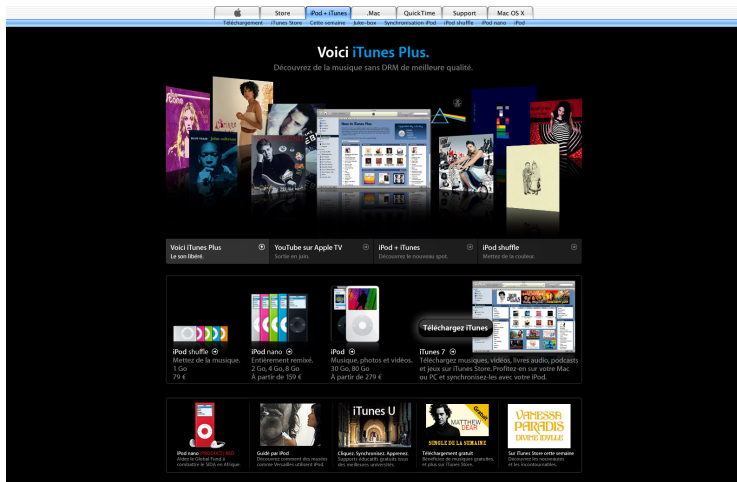
Espace Sécurité sur eBay
Tous les conseils Cliquez ici

n'attendez plus que vous !
Cliquez ici

Allemagne | Australie | Autriche | Belgique | Canada | Chine | Corée | Espagne | États-Unis | Hong-Kong | Inde | Islande | Italie | Japon | Nouvelle-Zélande | Pays-Bas | Philippines | Pologne | Royaume-Uni | Singapour | Suède | Suisse | Taiwan | Turquie

PayPal Programme eBay des développeurs Programme d'affiliation Opportunités de carrière
Annonces Kijiji Shopping.com Skype

Exemples de services Internet...



Exemples de services Internet...

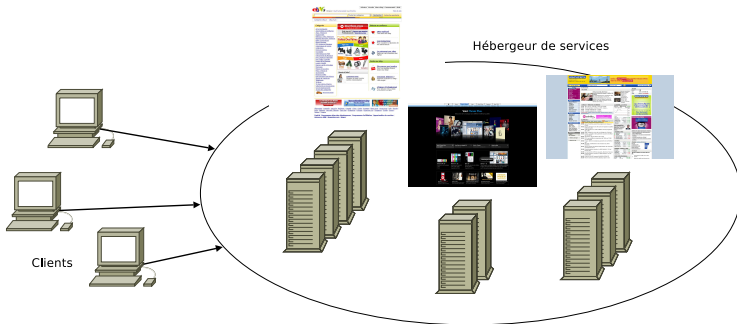
The screenshot displays the Boursorama website interface. At the top, there's a navigation bar with the Boursorama logo and a search bar. Below this, a sidebar on the left offers various service categories like 'Accès client', 'Bourse', 'Épargne', and 'Crédits'. The main content area is divided into several sections: 'LES TITRES' featuring an article about China's bond yield, 'CONSEILS & ANALYSES TECHNIQUES' with market news, 'ECONOMIE : Les indices' showing a line graph for the CAC 40, 'ACTUALITES' with a list of financial news items, 'Prêt immobilier' with a calculator, 'PARIS-MARCHÉ PARIS' with a table of market indicators, 'FUTURES' with a table of futures contracts, 'PETROLE / DEBITES / TAUX', and 'SELECTION clicklogiciels'. On the right side, there's a 'Ma liste' section and a 'boursorama.warrants' advertisement. The bottom of the page features a 'sandages' section with a bar chart and a 'Après la baisse récente du Cac 40' section with a line graph.

Hébergeur

Hébergeur de services Internet

Combien de ressources allouer au service pour :

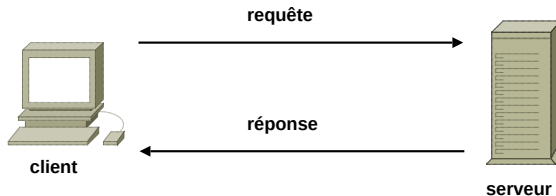
- garantir de bonnes performances au service
- minimiser le coût du service (en nombre de ressources)



Plan

- ① Problématique
- ② **Définitions / rappels**
 - **Organisation d'un service Internet**
 - Paramètres de configuration d'un service Internet
- ③ Principes de conception
- ④ Evaluation
- ⑤ Conclusion et perspectives

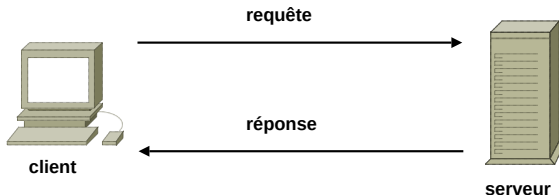
Modèle client serveur et critères de performances



Latence

Temps entre le moment où un client envoie une requête au système, et le moment où il reçoit la réponse (en secondes).

