

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

PARIS

Examen probatoire
Spécialité informatique

Session de janvier 2004

Evaluation des performances des
systèmes dans le domaine des
entrées-sorties.

par Omar BENGUEAH

[\(\[omar@ziliz.com\]\(mailto:omar@ziliz.com\)\)](mailto:omar@ziliz.com)

JURY

PRESIDENT : Professeur Bernard LEMAIRE

Table des matières

INTRODUCTION.....	5
CHAPITRE 1. NECESSITE D’EVALUER LA PERFORMANCE DES SYSTEMES	6
1.1. PERFORMANCES DES APPLICATIONS : UN ENJEU DE TAILLE.....	6
1.2. UNE APPLICATION PERFORMANTE NECESSITE UN SYSTEME ADAPTE	6
1.3. QU’EST-CE QUE L’EVALUATION DES PERFORMANCES SYSTEMES ?	7
1.4. OBJECTIFS DE L’EVALUATION	7
1.5. TECHNIQUES D’EVALUATION	9
CHAPITRE 2. LES ENTREES-SORTIES AU CŒUR DU SYSTEME.	10
2.1. BREF RAPPEL D’ARCHITECTURE DES SYSTEMES.....	10
2.1.1. <i>Place des E/S dans l’architecture du système</i>	10
2.1.2. <i>Structure et fonctionnement d’une E/S</i>	11
2.2. I/O, MAILLON FAIBLE DE LA PERFORMANCES DES SYSTEMES.	12
2.2.1. <i>Evolution technologies : Microprocesseurs vs. E/S</i>	12
2.2.2. <i>Cheminement d’une lecture d’information dans le système</i>	13
2.2.3. <i>Techniques pour augmenter les performances</i>	14
2.3. NOUVELLES ARCHITECTURES D’E/S ET EVALUATION DES PERFORMANCES	15
2.3.1. <i>Tendances</i>	15
2.3.2. <i>RAID</i>	15
2.3.3. <i>Systèmes parallèles : SAN, NAS</i>	16
CHAPITRE 3. MESURE DE LA PERFORMANCE	18
3.1. METRIQUES.....	18
3.1.1. <i>Le débit du système</i>	18
3.1.2. <i>Le temps de réponse</i>	18
3.2. METHODOLOGIE DE MESURE.....	19
3.3. MONITEUR DE MESURE.....	19
3.3.1. <i>Moniteurs matériels</i>	19
3.3.2. <i>Moniteurs logiciels</i>	20
3.4. INTERETS ET LIMITES.....	21

CHAPITRE 4. LES ETALONS DE PERFORMANCE OU BENCHMARKS	22
4.1. VUE D'ENSEMBLE	22
4.1.1. <i>Définition et objectifs</i>	22
4.1.2. <i>Architecture d'un benchmark</i>	22
4.1.3. <i>Positionnement des benchmarks</i>	22
4.2. LES BENCHMARKS SUR LE MARCHÉ	24
4.2.1. <i>Survol des benchmarks disponibles</i>	24
4.2.2. <i>Zoom : SPC-1</i>	26
4.3. CRITIQUE DES BENCHMARKS	28
CHAPITRE 5. LA CHARGE DE TRAVAIL	29
5.1. IMPORTANCE DE LA CARACTERISATION DE LA CHARGE DE TRAVAIL	29
5.2. APPLICATIONS ET UTILISATION E/S : DES PROFILS VARIES	29
5.3. CARACTERISTIQUES DES CHARGES DE TRAVAIL	30
5.3.1. <i>Choix des paramètres</i>	30
5.3.2. <i>Paramètres relatifs aux E/S</i>	30
5.4. CARACTERISATION : METHODOLOGIE	31
5.4.1. <i>Collecte de données</i>	31
5.4.2. <i>Traitement des données statistiques</i>	33
5.5. ANALYSE.....	35
CHAPITRE 6. MODELISATION ET SIMULATION	36
6.1. DEFINITION ET OBJECTIFS.....	36
6.2. MODELISATION	36
6.2.1. <i>La charge de travail</i>	36
6.2.2. <i>Le système de fichiers</i>	37
6.2.3. <i>Le sous-système physique de stockage (disque)</i>	37
6.3. ZOOM : DISKSIM.....	39
6.3.1. <i>Présentation</i>	39
6.3.2. <i>Fonctionnement</i>	40
6.4. INTERETS ET LIMITES.....	41
CONCLUSION	42
GLOSSAIRE	43
BIBLIOGRAPHIE.....	44
TABLE DES ILLUSTRATIONS	46

