



it-expert

QUALITE OPERATIONNELLE DU LOGICIEL

PAGE 6

J2EE 1.4
PAGE 33

XP, YES YOU CAN

PAGE 53

LES OUTILS DE TESTS

PAGE 39

IT-expert n°42 - Mars/Avril 2003

Image PostScript
(EPSMIC)

Capacity Planning

PAGE 19

www.it-expertise.com



IT-expert, 60 pages

pour rendre vos équipes informatiques plus performantes !

BASES DE DONNÉES
ARCHITECTURE
MÉTHODES
LOGICIELS

*IT-expert, la référence technique
des professionnels de l'informatique*

*Bimestriel de conseil et d'expertise technique,
IT-expert vous offre l'information essentielle
pour vous former et décider.*

Pour tous renseignements :

IT-expert - 3, rue Marcel Allégot - 92190 MEUDON - FRANCE

Tél. : +33 (0)1 46 90 21 21

e-mail : abonnement@it-expertise.com

LA RÉFÉRENCE TECHNIQUE DES PROFESSIONNELS DE L'INFORMATIQUE
it-expert

édito

LA RÉFÉRENCE TECHNIQUE DES PROFESSIONNELS DE L'INFORMATIQUE
it-expert

EDITEUR

Press & Communication France
Une filiale du groupe CAST
3, rue Marcel Allégot
92190 Meudon - FRANCE
Tél. : 01 46 90 21 21
Fax. : 01 46 90 21 20
[http : //www.it-expertise.com](http://www.it-expertise.com)
Email : redaction@it-expertise.com

RÉDACTEUR EN CHEF

Jean-Pierre Forestier
Email : jp.forestier@it-expertise.com

ABONNEMENTS/PUBLICITE

Aurélié Magniez
Email : a.magniez@it-expertise.com

MAQUETTISTE

C. Grande
Email : c.grande@it-expertise.com

PHOTO

Eric Maréchal (page 39)

IMPRIMEUR

MOUTOT Imprimeurs

PARUTION

IT-expert - (ISSN 1270-4881) est un journal édité 6 fois par an, par P & C France, sarl de presse au capital de 60976,61 €.

AVERTISSEMENT

Tous droits réservés. Toute reproduction intégrale ou partielle des pages publiées dans la présente publication sans l'autorisation écrite de l'éditeur est interdite, sauf dans les cas prévus par les articles 40 et 41 de la loi du 11 mars 1957. © 1996 P&C France. Toutes les marques citées sont des marques déposées. Les vues et opinions présentées dans cette publication sont exprimées par les auteurs à titre personnel et sont sous leur entière et unique responsabilité. Toute opinion, conseil, autre renseignement ou contenu exprimés n'engageant pas la responsabilité de Press & Communication.

ABONNEMENTS

01 46 90 21 21
Prix pour 6 numéros (1 an)
France : 50 € TTC
U.E., DOM TOM : 62,50 € TTC
Autres Pays : 72 € TTC
Un bulletin d'abonnement se trouve en pages 51 et 52 de ce numéro.
Vous pouvez vous abonner sur
www.it-expertise.com
[/Abonnement/Abonnement2.asp](#)
ou nous écrire à
abonnement@it-expertise.com



En tant qu'utilisateurs nous appelons tous de nos vœux des logiciels de qualité. En tant que professionnels des technologies de l'information, nous affirmons, la main sur le cœur, que nous mobilisons nos énergies dans ce sens. Dans la pratique, il ne faut pas rêver et mieux vaut sans doute faire notre deuil de ce noble et utopique objectif.

La qualité n'est pas aujourd'hui un élément prioritaire des développements informatiques. Le critère

essentiel du dialogue entre client et fournisseur reste le délai d'obtention du logiciel ou de la solution qui l'englobe. Ce sacro-saint délai est toujours soumis à une multitude de forces qui tendent à le remettre en cause. Remarquez au passage qu'il n'y a jamais de « bonne » surprise. Le glissement du délai s'effectue toujours dans le même sens : celui de son étirement. Or, ce délai est une donnée toujours considérée comme intangible et impossible à remettre en cause, bref un dogme. Face au glissement, les moyens d'actions sont limités. Pour rester en conformité avec la prévision initiale, les leviers sont généralement peu nombreux. La solution retenue consiste la plupart du temps à supprimer ou remettre à plus tard les tâches qui ne sont pas absolument essentielles.

Les premières visées sont donc celles qui ne démontrent pas une productivité immédiate. Les actions ou « procédures » visant à la qualité sont évidemment les premières victimes. Généralement, elles précèdent de peu à la trappe les tests et autres activités de qualification. Plus grave, la récurrence de ces errements aboutit à un état d'esprit proprement aberrant de la part des utilisateurs et des éditeurs de logiciels. Nous en sommes venus à considérer comme normal la présence de défauts dans les logiciels que nous utilisons. Cette situation est si couramment répandue qu'elle aboutit à une véritable démarche des éditeurs ; faire assurer les tests par des utilisateurs volontaires, les fameux « bêta-testers ». Quel autre secteur de l'industrie pourrait se permettre une telle approche ? Lorsque vous faites l'acquisition d'une voiture toléreriez-vous l'éventualité d'un bogue dans le système de freinage ? Lorsque vous confiez la réalisation de votre maison à un architecte, pourrait-il se permettre de ne pas étudier la stabilité du terrain pour gagner quelques jours ? Alors pourquoi supportons-nous dans le domaine de l'informatique ce qui dans tout autre secteur déclencherait un tollé immédiat ? Toujours est-il qu'aussi longtemps que perdurera notre laxisme vis-à-vis des éditeurs, ceux-ci auront beau jeu de ne pas mettre la qualité au hit parade de leurs préoccupations.