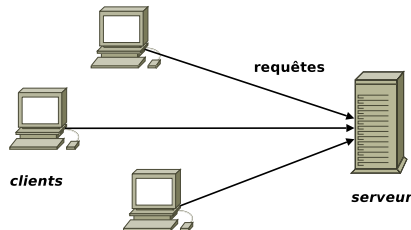
Modèle client serveur et critères de performances



Latence

Temps entre le moment où un client envoie une requête au système, et le moment où il reçoit la réponse (en secondes).

Modèle client serveur et critères de performances



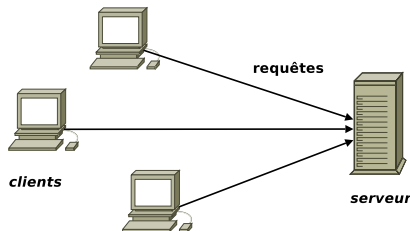
Débit

Nombre de requêtes servies par le système par unité de temps (en requêtes / secondes).

Taux d'échec

Ratio de requêtes ne pouvant pas être traitées correctement par le serveur (en %)

Modèle client serveur et critères de performances



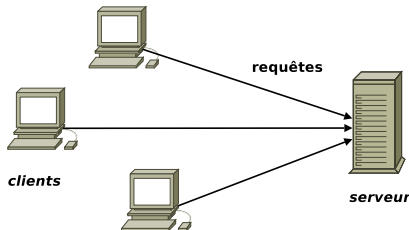
Débit

Nombre de requêtes servies par le système par unité de temps (en requêtes / secondes).

Taux d'échec

Ratio de requêtes ne pouvant pas être traitées correctement par le serveur (en %)

Modèle client serveur et critères de performances



Débit

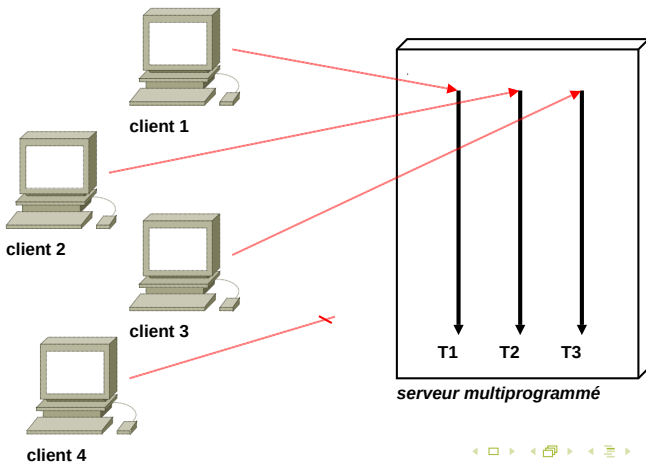
Nombre de requêtes servies par le système par unité de temps (en requêtes / secondes).

Taux d'échec

Ratio de requêtes ne pouvant pas être traitées correctement par le serveur (en %)

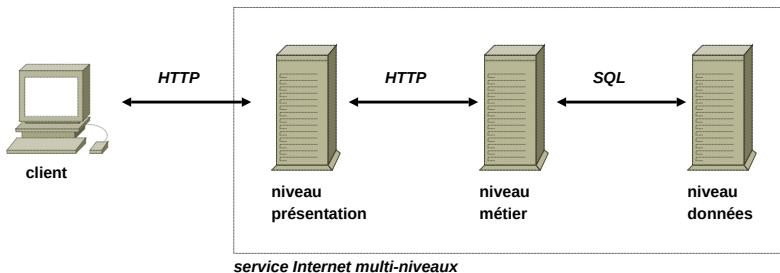
Multiprogrammation

- La plupart des serveurs sont multiprogrammés pour traiter plusieurs requêtes en parallèle



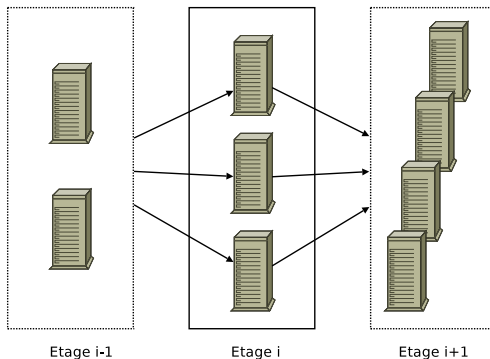
Serveurs en série : services Internet multi-niveaux

- Séparation de chaque niveau du service pour gérer la complexité
- Architectures multi-niveaux (standard J2EE)



Duplication

- A chaque niveau du système, on peut dupliquer les serveurs
 - passage à l'échelle
 - performances



Plan

- ① Problématique
- ② **Définitions / rappels**
 - Organisation d'un service Internet
 - **Paramètres de configuration d'un service Internet**
- ③ Principes de conception
- ④ Evaluation
- ⑤ Conclusion et perspectives

Paramètre de configuration au niveau local : MPL

Degré de multiprogrammation maximal
(*MultiPrograming Level (MPL)*)

Le MPL définit le nombre maximum de requêtes autorisées à s'exécuter en parallèle dans le système [4]

- MPL trop bas : débit limité
- MPL trop haut : l'écroulement (*trashing*) du serveur survient