Introduction

Alors que les performances des processeurs doublent tous les 18 mois, comme l'avait envisagée la fameuse prédiction de Moore, celles des systèmes de stockage progressent péniblement d'une dizaine de pour cent par an, creusant un fossé de plus en plus grand, et tirant les performances globales des systèmes vers le bas. L'émergence de nouvelles architectures d'entrées-sorties n'est pas anodine. Elles sont le seul support qui permettent un accès rapide à de grandes quantités de données indispensables au fonctionnement de nombreuses applications.

Partant donc du constat qu'un système et une application performante ne peuvent se concevoir sans des entrées-sorties adaptées et que celles-ci ont des ressources limitées, l'évaluation des performances devient alors le moyen unique pour optimiser les ressources disponibles voire de tenter de combler le fossé.

Ce travail se propose de présenter les différentes techniques d'évaluation des performances des entrées-sorties, leurs enjeux, leurs fonctionnements ainsi que leurs utilisations. Pour se faire, seront tout d'abord explicités les termes évaluation et performance et leur implication au travers de leurs objectifs et acteurs, pour ensuite proposer une explication du fonctionnement interne des entrées-sorties et de leurs évolutions. Le cœur du document présente les différentes techniques d'évaluation de façon progressive : la mesure de la performance, les benchmarks, les charges de travail et leur caractérisation. Enfin, la simulation, ses enjeux et les différents outils mis en œuvre. Chaque partie sera illustrée par la présentation d'un outil représentatif.

Chapitre 1.Nécessité d'évaluer la performance des systèmes

1.1. Performances des applications : un enjeu de taille

Le rôle essentiel de toute application informatique est d'accomplir les fonctions pour laquelle elle a été conçue, et d'offrir une performance adaptée à un coût raisonnable.

La performance des applications et des services est un facteur clé de leurs succès : un site Internet lent fera fuir ses visiteurs, un hébergeur proposant des services non performants verra ses clients partir vers la concurrence. Pire, pour une application stratégique, la lenteur peut ruiner la productivité de tout un service ou entreprise, et engendrer des coûts supplémentaires substantiels.

Une application non performante est également source de nombreux problèmes relatifs à la disponibilité et la maintenance.

1.2. Une application performante nécessite un système adapté

La performance des applications dépend de multiples facteurs, nombreux sont relatifs à une bonne conception de celles-ci. Le courant actuel et prépondérant dans le génie logiciel est une séparation complète de la partie logique et physique : une application conçue suivant les règles de l'art est une application dont les fonctionnalités ont été correctement abstraites, qui ne dépendent pas des possibilités offertes par les systèmes et dont les détails d'implémentation sont complètement masqués, cette démarche nécessite donc une approche descendante : de l'analyse des besoins vers l'implémentation physique.

Cette approche, bien qu'avantageuse à plusieurs points de vue pose certains problèmes de compréhension, en effet elle sous-entend aux yeux des concepteurs des ressources systèmes illimitées, instantanées et presque gratuites. La réalité est quelque peu différente, non seulement les ressources systèmes sont limitées par le coût maximum fixé pour le développement et l'exploitation, mais les avancées flagrantes des performances des systèmes sont à tempérer (voir 2.2).

Il en résulte une détection tardive des problèmes de performances et un traitement à chaud de ceux-ci, ce qui s'apparente dans la pratique à des interventions d'urgence visant à sauver les applications de la noyade via divers rustines, et qui finissent par déstructure l'application rendre sa fameuse conception modulaire toute relative.

La solution à ce problème revient à une prise en compte des enjeux et des aspects de la performance et ceci dès les premières phases de la conception des applications, et ceci au travers des différentes techniques d'évaluation des performances des systèmes étudiés par la suite.

1.3. Qu'est-ce que l'évaluation des performances systèmes ?

L'évaluation des performances des systèmes dans le domaine des entrées/sorties consiste à « Déterminer approximativement la qualité des résultats optimaux obtenus par les entrées/sorties des systèmes dans le cadre de l'accomplissement de la tâche qui leur a été assignée »

En d'autres termes, elle consiste à mesurer le degré de correspondance entre le résultat offert par les entrées-sorties et les besoins qu'elles sont censées satisfaire.