Jean-Pierre FORESTIER
Rédacteur en Chef

Sommaire

QUALITE OPERATIONNELLE DU LOGICIEL



Construisez des systèmes à vos mesures :

le Capacity Planning



6 Dossier

Qualité opérationnelle du logiciel

Le faible niveau de qualité des logiciels aujourd'hui disponibles pose de plus en plus de questions, tant aux utilisateurs qu'aux Directions Informatiques. Ce dossier se propose de faire le point sur les méthodes et certains outils qui permettent depuis quelques années une amélioration en la matière.

19 Technique

Construisez des systèmes à vos mesures : le capacity planning

Dimensionner par avance l'infrastructure d'un futur système est une tâche qu'aucun projet ne peut plus ignorer, ne serait-ce que pour des questions d'économie. Cet article s'intéresse aux techniques qui permettent aujourd'hui de procéder à un tel calcul.

27 Entre les lignes

Cette rubrique met en lumière la politique des grands éditeurs en lisant entre les lignes les relations purement factuelles dont la presse se fait l'écho.

28 Actualités Internationales

Actualité internationale et annonces produits chez les éditeurs...

33 Quoi de neuf docteur ?

J2EE 1.4, nouvelle version

La version 1.4 de J2EE a connu un retard considérable. Elle est désormais dans sa phase finale de spécification. En voici donc enfin un aperçu.

38 Livres

Automated Software Testing et **Test Driven Development by Example** sont les deux ouvrages que nous vous recommandons ce mois-ci.

39 Comment ça marche ?

Les outils de tests

Les outils automatisant certaines parties des tests fleurissent depuis quelques années. Ils occupent une place croissante au sein de l'incontournable activité de test. Mais au fait, comment fonctionnent-ils et quelles sont leurs limites ?

46 Fenêtre sur cour

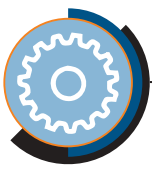
Interview KarmicSoft

Fondateur de KarmicSoft, Michel Zam possède une expérience de longue date des aspects qualitatifs liés au développement de logiciels. Il nous propose aujourd'hui de partager cette vision avec nos lecteurs.

53 Rubrique à bras

XP, yes you can

Windows XP, dernier né de la gamme des systèmes d'exploitation de Microsoft, est-il un gadget pour l'informatique domestique ou peut-il trouver sa place dans les entreprises ?



Construisez des systèmes à vos mesures :

le Capacity Planning

Mettre en place un système qui se révélera incapable d'offrir les performances attendues est la hantise de tout directeur informatique. Il est en effet particulièrement frustrant de passer des mois à écrire et mettre au point un système logiciel et de voir son travail réduit à néant parce qu'il est inutilisable dans la pratique pour « un simple problème de machine »... Il est tout aussi rageant de voir un système qui fonctionnait très bien se trouver tout à coup saturé et incapable de jouer son rôle dans l'architecture technique de l'entreprise.

Il existe pourtant des méthodes pour évaluer les performances d'une infrastructure technique.

Elles permettent d'une part de prévoir les performances et d'autre part de suivre leur évolution dans le temps afin d'anticiper d'éventuels points de rupture.

Voici donc un aperçu de ces méthodes dites « de Capacity Planning ».

Le bon dimensionnement d'un système

Voici quelques exemples qui permettent de mieux comprendre l'intérêt des études de capacity planning.

● Histoires de plombier

Soit un robinet qui fuit à raison de 15 cl à la minute. Sachant que le plombier n'arrivera pas avant 6 heures, quelle est la capacité minimale du récipient qui doit être placé sous le robinet pour éviter d'inonder la salle de bain ?

Quelle est la capacité du nouvel accumulateur d'eau pour que chaque membre d'une famille puisse prendre une douche chaude le matin ?

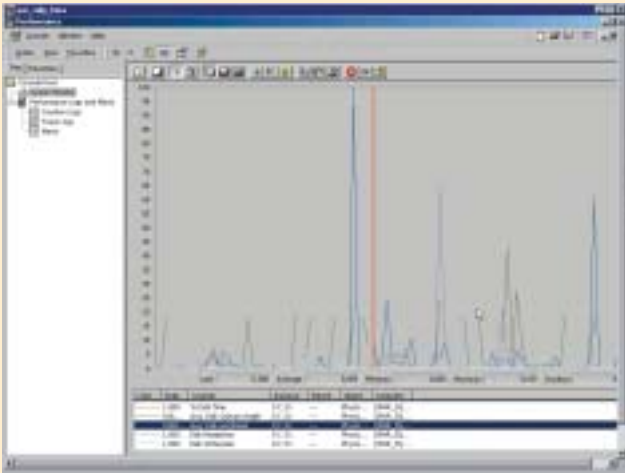
● Au supermarché...

Un nouveau centre commercial vient de se créer dans le voisinage et il a le bon goût d'être ouvert le dimanche, 300 magasins, un parking de 3000 places... Madame et



Comment applique-t-on la théorie des files d'attente à la modélisation des ordinateurs ?