Paramètre de configuration au niveau local : MPL

Degré de multiprogrammation maximal
(*MultiPrograming Level (MPL)*)

Le MPL définit le nombre maximum de requêtes autorisées à s'exécuter en parallèle dans le système [4]

- MPL trop bas : débit limité
- MPL trop haut : l'écroulement (*trashing*) du serveur survient

Paramètre de configuration au niveau local : MPL

Degré de multiprogrammation maximal
(*MultiPrograming Level (MPL)*)

Le MPL définit le nombre maximum de requêtes autorisées à s'exécuter en parallèle dans le système [4]

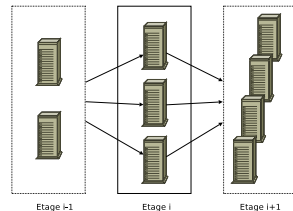
- MPL trop bas : débit limité
- MPL trop haut : l'écroulement (*trashing*) du serveur survient

Paramètre de configuration au niveau architectural : degré de duplication

Degré de duplication

Le degré de duplication est le nombre de duplicatas à un niveau de l'application

- Trop bas :
 - ⊕ coût plus faible
 - ⊖ mauvaises performances rendues au client
- Trop haut :
 - ⊕ meilleures performances
 - ⊖ coût élevé du service

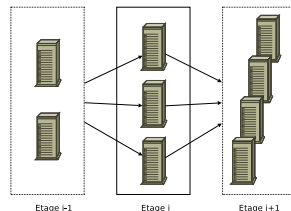


Paramètre de configuration au niveau architectural : degré de duplication

Degré de duplication

Le degré de duplication est le nombre de duplicatas à un niveau de l'application

- Trop bas :
 - ⊕ coût plus faible
 - ⊖ mauvaises performances rendues au client
- Trop haut :
 - ⊕ meilleures performances
 - ⊖ coût élevé du service

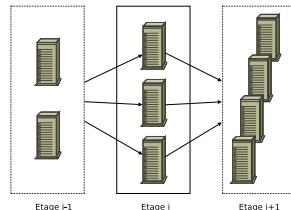


Paramètre de configuration au niveau architectural : degré de duplication

Degré de duplication

Le degré de duplication est le nombre de duplicatas à un niveau de l'application

- Trop bas :
 - ⊕ coût plus faible
 - ⊖ mauvaises performances rendues au client
- Trop haut :
 - ⊕ meilleures performances
 - ⊖ coût élevé du service



Contrat de qualité de service

Point de vue du **fournisseur** de service

- Contrat spécifiant les **performances** requises du service
- entre hébergeur et fournisseur de service
- Exemples de contrat :
 - $\text{lat} \leq \text{lat}_{max}$
 - $\text{taux_echec} \leq \text{taux_echec}_{max}$

Point de vue de l'**hébergeur** de service

- Minimiser le coût de fonctionnement d'un service hébergé

⇒ Maximiser les **performances** en minimisant le **coût**

Contrat de qualité de service

Point de vue du **fournisseur** de service

- Contrat spécifiant les **performances** requises du service
- entre hébergeur et fournisseur de service
- Exemples de contrat :
 - $lat \leq lat_{max}$
 - $taux_echec \leq taux_echec_{max}$

Point de vue de l'**hébergeur** de service

- Minimiser le coût de fonctionnement d'un service hébergé
- ⇒ Maximiser les **performances** en minimisant le **coût**

Plan

- 1 Problématique
- 2 Définitions / rappels
- 3 **Principes de conception**
- 4 Evaluation
- 5 Conclusion et perspectives

Principes de conception

Modélisation du service Internet

- Entrées : configuration du service Internet *et* charge appliquée
- Sortie : performances et taux d'échec
- Précision (proximité avec système réel)
- Efficacité (temps de calcul)

Méthode de recherche d'une configuration optimale/optimum

- Entrées : contrat de qualité de service *et* charge à gérer
- Sortie : configuration optimale
- Précision (configuration la plus proche de l'optimal possible)
- Efficacité (temps de calcul)

Principes de conception

Modélisation du service Internet

- Entrées : configuration du service Internet *et* charge appliquée
- Sortie : performances et taux d'échec
- Précision (proximité avec système réel)
- Efficacité (temps de calcul)

Méthode de recherche d'une configuration optimale/optimum

- Entrées : contrat de qualité de service *et* charge à gérer
- Sortie : configuration optimale
- Précision (configuration la plus proche de l'optimal possible)
- Efficacité (temps de calcul)

Principes de conception

Modélisation du service Internet

- Entrées : configuration du service Internet *et* charge appliquée
- Sortie : performances et taux d'échec
- Précision (proximité avec système réel)
- Efficacité (temps de calcul)

Méthode de recherche d'une configuration optimale/optimum

- Entrées : contrat de qualité de service *et* charge à gérer
- Sortie : configuration optimale
- Précision (configuration la plus proche de l'optimal possible)
- Efficacité (temps de calcul)