Ces besoins fixés dépendent des acteurs :

Concepteur et vendeurs d'entrées/sorties : son rôle est de concevoir un système capable de répondre à des besoins les plus larges possibles et donc maximiser et ou mettre en avant tous les indicateurs cruciaux pour l'utilisateur.

Gestionnaires / administrateurs des systèmes : Ils ont pour mission de veiller à rendre un service satisfaisant aux usagers qui ont généralement un besoin spécifique et ceci à un coût raisonnable.

Analystes et programmeurs (utilisateurs) : ils ont une vision centrée sur l'application et doivent la concevoir au mieux en comprenant au mieux le fonctionnement du système et les réactions de leurs applications sur celui-ci.

1.4. Objectifs de l'évaluation

Les objectifs de l'évaluation peuvent s'articuler autour de trois axes et dépendent des acteurs :

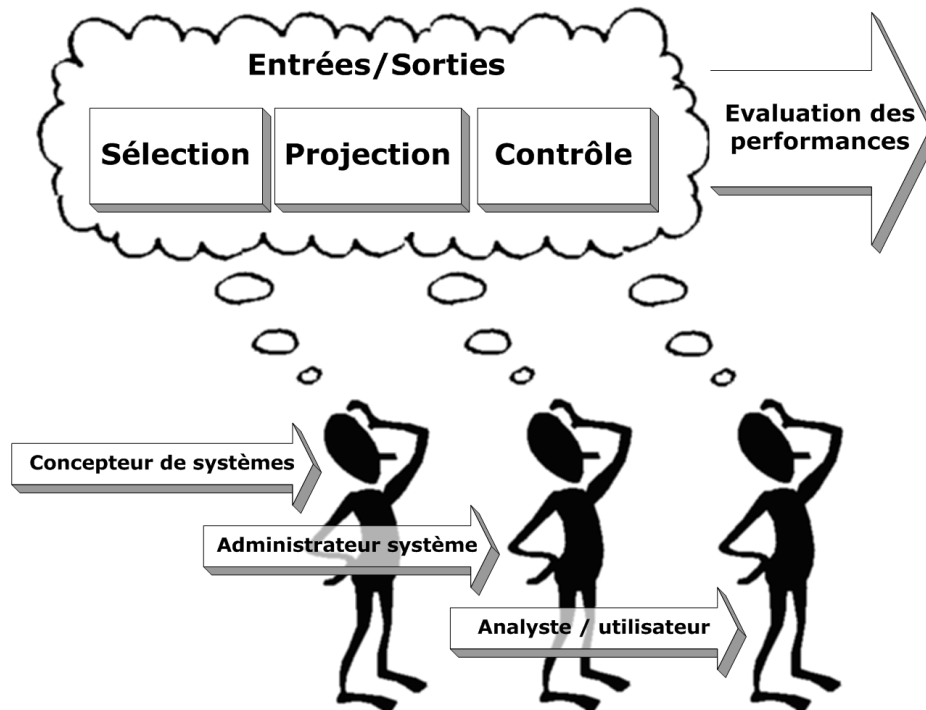


Figure 1 : Objectifs et acteurs de l'évaluation des performances

La sélection

- Elle permet de choisir une architecture adaptée
 - Administrateur : Sélection d'une architecture RAID / NAS
 - Utilisateur : adapter les ressources à utiliser par l'application conçue

- Comparer des solutions matériels existantes en vue d'en choisir une ou dans un objectif marketing
 - Administrateur :Sélection de différentes marques de disques durs
 - Concepteur : effectuer des mesures comparatives des différentes solutions sur le marché via des benchmarks adaptés

La prévision et projection

- Dimensionner (quantité et taille) les différents éléments du système d'entrées/sorties.
- Utilisateur : fixer les possibilités / marges de manœuvres réalisables par sa future application.
- Concepteur : simuler les résultats d'un système futur.
- Administrateur : Prévoir les performances des charges de travail futures (estimations)