Les graphes du « performance monitor » de Windows sont facilement accessibles. L'objet de cet exemple n'est pas d'interpréter les courbes de performances, mais d'illustrer quelques lois simples qui lient entre elles les valeurs collectées.



Si la compréhension de l'algorithme de la modélisation demande une bonne maîtrise de la théorie des files d'attente, les lois fondamentales sur lesquelles elles s'appuient sont très simples.

En prenant pour exemple le système d'Entrées/Sorties (E/S) d'un ordinateur cet article explique les lois de « Little » et de « l'utilisation » qui gouvernent les relations entre les indicateurs de performances retournés par des outils comme le « performance monitor » (Windows), « sar -d » ou « iostat » (Unix). Avant toute chose, un disque est un composant électro-mécanique qui ne sert qu'une E/S à la fois. Le système d'exploitation utilise un mécanisme de FIFO sur chaque disque afin de mettre en attente les E/S avant de les servir. La longueur de la file d'attente (disk queue length), le temps de séjour dans la file (avgwait dans sar -d) et le temps de réalisation de l'E/S physique (avgserv) sont facilement mesurables. Ces outils donnent enfin l'« utilisation » du disque : l'« utilisation » est la fraction par unité de temps pendant laquelle un disque sert une E/S. Une utilisation de 50 % d'un disque signifie donc que sur une période de 100ms, le disque sert des E/S pendant 50ms.

Loi de Little

Cette loi s'énonce de la façon suivante : le nombre moyen de requêtes présentes sur une ressource A est égal au produit du débit de A par le temps moyen passé par une requête sur A. Soit :

- N : Nombre moyen de requêtes présentes sur

- la ressource à un temps t,
- R : (« Résidence ») durée moyenne de présence d'une requête dans le système,
- X : Débit (nombre de transactions par unité de temps).

| | |
|---------------|------|
| Loi de Little | N=XR |
|---------------|------|

Exemple : Relations entre la longueur de la file d'attente sur un disque, le débit du disque (nombre de transactions par secondes) et le temps pris par les Entrées/Sorties :

La file d'attente mesurée sur un disque est en moyenne de 4, en comptant la requête en cours, N=5. Le disque sert 75 Entrées/Sorties par seconde.

La loi de Little nous donne donc le temps passé par une requête d'E/S, soit $R=N/X=67$ ms, ventilés en $1/75=13$ ms passées en accès du disque lui-même, et donc $67-13=54$ ms passées dans la file d'attente.

Loi de l'utilisation

Prenons un référentiel constitué d'un disque sans sa file d'attente.

- D (« Demande ») est le temps de service du disque (rotation, déplacement axial, temps de transfert sur le bus),
- X est le débit, c'est-à-dire le nombre de requêtes d'E/S traités par le disque par unité de temps,
- U est l'utilisation de la ressource.

La relation entre U, X et D s'énoncera par :

| | |
|----------------------------------------------------------|------|
| Utilisation en fonction des temps de service et du débit | U=XD |
|----------------------------------------------------------|------|

Exemples de collectes par « sar -d » sur un disque :

| Heure | Util % | Service time (ms) | Read/s | Write/s |
|-------|--------|-------------------|--------|---------|
| 15:07 | 70,8 | 10 | 4 | 67 |
| 15:12 | 67,3 | 12,3 | 30,6 | 24,1 |
| 15:17 | 67,8 | 12 | 33,7 | 22,7 |

On vérifie bien à chaque instant que :
Util %=(Read/s+Write/s) * Service time



Monsieur décide donc d'aller y faire une petite visite. En arrivant à proximité, catastrophe : les voilà pris dans un bouchon inextricable. Il leur faudra 45 minutes avant de pouvoir atteindre le parking, qui n'est pas saturé d'ailleurs. Auront-ils envie d'y retourner... pas sûr, puisqu'il est probable que ce problème se répétera chaque dimanche.

Il aurait simplement fallu prévoir des chemins d'accès avec des débits plus conséquents et permettant le passage simultané de plus de voitures.

Ce sont les soldes. Les magasins sont bondés. Pourtant, Madame s'est bien organisée cette année. Elle a fait des repérages la semaine dernière. Il suffit de prendre dans les rayons les articles sélectionnés et de se présenter à la caisse... où la durée d'attente moyenne est de 40 minutes. Désespérant !

Ces exemples de mauvais dimensionnement d'infrastructure ne portent pas à conséquence. A part pour les magasins en question qui ont sans doute perdu des clients.

● **D'autres exemples peuvent être cependant dramatiques**

Dans le service d'urgences d'un hôpital, il faudra prévoir suffisamment de lits pour pouvoir soigner les blessés lors d'un accident. Un extracteur de fumée dans un lieu confiné se doit d'être suffisamment puissant et les issues de secours suffisamment larges pour évacuer toute la clientèle en cas d'incendie.

Ces petits cas d'école mettent en lumière les deux notions principales à prendre en compte lors d'études de capacity planning :

• **Capacité :**

La capacité d'un système est relative à l'espace physique qu'il propose. Un parking de supermarché de 10000 places, une bassine 40 litres, un système de stockage de 250 Go...

• **Débit :**

Le débit d'un système est relatif au nombre d'opérations que le système est capable de fournir. Un débit de 5000 voitures par heure, un disque qui permet 80 accès par seconde. On parle aussi de bande passante.