Principes de conception

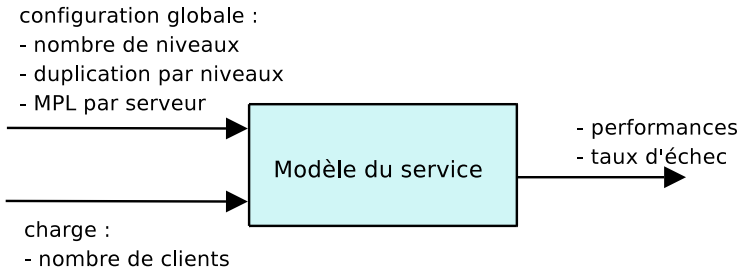
Modélisation du service Internet

- Entrées : configuration du service Internet *et* charge appliquée
- Sortie : performances et taux d'échec
- Précision (proximité avec système réel)
- Efficacité (temps de calcul)

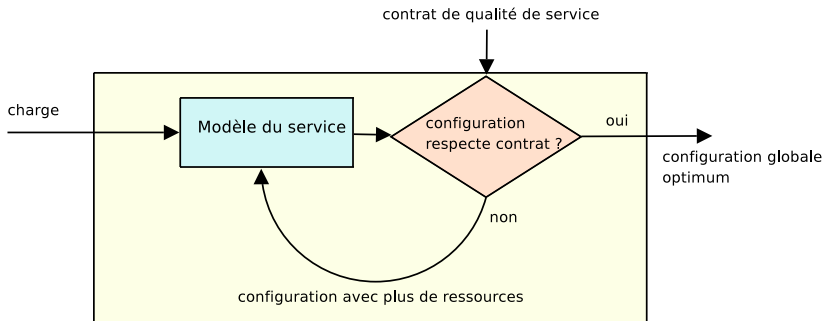
Méthode de recherche d'une configuration optimale/optimum

- Entrées : contrat de qualité de service *et* charge à gérer
- Sortie : configuration optimale
- Précision (configuration la plus proche de l'optimal possible)
- Efficacité (temps de calcul)

Interfaces du modèle



Méthode de recherche d'une configuration optimale



Algorithme de recherche de configuration optimale

MoKa : Modeling and capacity planning for Internet services

- Nous avons développé un canevas logiciel permettant de calculer la configuration optimale (ou proche de l'optimal) d'un service Internet [1]
- Organisation de MoKa
 - un module contenant l'implantation des modèles
 - un module contenant l'implantation des méthodes de recherche
 - un module de comparaison des différents modèles / méthodes
- Objectif :
 - Pour un modèle, comparer plusieurs méthodes (critères : précision et surcoût)
 - Pour une méthode, comparer plusieurs modèles (critères : précision et surcoût)

MoKa : Modeling and capacity planning for Internet services

- Nous avons développé un canevas logiciel permettant de calculer la configuration optimale (ou proche de l'optimal) d'un service Internet [1]
- Organisation de MoKa
 - un module contenant l'implantation des modèles
 - un module contenant l'implantation des méthodes de recherche
 - un module de comparaison des différents modèles / méthodes
- Objectif :
 - Pour un modèle, comparer plusieurs méthodes (critères : précision et surcoût)
 - Pour une méthode, comparer plusieurs modèles (critères : précision et surcoût)

- Nous avons développé un canevas logiciel permettant de calculer la configuration optimale (ou proche de l'optimal) d'un service Internet [1]
- Organisation de MoKa
 - un module contenant l'implantation des modèles
 - un module contenant l'implantation des méthodes de recherche
 - un module de comparaison des différents modèles / méthodes
- Objectif :
 - Pour un modèle, comparer plusieurs méthodes (critères : précision et surcoût)
 - Pour une méthode, comparer plusieurs modèles (critères : précision et surcoût)

Exemples d'implantation de modèles et méthodes

Actuellement, nous avons implanté et intégré à MoKa :

- Un modèle de service Internet :
 - MVA (Mean Value Analysis)
- 3 méthodes de recherche de configuration :
 - Recherche exhaustive
 - Recherche incrémentale
 - Recherche dichotomique

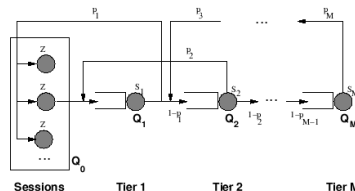
Exemples d'implantation de modèles et méthodes

Actuellement, nous avons implanté et intégré à MoKa :

- Un modèle de service Internet :
 - MVA (Mean Value Analysis)
- 3 méthodes de recherche de configuration :
 - Recherche exhaustive
 - Recherche incrémentale
 - Recherche dichotomique

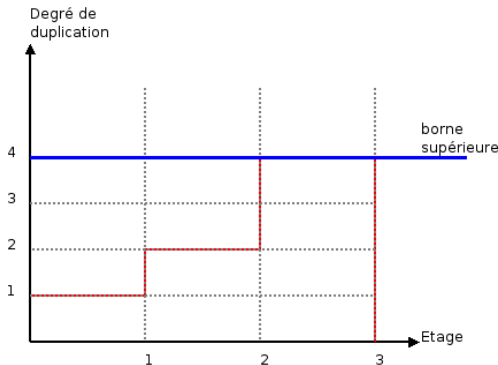
Modèle utilisé : Modèle MVA

- L'algorithme MVA [5] permet d'estimer le temps de réponse d'un réseau de files d'attente
- Prédiction du temps de réponse de l'application à partir de sa configuration
- Nous faisons les hypothèses suivantes :
 - Répartiteur de charge équilibrable (même charge sur tous les serveurs)
 - Machines homogènes
- Extension de l'algorithme MVA pour introduire la duplication