Le contrôle

- Utilisateur : valider les performances attendues par son application.
- Détecter et éliminer les problèmes de performances en localisant les goulots d'étranglement :
 - Utilisateur / administrateur : en établissant le profil des charges de travail sollicitant le système
- Ajuster les paramètres systèmes afin d'obtenir un résultat optimal
 - Administrateur : contrôle de taille idéale d'un cluster

1.5. Techniques d'évaluation

Les techniques d'évaluation existantes, qui seront détaillées tout au long du rapport sont :

- Intuition
- Mesures
- Benchmarks
- Simulation

Le choix d'une des techniques d'évaluation ci-dessous dépendra des besoins souhaités ainsi que des moyens disponibles :

	Intuition	Mesures	Benchmarks	Simulation
Coût de réalisation	++	-	+	---
Faisabilité de la démarche	+	-	+	--
Fiabilité des résultats	-	++	+	+
Existence des systèmes	+	--	--	++
Comparaison entre systèmes	-	+	++	+
Adéquation avec l'usage réel	+	++	--	+

Chapitre 2. Les entrées-sorties au cœur du système.

La première technique d'évaluation des performances du sous-système de stockage est l'intuition, ou plutôt l'expertise. Elle repose sur une compréhension du fonctionnement des systèmes de stockage et de leur place dans le système.

La compréhension des mécanismes des entrées-sorties est également utile pour deux aspects : elle permet, d'une part, d'identifier les goulots d'étranglement applicatifs et donc de faire évoluer les entrées-sorties et/ou la conception des applications (voir 5.2), et est, d'autre part, également indispensable pour la construction d'un modèle de simulation réaliste (voir 6.2)

2.1. Bref rappel d'architecture des systèmes

2.1.1. Place des E/S dans l'architecture du système

Les entrées/sorties ou systèmes de stockage¹ permettent de conserver les données de travail de manière permanente. Elles offrent par rapport à la mémoire RAM – en plus de la non volatilité – une capacité de stockage beaucoup plus importante à un coût moindre (plusieurs centaines de Go). L'espace mémoire des disques durs est également utilisé en extension à la mémoire principale lors de l'exécution des programmes, il est appelé mémoire virtuelle.

La communication avec les autres périphériques du système se fait via un ensemble de liaisons physiques appelés bus. Il existe plusieurs bus de nature et de vitesse différente : le bus système pour le processeur et la mémoire vive, et les bus d'extension tel que le bus *PCI* (appelés également bus d'entrée-sortie).

¹ Le terme E/S regroupe un ensemble de composants beaucoup plus large, tel que les claviers, écran, scanners... nous nous limiterons ici aux systèmes de stockage sur support magnétique, comme cadré dans la bibliographie du sujet.