Une bonne étude de capacity planning : un enjeu d'abord économique

Le premier but d'un système est de fournir un service correct aux utilisateurs.

Il est communément admis, par exemple, que lorsqu'un site Internet n'est pas capable d'offrir des temps de réponse inférieurs à 6 secondes, les surfeurs arrêtent la navigation... et vont voir ailleurs. Pourtant, tout bon directeur commercial le sait : un client perdu se tourne vers la concurrence et ne reviendra pas. Il est donc indispensable de pouvoir disposer de systèmes fournissant une puissance

ce nécessaire pour éviter ce type de déconvenues et ce sans attendre les premiers embouteillages.

La définition des fonctionnalités d'un système est une opération maintenant relativement bien maîtrisée. Il existe de nombreuses méthodes de conception et d'analyse, déjà décrites dans ces colonnes. De plus, il est presque toujours possible de faire évoluer un logiciel en cas d'erreurs au moment de sa conception ou si des fonctionnalités doivent être ajoutées.

Il est plus difficile, en revanche, de décrire « a priori » les attendus de la partie physique d'un système. Machines, processeurs, mémoire, réseau... pourtant, si le système ne convient pas, il est très difficile de le faire évoluer. Le plus souvent, il faut changer des composants ce qui peut avoir des conséquences importantes sur l'activité : changement de technologie, indisponibilité temporaire...

La mauvaise définition d'un système est donc doublement pénalisante pour l'économie de l'entreprise qui compte sur lui.

Alors comment prévoir ?

Pour obtenir un système correspondant à leurs attentes, les directions informatiques ont à leur disposition 4 types de méthodes.

● **La méthode « intuitive »**

Essentiellement basée sur l'intuition de celui qui mène l'étude, il s'agit d'une approche plutôt empirique.

En comparant le besoin approximatif du nouveau système à d'autres systèmes apparemment similaires, il est possible d'avoir une bonne idée de ce qu'il convient de mettre en place. Cette méthode, qui est certainement la plus utilisée, n'est adaptée qu'à des systèmes de petite dimension. En cas de doute, il suffit de « gonfler » un peu la configuration, ce qui se fait souvent en restant dans une même fourchette de prix. Ce n'est bien sûr pas envisageable dans le cadre d'importantes configurations dont le montant atteint des centaines de milliers d'euros.

● **La méthode « projective »**

Cette méthode est relativement proche de la précédente. Dans ce cas, le but est d'estimer la meilleure façon de faire évoluer un système, l'idée directrice étant de suivre la croissance de l'activité grâce à des mises à jour régulières.

L'application de cette méthode se heurte le plus souvent à l'écueil suivant : les capacités d'un système ne croissent pas de façon linéaire, rendant difficile une définition fiable des évolutions à mettre en œuvre.

● **La méthode « stressante »**

Le principe est simple. Pour vérifier qu'un système convient au but pour lequel il est prévu, il est stressé en simulant des conditions d'exploitation.



Des outils existent pour stimuler un système. Ils reposent tous sur les étapes suivantes :

- La constitution d'un jeu d'essai représentatif des opérations qui seront effectuées en phase de production,
- La création de scripts qui vont être utilisés pour stresser le système. Cette phase est particulièrement fastidieuse pour les personnes qui en sont chargées,
- Le stress du système à proprement parler, grâce aux scripts précédemment constitués,
- La validation de l'architecture par la mesure de sa réaction au stress.

Les principaux outils disponibles dans ce domaine sont : winrunner de Mercury Interactive et PATROL Perform de BMC software.

Cette méthode présente aussi un certain nombre de failles.

- L'étude est forcément longue : il faut compter plusieurs semaines (parfois plusieurs mois) de travail pour obtenir des résultats satisfaisants pour des systèmes moyennement importants.
- Il faut disposer d'un système à stresser... cela paraît évident mais cette méthode permet de valider l'adéquation d'un matériel physiquement existant à un besoin. Il n'est pas question, devant les coûts engagés, de tester une multitude de configurations. En départager deux ou trois est un maximum.

Dans de telles conditions, cette méthode doit être privilégiée dans les cas où un système doit être modifié, par exemple lorsqu'il s'agit de mettre à jour un progiciel. On valide alors le fait que la mise à jour n'aille pas provoquer une régression dans les performances globales.

● La méthode « prédictive »

Cette méthode est certainement la plus objective. L'idée est de créer un modèle mathématique du futur système afin d'en définir les caractéristiques avant de le construire. On est ainsi sûr qu'il correspondra aux besoins... si l'étude est convenablement menée. (Cf. : encadrés)

Que peut-on prévoir ?

En mettant en place une étude de type capacity planning, les équipes informatiques doivent déduire le niveau de service qu'ils doivent fournir en fonction des attentes des « clients ».

Les difficultés principales viennent des constatations suivantes :

- **La prédiction du futur n'est pas simple :** le but est d'apporter une solution concrète et tech-

nique (logicielle et matérielle) à des besoins futurs difficiles à estimer. Chaque entreprise espère décupler son activité dans les prochaines années... et que se passe-t-il si un concurrent est acheté ? L'activité du système sera-t-elle du même ordre que celle estimée ?