1 - Recherche exhaustive

- 1- Recherche d'une borne de duplication maximale

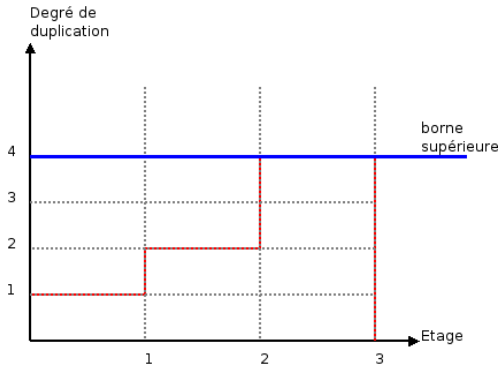


- 2- Recherche exhaustive dans l'espace restant : essayer toutes les combinaisons possibles.

Ex : [1,1,1] [1,1,2] ... [4,4,4]

1 - Recherche exhaustive

- 1- Recherche d'une borne de duplication maximale

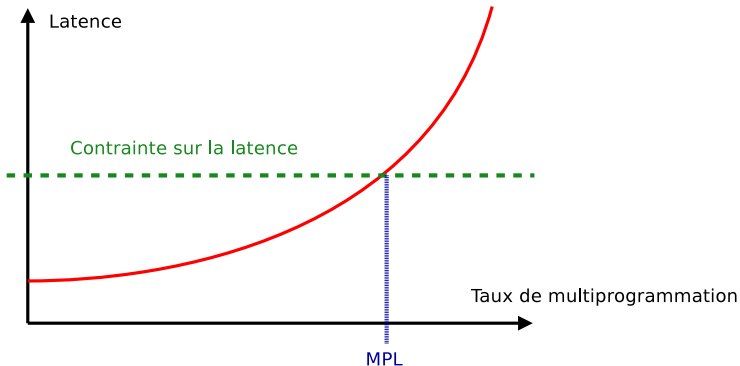


- 2- Recherche exhaustive dans l'espace restant : essayer toutes les combinaisons possibles.

Ex : [1,1,1] [1,1,2] ... [4,4,4]

1 - Recherche exhaustive

● 3 - Recherche de la configuration locale



2 - Recherche incrémentale

- Proposé par Urgaonkar et al. dans [6]
- Pour chaque ressource, calculer sur quel étage son allocation donnerait la meilleure amélioration de performances.
⇒ Essayer [1,1,2], [1,2,1] puis [2,1,1] et continuer avec la meilleure configuration
- Arrêter l'allocation lorsque les contraintes de performances sont vérifiées
- Recherche de la configuration locale

3 - Recherche dichotomique

- Recherche d'une borne de duplication maximale
- Recherche dichotomique dans l'espace restant, plus efficace que les recherches précédentes
- Recherche de la configuration locale
- $\Rightarrow$ Gain par rapport à la recherche incrémentale : entre M (pire cas) et M^2 (meilleur cas) ¹

¹où M est le nombre d'étage de l'application

3 - Recherche dichotomique

- Recherche d'une borne de duplication maximale
- Recherche dichotomique dans l'espace restant, plus efficace que les recherches précédentes
- Recherche de la configuration locale
- $\Rightarrow$ Gain par rapport à la recherche incrémentale : entre M (pire cas) et M^2 (meilleur cas) ¹

¹où M est le nombre d'étage de l'application

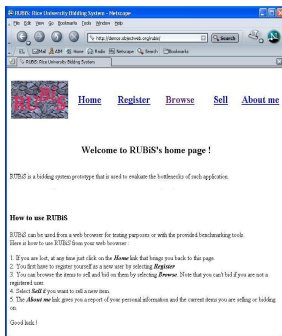
Plan

- 1 Problématique
- 2 Définitions / rappels
- 3 Principes de conception
- 4 **Evaluation**
- 5 Conclusion et perspectives

Environnement d'évaluation

- Grappe de Lille de Grid'5000 :
 - Bi-opterons (2,2 et 2,6 GHz)
 - 4 Go mémoire, gigabit Ethernet
- Banc d'essai RUBiS (*Rice University Bidding System*)
- Modélisé d'après eBay
- 26 types d'interaction (inscription, consultation, enchère, vente ...)
- Respecte les normes TPC-W^a
- 2 types de trafic :
 - *browsing mix* : 100% de requêtes Web en lecture seule
 - *bidding mix* : 85% de requêtes Web en lecture et 15% en écriture