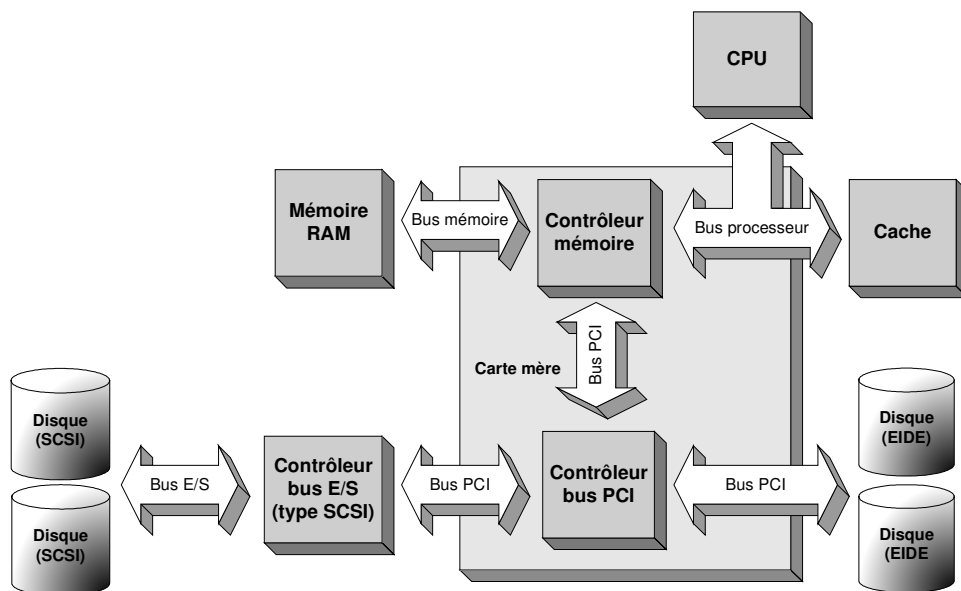


Figure 2 : communication entre composants et E/S

2.1.2. Structure et fonctionnement d'une E/S

Chaque surface de disque est divisée en cercles concentriques, appelés *pistes*. Chaque piste est elle-même divisée en *secteurs* qui contiennent l'information : chaque piste peut avoir 32 secteurs. Le secteur est la plus petite unité qui peut être accédée en lecture ou écriture. La séquence enregistrée sur le matériau magnétique est le numéro de secteur, un intervalle, l'information pour ce secteur y compris le code de correction d'erreurs, un intervalle, le numéro du secteur suivant et ainsi de suite.

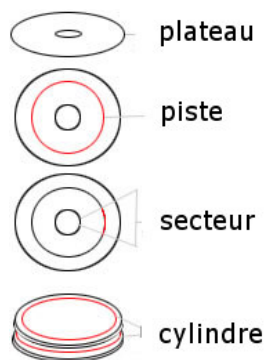


Figure 3 : Structure logique des plateaux

Pour lire ou écrire l'information dans un secteur, un *bras* mobile contenant une tête de lecture/écriture est situé au-dessus de chaque surface. Les bits sont enregistrés avec un code de longueur limitée, qui améliore la densité d'enregistrement du support magnétique. Les bras de chaque surface sont solidaires et se déplacent ensemble, de telle sorte que

chaque bras est au-dessus de la même piste de toutes les surfaces. Le terme *cylindre* est utilisé pour appeler toutes les pistes sous les bras en un point donné de toutes les surfaces.

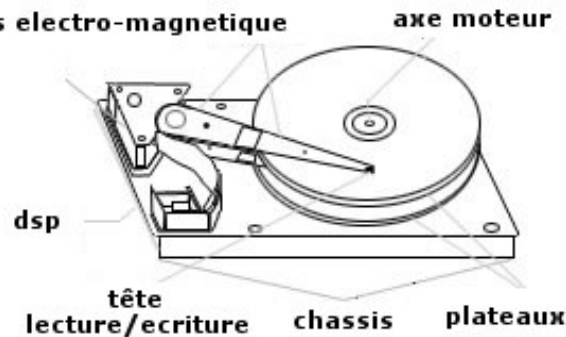


Figure 4 : Structure physique du disque

Pour lire ou écrire un secteur, le contrôleur de disque envoie une commande pour déplacer le bras sur la bonne piste. Cette opération est appelée une recherche, et le temps pour déplacer le bras sur la piste désirée est appelé *temps de recherche*. Le temps de recherche moyen pour un disque est d'environ 4 ms à 15 ms

Le temps pour le secteur désiré de se déplacer jusqu'à la tête est appelé le *délai de rotation*. La plupart des disques tournent à 7200 tours à 15000 tours par minute, et un demi-tour du disque est le délai moyen jusqu'à l'information désirée : le délai de rotation moyen pour la plupart des disques est donc :

$$(0,5 / 7200) * 60 = 0,0042 = 4,2 \text{ ms}$$

La composante suivante d'un accès disque, le *temps de transfert*, est le temps pour transférer un bloc de bits, typiquement un secteur, lors du passage sous la tête de lecture/écriture. C'est en fonction de la taille du bloc, de la vitesse de rotation, de la densité d'enregistrement d'une piste et de la vitesse de l'électronique interconnectant le disque à l'ordinateur. Les débits de transfert en 2004 sont typiquement de 5 à 30 Mo par seconde.

2.2. I/O, maillon faible de la performances des systèmes.

2.2.1. Evolution technologies : Microprocesseurs vs. E/S

Ces dernières décennies, les performances des microprocesseurs ont augmenté de plus de 50% par an, alors que les performances des disques durs n'ont cru que de 5% à 15%, creusant un écart de performances de plus en plus significatif.