- **Les pics d'utilisation doivent être pris en compte** et il est difficile de les prévoir. Quel sera l'effet, par exemple, de l'annonce d'une promotion sur le site web d'une société de commerce en ligne ? Une fréquentation doublée, triplée ?
- **Les utilisateurs sont imaginatifs :** ils trouveront toujours de nouvelles utilisations à leur système. Nouvelles utilisations qui viendront immédiatement fausser les prévisions initiales.
- **Les organisations sont complexes :** il est impossible de trouver des spécialistes maîtrisant tous les domaines fonctionnels et techniques du champ de l'étude. Il est souvent difficile de réunir tous les interlocuteurs nécessaires.

Alors que le capacity planning peut ressembler pour certain à une sorte de prédiction, il est possible de suivre une démarche quasi-scientifique. Il faut cependant garder à l'esprit le fait que, s'il est facile de prédire qu'un système ne convient pas, il est souvent beaucoup plus difficile de démontrer qu'il conviendra.

Si un élément du système n'a pas la capacité requise, c'est une certitude : le système sera sous-dimensionné. Au contraire, il est plus hasardeux d'affirmer que tous les éléments sont de capacité suffisante et que tous les débits sont satisfaisants en raison de la complexité des systèmes modernes.



Le capacity planning, éviter d'atteindre le point de rupture

Un système est dimensionné en fonction d'un certain nombre de critères débit/capacité...

Comme l'activité des entreprises évolue, il faut s'assurer que le système reste toujours dans les limites pour lesquelles il a été construit, afin de le faire évoluer alors qu'il est encore temps.

Exemples

- L'infrastructure d'un site web est conçue pour fournir un nombre maximal de pages par seconde ; lorsque cette valeur est atteinte, certains utilisateurs risquent de se voir rejetés.
- Un système de type datawarehouse est prévu pour stocker 800 Go de données. Attention en cas de dépassement de cette limite : les disques vont déborder, les sauvegardes ne seront plus possibles...



La création d'un modèle de simulation

● Quelles sont les données de sorties d'un modèle ?

Il s'agit essentiellement de construire « une boîte noire » qui représentera l'ordinateur et qui permettra d'en prévoir le comportement. Les algorithmes qui permettent de paramétrer la « boîte noire » ne sont pas propres à la prévision de charges. Ils peuvent également être utilisés pour modéliser les caisses de supermarchés ou encore les difficultés de circulation automobile. On peut en fait obtenir deux gammes de résultats : une première série donnera une vue orientée vers les utilisateurs. Il s'agit du temps de réponse d'une application, de la durée d'exécution d'un batch ou encore du débit d'un batch exprimé en nombre de transactions (au sens métier du terme) par unité de temps. Au-delà de ces résultats externes, des résultats plus techniques peuvent être parfois utiles : saturation des ressources, par exemple en vue d'un changement de CPU, de disques en vue d'un passage de disques traditionnels vers des baies de stockage, etc.

● Comment représenter un ordinateur ?

Un ordinateur ou un cluster est classiquement représenté sous la forme d'un réseau de centres de services. Chaque centre de service est doté d'une file d'attente dans laquelle sont placées les requêtes en attendant que le centre soit disponible pour les traiter. Chaque CPU, par exemple, est considéré comme un centre de service. Les disques peuvent être également représentés de cette façon si les cas où existent des caches sur des baies ou sur les disques sont exclus. La représentation la plus simple d'un quadriprocesseur sur lequel est attaché quatre disques est donc un réseau de 8 centres de services.

● Quelles sont les données d'entrées ?

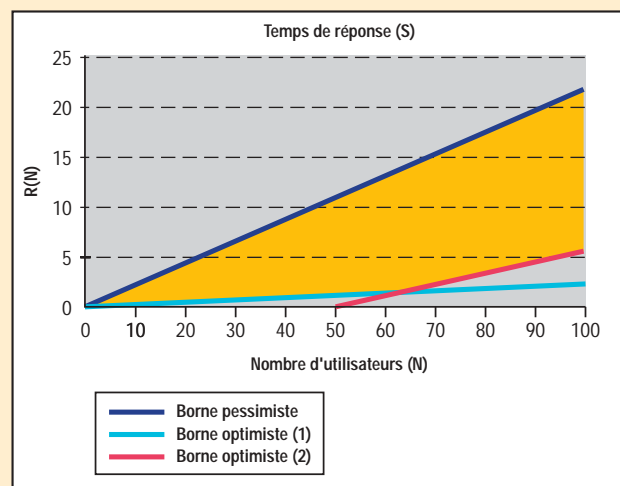
Le modèle va s'appliquer à des transactions identifiées. Chaque transaction est caractérisée par deux types de variables : la première est le nombre de transactions présentes dans le modèle à chaque instant. Ce nombre peut varier considérablement en fonction du moment choisi, en particulier pour les applications transactionnelles. Les paramètres de sorties (le temps de réponse par exemple) sont donc souvent exprimés en fonction du nombre de transactions actives. Sur une machine qui ne change pas de configuration, le deuxième jeu de paramètres est en revanche constant : il s'agit « du temps de séjour » de chaque transaction sur les centres de services. On pourra ainsi dire qu'une transaction T1 nécessite 50ms de CPU, 80ms sur un disque et 30 ms sur le second. Ces valeurs sont indépendantes des autres transactions, car les temps d'attente sur les ressources ne sont pas comptabilisés.