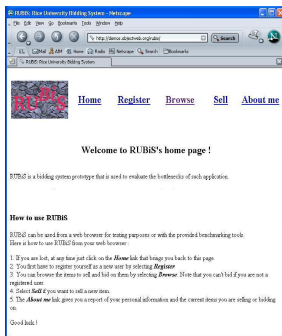
^a<http://www.tpc.org>



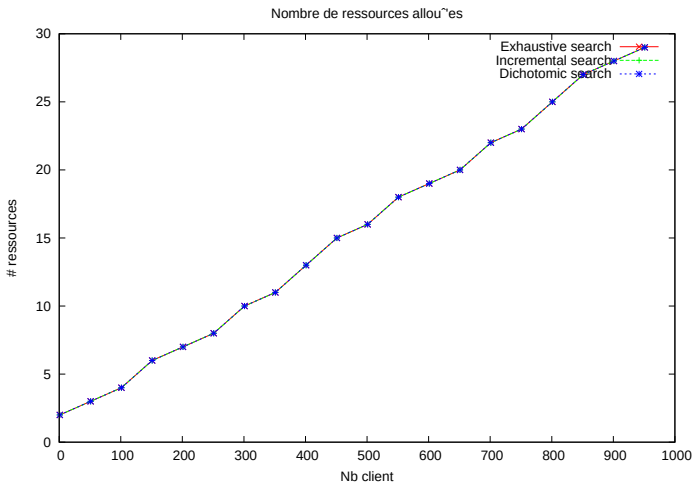
Environnement d'évaluation

- Grappe de Lille de Grid'5000 :
 - Bi-opterons (2,2 et 2,6 GHz)
 - 4 Go mémoire, gigabit Ethernet
 - Banc d'essai RUBiS (*Rice University Bidding System*)
- Modélisé d'après eBay
 - 26 types d'interaction (inscription, consultation, enchère, vente ...)
 - Respecte les normes TPC-W^a
 - 2 types de trafic :
 - *browsing mix* : 100% de requêtes Web en lecture seule
 - *bidding mix* : 85% de requêtes Web en lecture et 15% en écriture

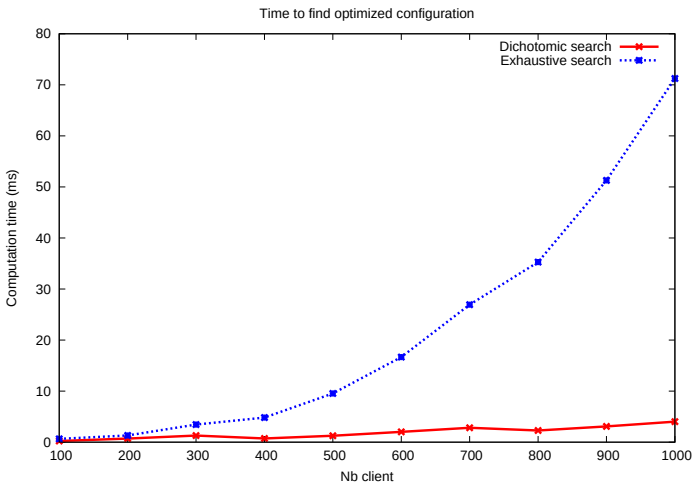
^a<http://www.tpc.org>



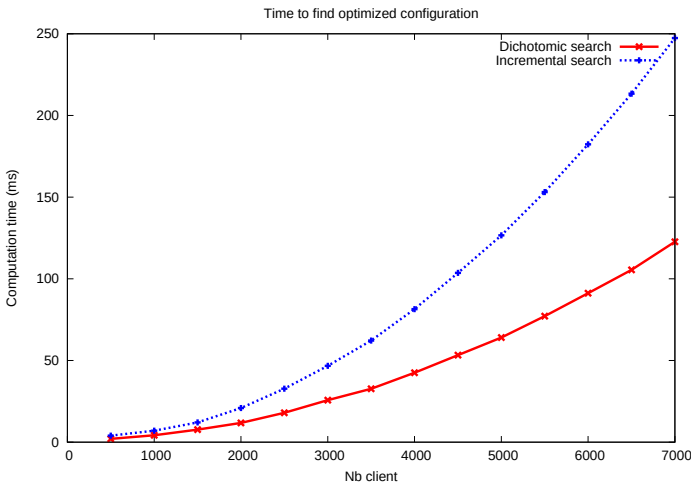
Prédictions des algorithmes



Coût des algorithmes



Coût des algorithmes



Plan

- 1 Problématique
- 2 Définitions / rappels
- 3 Principes de conception
- 4 Evaluation
- 5 **Conclusion et perspectives**

Conclusion

- Approche basée sur modèle pour l'allocation de ressources pour des services Internet
- Proposition d'un algorithme de recherche
 - plus efficace que l'existant : gain d'un facteur M/M^2 par rapport à la méthode incrémentale
 - aussi précis : vérifié par l'expérience, reste à prouver
- Conception du framework MoKa pour intégrer différents modèles et algorithmes
 - Définition d'interfaces génériques modèle/algorithmes
 - Méthodes de comparaison pour les modèles et les algorithmes