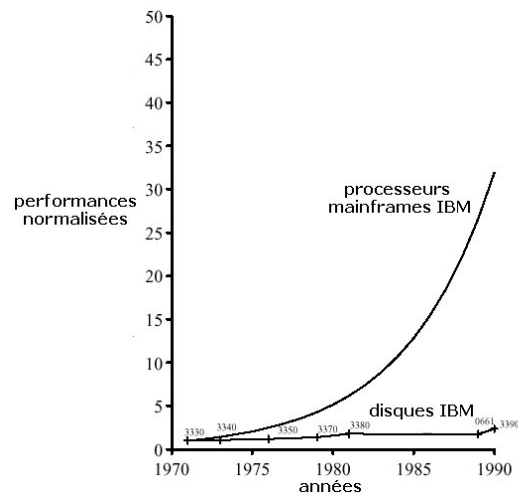


Figure 5 : Evolution des performances des processeurs et des entrées-sorties au cours des dernières décennies

2.2.2. Cheminement d'une lecture d'information dans le système

La performance d'un système est limitée par la partie la plus lente du chemin entre le processeur et les composants d'E/S (voir Figure 2) :

1. Le processeur demande au contrôleur disque l'accès à un emplacement précis du disque dur (via le bus PCI).
2. Le contrôleur acquitte réception de la demande auprès du processeur et transmet la requête au disque dur en même temps.
3. Le processeur se met en veille, ou va faire autre chose en attendant la suite.
4. Le disque dur pendant ce temps commence à déplacer ses têtes de lecture à l'emplacement où se trouve l'information.
5. Une fois les têtes de lecture en place, l'information est lue et stockée dans la mémoire cache du disque dur.
6. Le disque transmet ensuite l'information au contrôleur.
7. Le contrôleur copie l'information en mémoire puis notifie le processeur que l'information est arrivée.
8. Le processeur revient à son travail et continue de traiter les données.

Pour bien comprendre l'importance de chaque étape, il faut garder en mémoire que les temps d'accès et de lecture d'un disque dur se comptent en millisecondes (ce sont des opérations utilisant de la mécanique), alors que le reste de l'électronique travaille plutôt à des vitesses de l'ordre de la *nanoseconde*. Par conséquent, les étapes 1. et 2. sont presque instantanées, d'autant qu'elles ne nécessitent que l'échange de quelques octets.

Les étapes 4. 5. 6. et 7. sont en revanche relativement lentes puisqu'elles se comptent en... *millisecondes*! On peut cependant, dans une certaine mesure, optimiser les étapes 6. et 7. en intervenant d'une part sur l'interface et le contrôleur disque, au niveau de l'étape 6., et d'autre part sur le bus PCI et la vitesse de la mémoire vive pour l'étape 7.

2.2.3. Techniques pour augmenter les performances

Les systèmes de stockages possèdent une mémoire tampon d'une taille variable (jusqu'à plusieurs Mégaoctets) permettant notamment de stocker les données temporaires en attente de lecture ou d'écriture. Cette mémoire, infiniment plus rapide que le disque magnétique permet des gains de performance conséquents. Les techniques qui suivent, reposent sur une gestion complexe de la mémoire tampon du disque dur.

2.2.3.1. Ordonnement

La stratégie la plus simple pour gérer la file d'attente d'une unité de disque consiste à traiter les demandes dans l'ordre d'arrivée (*FIFO*). Bien qu'il s'agisse d'une stratégie équitable, les performances sont parfois très médiocres : il suffit que les requêtes qui se suivent demandent alternativement des accès à des parties du disque opposés.

Les disques durs incorporent donc des algorithmes permettant d'optimiser les déplacements des têtes : ordonnancement suivant le plus court temps de recherche, ou l'ordonnement par balayage.

2.2.3.2. Préchargement / préfetching

Le préchargement ou *prefetching* consiste à anticiper les opérations que les entrées-sorties devront effectuer et ceci, pendant les moments où elles ne sont pas sollicitées. Il existe deux approches : une méthode manuelle (*informed prefetching*) où le programmeur de l'application au moyen d'un jeu d'instructions informe le sous-système de stockage des informations sur le disque dont il aura besoin, et une méthode automatique (*automatic prefetching*) de plus en plus utilisée qui consiste au moyen d'algorithmes divers (notamment grâce aux chaînes de Markov) de prédire les données dont l'utilisateur aura besoin au travers de l'analyse des requêtes déjà soumises.