● Processus de modélisation

Le processus de construction du modèle est relativement simple dans son principe. Une première phase consiste à évaluer le comportement du système vis-à-vis des goulots d'étranglement (CPU, disques) de la machine. En pratique, deux précautions doivent être prises. Il ne faut pas que la mémoire soit la cause d'une dégradation de performances dans la mesure où cette dernière n'est pas modélisée. Il faut ensuite soulever les problèmes de performances intrinsèquement liés à une mauvaise configuration du système ou des bases de données. Un taux d'Entrées/Sorties très important peut par exemple saturer un disque à cause d'une mauvaise configuration des redo logs, d'une seule zone de swap, etc. La deuxième étape est celle de la modélisation elle-même. Il s'agit de localiser les transactions qui seront intégrées dans le modèle, d'évaluer les temps de séjour sur les centres de services, etc. La troisième étape permet de valider le modèle. Il s'agit ici de comparer les données issues du modèle avec les données mesurées dans la réalité. La quatrième et dernière étape est celle de la simulation. On pourra faire varier le nombre d'utilisateur de telle ou telle application, ajouter des disques ou encore augmenter la puissance des CPU.

● Encadrements vs simulation fine

Les algorithmes de simulation sont itératifs et trop complexes pour être intégrés tels quels dans un tableau. C'est pour cette raison que les programmes de simulation sont particulièrement utiles (des produits existent sur le marché). En simplifiant à l'extrême le modèle, il est en revanche possible d'encadrer simplement les temps de réponse. On s'appuie pour cela d'une part sur des constatations simples (le centre de service qui occasionnera le goulot d'étranglement principal sera celui pour lequel le débit sera minimal), d'autre part sur les asymptotes des formules utilisées dans les programmes de simulation :





Les différentes étapes de la modélisation

● Comment caractériser l'environnement à modéliser ?

La première tâche à accomplir lors de la réalisation de travaux de prédiction de performances est la caractérisation des systèmes soumis à l'étude. C'est en général la plus complexe, car il s'agit de réduire une machine ou un groupe de machines à un ensemble de caractéristiques censées les représenter. L'étape de validation, effectuée après la modélisation, permettra a posteriori de confirmer ou d'infirmer les hypothèses de construction du modèle. Dans la pratique, il faut généralement distinguer des « charges de travail », qui peuvent être définies comme des unités de travail relativement homogènes. Les attributs donnés à ces charges de travail dépendent de l'objectif de la modélisation : l'efficacité d'un serveur proxy de cache pour un site web sera évalué sur la base des fréquences d'accès aux documents et sur leur taille. S'il s'agit de déterminer l'impact de l'ajout de processeurs sur un ordinateur, les attributs des charges de travail seront le temps CPU et le nombre moyen d'Entrées/Sorties nécessaire à l'exécution des requêtes.

● Comment identifier les charges de travail ?

Une charge de travail se définit avant tout par la présence d'un composant de base quantifiable. Ce composant peut être une unité de travail (saisie d'une commande), de volume (chargement d'un fichier par ftp à partir d'un site web), etc. On utilise fréquemment un découpage à trois niveaux pour déterminer les charges de travail. Le premier niveau est fonctionnel, la charge de travail s'exprime vis-à-vis de l'utilisateur du système en termes de quantités, par exemple un nombre d'employés, de factures par clients, etc. Le second niveau est applicatif, nombre et volumes de requêtes HTTP, d'appel à certaines procédures stockées, etc. Le dernier niveau est vu sous l'angle de la consommation de ressources : nombre d'Entrées/Sorties moyen, consommations CPU et réseau. D'une façon plus générale, chaque charge de travail peut représenter une application dans le cas d'un serveur multi-applications. Une autre approche de la technique de partitionnement consiste à ventiler les charges de travail en fonction de l'usage des ressources :

| Transactions | Fréquence | CPU/Transaction (ms) | Nombre moyen E/S /Transactions |
|--------------|-----------|----------------------|--------------------------------|
| Légères | 20 % | 50 | 1 |
| Moyennes | 50 % | 300 | 5 |
| Lourdes | 30 % | 600 | 55 |

● Constitutions de transactions artificielles

Les charges de travail sont parfois difficiles à cerner. Un serveur de bases de données peut par exemple mutualiser les accès en provenance de plusieurs applications

qui accèdent aux mêmes données. Le temps à passer pour caractériser les charges de travail peut être considérable. Enfin, les dimensionnements des systèmes sont souvent demandés par des services d'infrastructures qui ne disposent pas d'informations suffisamment précises pour caractériser les applications sur le plan fonctionnel. La méthode à appliquer consiste alors à constituer des transactions artificielles qui représentent le système. Le principe de cet assemblage est d'effectuer une collecte du type (nombre de transactions actives, consommation CPU, E/S). On considère donc à chaque mesure que toutes les transactions actives consomment la même quantité de temps sur les ressources. Une collecte de points peut faire visuellement ressortir des agglomérats qui peuvent alors représenter des transactions et leurs consommations. Il est préférable de collecter un nombre important de points (plusieurs milliers, un point de collecte toutes les 2' par exemple sur les heures ouvrées pendant deux semaines) afin de limiter les erreurs (figure 1, un seul disque, un seul CPU). La représentation visuelle s'avère parfois insuffisante. Un algorithme de « clusterisation » de points est alors mis en œuvre afin de déterminer les barycentres des charges de travail (figure 2).