Conclusion

- Approche basée sur modèle pour l'allocation de ressources pour des services Internet
- Proposition d'un algorithme de recherche
 - plus efficace que l'existant : gain d'un facteur M/M^2 par rapport à la méthode incrémentale
 - aussi précis : vérifié par l'expérience, reste à prouver
- Conception du framework MoKa pour intégrer différents modèles et algorithmes
 - Définition d'interfaces génériques modèle/algorithmes
 - Méthodes de comparaison pour les modèles et les algorithmes

Conclusion

- Approche basée sur modèle pour l'allocation de ressources pour des services Internet
- Proposition d'un algorithme de recherche
 - plus efficace que l'existant : gain d'un facteur M/M^2 par rapport à la méthode incrémentale
 - aussi précis : vérifié par l'expérience, reste à prouver
- Conception du framework MoKa pour intégrer différents modèles et algorithmes
 - Définition d'interfaces génériques modèle/algorithmes
 - Méthodes de comparaison pour les modèles et les algorithmes

Perspectives

- Implanter d'autres modèles plus efficaces/précis :
 - Heiss et al. [3] : approximation parabolique
 - NECS (projet INRIA)
- Prouver optimalité des configurations calculées
(Mêmes prédictions des algorithmes exhaustifs et dichotomiques)
- Intégration dans une boucle de contrôle pour l'adaption
dynamique de la configuration à la charge du système

Perspectives

- Implanter d'autres modèles plus efficaces/précis :
 - Heiss et al. [3] : approximation parabolique
 - NECS (projet INRIA)
- Prouver optimalité des configurations calculées
(Mêmes prédictions des algorithmes exhaustifs et dichotomiques)
- Intégration dans une boucle de contrôle pour l'adaption
dynamique de la configuration à la charge du système

Perspectives

- Implanter d'autres modèles plus efficaces/précis :
 - Heiss et al. [3] : approximation parabolique
 - NECS (projet INRIA)
- Prouver optimalité des configurations calculées
(Mêmes prédictions des algorithmes exhaustifs et dichotomiques)
- Intégration dans une boucle de contrôle pour l'adaption dynamique de la configuration à la charge du système



Jean Arnaud and Sara Bouchenak.

Gestion de ressources dans les services internet.

In *CFSE*, Fribourg, Suisse, February 2008.



Sara Bouchenak, Noël De Palma, Daniel Hagimont, and
Christophe Taton.

Autonomic management of clustered applications.

In *IEEE International Conference on Cluster Computing*,
Barcelona, Spain, September 2006.



Hans-Ulrich Heiss and Roger Wagner.

Adaptive load control in transaction processing systems.

In *VLDB*, pages 47–54, 1991.



J.M. Milan-Franco, Ricardo Jiménez-Peris, Marta
Patino-Martinez, and Bettina Kemme.

Adaptive middleware for data replication.

In *Middleware '04 : Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 175–194, New York, NY, USA, 2004. Springer-Verlag New York, Inc.



M. Reiser and S. S. Lavenberg.

Mean-value analysis of closed multichain queuing networks.
J. ACM, 27(2) :313–322, 1980.



Bhuvan Urgaonkar, Giovanni Pacifici, Prashant Shenoy, Mike Spreitzer, and Asser Tantawi.

Analytic modeling of multitier internet applications.
ACM Trans. Web, 1(1) :2, 2007.

Jade [2]

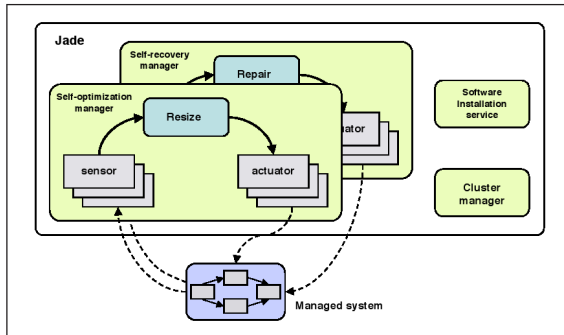


FIG.: Boucles de commande dans Jade

Comportement dynamique d'un serveur multiprogrammé [3]

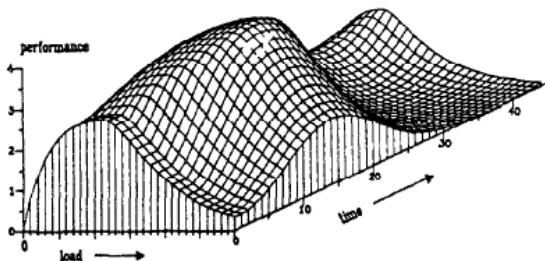


FIG.: Comportement dynamique d'un serveur multiprogrammé [3]

Niveau architectural : algorithme MVA [5]