La représentation artificielle a bien sûr le défaut de ne pas offrir la possibilité de faire varier en entrée du modèle le nombre d'utilisateurs d'une transaction particulière, mais elle offre en revanche une meilleure image du système : les seuls cas d'échecs de la validation du modèle sont liés à des erreurs d'éléments collectés, jamais de représentativité de transactions.

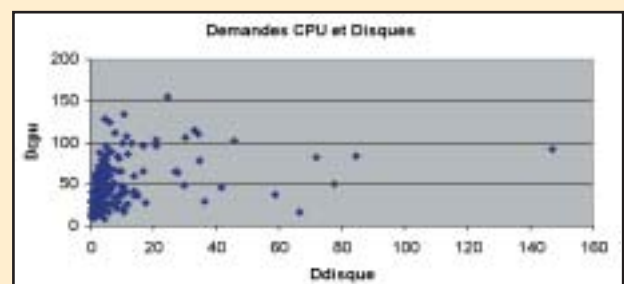


Figure 1: Représentation visuelle des transactions mesurées et de leurs consommations

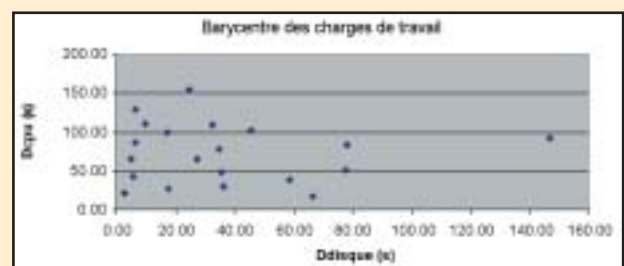
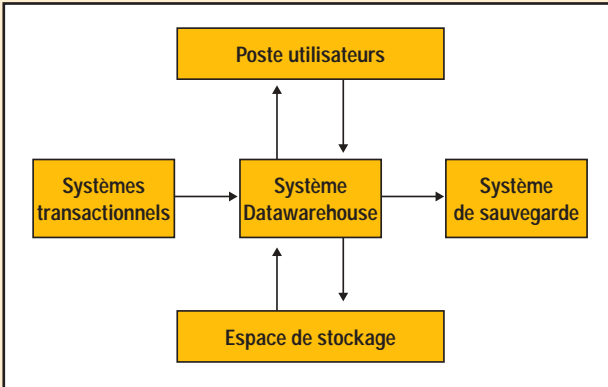


Figure 2: Représentation de 21 transactions dérivées des transactions de la figure 1



Ne rien oublier et prévoir les évolutions

Exemples des points à ne pas oublier dans le cas d'un dimensionnement d'un système de type datawarehouse.



Le système doit pouvoir gérer l'apport quotidien et la consolidation des données provenant de plusieurs systèmes transactionnels :

- le débit réseau doit être suffisant.
- la capacité de l'espace de stockage doit être suffisante.
- la capacité de calcul de consolidation doit être suffisante.

un certain nombre d'utilisateurs vont faire des requêtes sur ce système

- la capacité de calcul (CPU) doit être adaptée.
- le débit des disques doit être suffisant.
- le débit du réseau doit être suffisant.

dans le cas d'une sauvegarde délocalisée

- le débit du réseau doit être suffisant.
- la capacité de stockage du système de sauvegarde doit être suffisante.

Bien sûr, tous ces paramètres sont liés entre eux et doivent être étudiés dans la durée, que se passe-t-il par exemple en cas d'augmentation significative de l'activité transactionnelle ?

Quelques erreurs d'appréciations sont fréquentes.

- Une activité stable implique une augmentation linéaire de la capacité de stockage.
- Une activité qui croit de 50 % par an implique une augmentation de 50 % par an de la puissance CPU nécessaire et de 50 % par an de

l'espace disque : soit au bout de n années

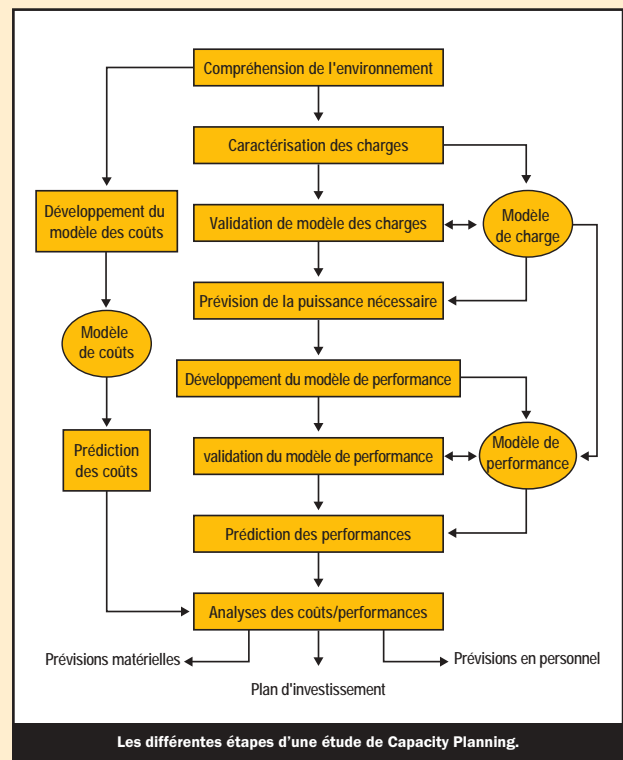
Puissance = puissance x (1+0.50)ⁿ

Capacité = capacité x ((1+0.50) + (1 + 0.50)² + ... + (1 + 0.50)ⁿ) ... Soit capacité x 7.125 au bout de 3 ans

Ne pas négliger des points qui peuvent rapidement être bloquants

- Le débit du réseau en cas de sauvegarde centralisée: il faut pour pouvoir transférer les données disposer d'une bande passante suffisante sinon une alternative doit impérativement être trouvée.