Entrées : $N, \bar{S}_m, V_m, r_m, 1 \leq m \leq M, \bar{Z}$

Sorties : $\bar{R}_m$ (latence moyenne sur la file Q_m), $\bar{R}$ (temps de réponse moyen)

```
1  début
2       $\bar{R}_0 = \bar{D}_0 = \bar{Z}$ 
3       $\bar{L}_0 = 0$ 
4      pour  $m = 1$  à  $M$  faire
5           $\bar{L}_m = 0$ 
6           $\bar{D}_m = \frac{V_m \cdot \bar{S}_m}{r_m}$  /* demande de service */
7      /* on insère les  $N$  clients un par un */
8      pour  $n = 1$  à  $N$  faire
9          pour  $m = 1$  à  $M$  faire
10              $\bar{R}_m = \bar{D}_m \cdot (1 + \bar{L}_m)$  /* latence moyenne */
11              $\tau = \left( \frac{n}{\bar{R}_0 + \sum_{m=1}^M \bar{R}_m} \right)$  /* débit */
12             pour  $m = 1$  à  $M$  faire
13                  $\bar{L}_m = \tau \cdot \bar{R}_m$  /* loi de Little */
14              $\bar{L}_0 = \tau \cdot \bar{R}_0$ 
15          $\bar{R} = \sum_{m=1}^M \bar{R}_m$  /* temps de réponse moyen */
16  fin
```

Modèle local - lectures seules

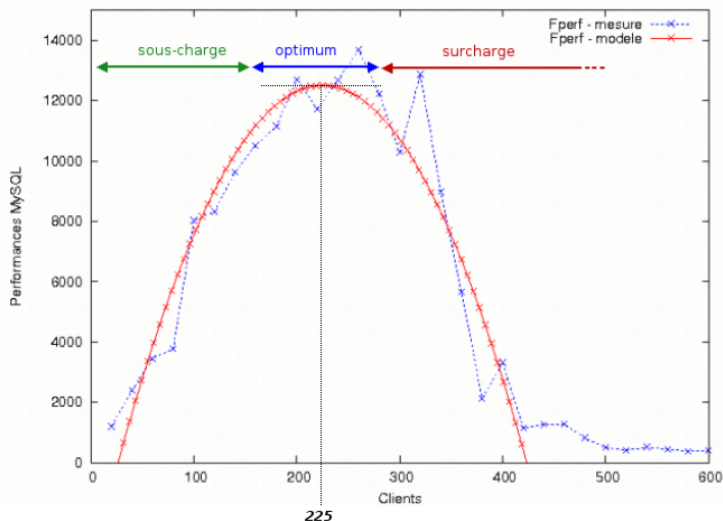


FIG.: Performances de l'étage données (MySQL) - *browsing mix*

Modèle local - trafic avec écritures

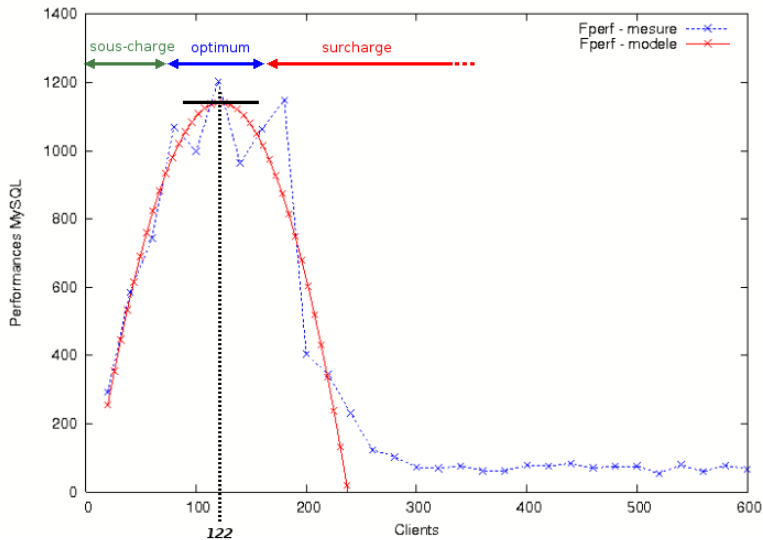


FIG.: Performances de l'étage données (MySQL) - *bidding mix*

Validation du modèle architectural

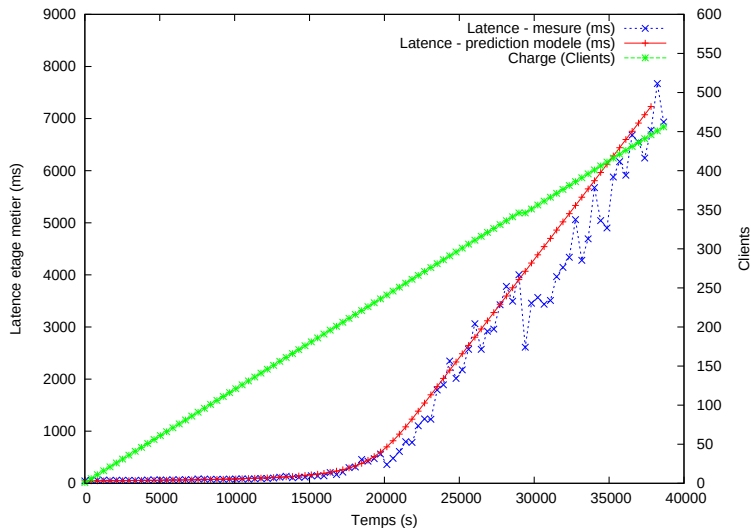


FIG.: Prédiction du modèle et mesure réelle