Enfin, d'une manière générale, les lois qui gouvernent la charge des différents éléments des systèmes ne sont pas linéaires. Deux processeurs n'autorisent pas une puissance deux fois plus importante qu'un seul...



Les différentes étapes d'une étude de Capacity Planning.

Il est donc nécessaire de mettre chaque système sous surveillance pour éviter d'atteindre un point de rupture. De fait chaque sous-système doit être contrôlé. Ainsi, si le datawarehouse atteint son point de rupture, c'est que les systèmes amont fournissent plus de données que prévues. Il est donc possible de détecter cette évolution à la source. Il existe de nombreux outils destinés à recueillir des

métriques indiquant l'activité d'un système. La majorité des systèmes en sont munis en standard (perfmon pour MS Windows, sar, vmstat, iostat pour UNIX). Il faut absolument les utiliser et c'est une des tâches essentielles des administrateurs systèmes et DBAs. Pour les aider dans leurs tâches, ils peuvent disposer de consoles logicielles permettant de suivre avec précision leurs environnements.



Il faut en général joindre à ces collectes systèmes des informations en provenance des bases de données (nombre de transactions actives par seconde en particulier).



Les limites des études de capacity planning

Il existe forcément des limites aux études de capacity planning.

● Des limites budgétaires

On l'a vu, l'élément déclencheur d'une telle étude est souvent lié à un problème financier. Il faut donc que le ratio « coût des études/coût du système » reste intéressant. En raison des performances toujours accrues des infrastructures techniques, il est souvent possible de réduire les études au strict minimum.

Il ne faut pas que l'étude qui servira à départager deux systèmes coûte plus cher que la différence de prix entre les deux... surtout si l'étude indique qu'il faut acquérir le plus cher.

● Les limites de la prédiction

Il est souvent complexe de modéliser un système à partir d'informations fiables. Il est presque impossible de prévoir l'évolution d'un système dans le temps alors que ses utilisateurs même n'ont qu'une vague idée de l'évolution de leur activité.

Il faut donc souvent envisager le pire lors des études.

Conclusion

Le développement ou l'acquisition d'un logiciel perfectionné ne doit pas faire oublier l'infrastructure qui devra le supporter.

Le capacity planning offre dans ce sens une réponse concrète au besoin des directions informatiques de construire des systèmes à leur mesure. L'aspect théorique et rébarbatif de certaines phases des études de capacity planning ne doit pas faire oublier qu'il s'agit d'une étape incontournable dans le processus de création d'un système. Même dans le cas de configurations simples, il est nécessaire de passer en revue les points qui pourront être bloquants dans le futur.

Enfin, si disposer d'un système bien dimensionné est un préalable indispensable, il ne faut pas oublier qu'il est tout aussi primordial d'exercer un contrôle continu sur son évolution pour éviter d'atteindre son point de rupture un peu plus tôt que prévu... et d'en subir alors les conséquences désagréables. ■

Pour en savoir PLUS :

PRODUITS

Outils de stress

winrunner (Mercury Interactive)

<http://www-svca.mercuryinteractive.com>

Outils de modélisation

Best/1 (BMC Software) : <http://www.bmc.com>

CONSULTING

Storage Foundation <http://www.storage-foundation.com>

LIVRES DE RÉFÉRENCE

« **Capacity Planning for Web Performance** » et « **Scaling the e-business** » par Daniel A. Menascé et Virgilio A.F. Almeida. Prentice Hall 1998 et 2000

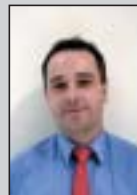
« **Quantitative System Performance: Computer System Analysis Using Queueing Network Models** » par Edward D. Lazowska, John Zahorjan, G. Scott Graham, Kenneth C. Sevcik. Publié en 1984, ce livre reste un ouvrage de référence, bien que les exemples ne soient plus très pertinents. Il est toujours disponible à la vente. Le copyright ayant été levé, le livre est disponible sur le web.

(<http://www.cs.washington.edu/homes/lazowska/qsp>)



Laurent PHILISPART

INSA Rouen
Consultant Sénior,
Expert Base de Données,
Architectures Distribuées et
Méthodes.



Christian BILLEN

Ingénieur de l'Ecole Supérieure d'Electronique de l'Ouest et diplômé de l'IEFSI-EDHEC, Christian Bilien a 13 ans d'expérience en ingénierie des systèmes et des bases de données, acquise chez les constructeurs et dans des grands comptes. Il s'est en particulier spécialisé dans les performances des systèmes et des bases de données. Il est actuellement directeur des opérations de Storage Foundation, une société spécialisée dans le conseil en infrastructures de stockage et de bases de données.

- cbilien@storage-foundation.com

- www.storage-foundation.com (en construction au 20 mars)