2.2.3.3. Conséquences

Ces techniques qui vont au-delà du simple déplacement de bras sur la surface du disque, ont un impact sur les performances qui n'est pas aisément identifiable et dépend énormément de la façon dont ils ont été mises en place. L'évaluation des performances tient donc ici un rôle fondamental.

2.3. Nouvelles architectures d'E/S et évaluation des performances

La variété des architectures d'entrées-sorties existantes et l'émergence de nouvelles architectures orientées réseaux rendent l'évaluation des performances indispensable pour la compréhension et la sélection du système le plus adapté.

2.3.1. Tendances

Le développement du commerce électronique, des *datawarehouses* ainsi que l'augmentation du nombre d'applications au sein des entreprises, notamment celles liées à la gestion de la relation client, sont autant de facteurs qui contribuent à l'accroissement exponentiel du volume des données.

Pour répondre aux besoins de stockage considérables qui en découlent, des solutions apparaissent à un rythme soutenu, dans un secteur, celui du stockage, en pleine explosion. Nombre d'acteurs de l'industrie informatique, jusque là présents sur d'autres segments, abordent ce marché fort en perspectives de profits importants à moyen et long terme.

A l'heure actuelle, plus de 95% de tous les systèmes de stockage informatiques, tels que les disques durs, les systèmes RAID, etc., sont directement reliés à des ordinateurs clients à travers divers adaptateurs *SCSI*, *Fibre Channel*, ou autres. Ce type de stockage, ou attachement direct, est généralement appelé DAS (Direct Attached Storage).

2.3.2. RAID

Le principe du RAID consiste à combiner plusieurs lecteurs de disque de petite taille et bon marché en un réseau permettant des performances supérieures à celles d'un lecteur de grande taille et coûteux. L'ordinateur "voit" ce réseau de lecteurs comme une unité de stockage logique ou un lecteur unique.

RAID est une méthode où les informations sont réparties sur plusieurs disques, à l'aide de techniques telles que l'agrégat par bandes (RAID 0) et la mise en miroir (RAID 1) afin d'obtenir une redondance, une moindre latence et/ou une bande passante plus importante pour la lecture et l'écriture sur les disques, et de maximiser la faculté de récupération après des pannes de disque dur.

Pourquoi utiliser un RAID ?

- vitesse accrue
- capacité de stockage accrue en utilisant un unique disque virtuel
- efficacité accrue pour la récupération après une défaillance de disque dur

Le RAID 0 correspond à la technique du stripping (répartition des données sur plusieurs disques). Cette technique permet des gains de vitesse en lecture et en écriture. La vitesse augmente avec le nombre de disques mais le risque de panne augmente également puisque les données sont réparties sur tous les disques.

Le RAID 1+0 (ou RAID 10) est la façon la plus performante d'organiser les données. C'est également la solution qui offre la fiabilité la plus grande et qui pénalise le moins le système en cas de panne de l'un des disques. La grappe est organisée par paires de disques miroirs (RAID 1). Les données sont réparties entre ces paires de disques suivant le principe du RAID 0. La capacité utile est égale à la moitié de la capacité physique des disques. C'est donc la solution la plus coûteuse.

Le RAID 5 permet d'optimiser la capacité utile de stockage, qui est égale à celle de la grappe moins un disque. Les données et les informations de parité sont stockées sur tous les disques. Un minimum de trois disques est nécessaire pour constituer un RAID 5. La vitesse de lecture sur un RAID 5 est, comme pour le RAID 1, proportionnelle au nombre de disques utilisés. En revanche la vitesse d'écriture est pénalisée. Elle sera seulement les 3/5 de celle du RAID 1.

Les autres niveaux de RAID 2,3,4 correspondent à d'autres organisations des informations de parité. Ils sont très peu utilisés car ils offrent peu ou pas d'avantages par rapport aux RAID 0,1,5.

2.3.3. Systèmes parallèles : SAN, NAS

La tendance à la consolidation des données ainsi que les besoins croissants d'un accès plus rapide à ces dernières dans les réseaux d'entreprises et pour les applications scientifiques a conduit au développement de deux architectures de stockage : les serveurs de stockage en réseau, ou NAS (Network Attached Storage), et le SAN (Storage Area Network).

Le NAS autant que le SAN sont des solutions de stockage qui ont été optimisées pour permettre une consolidation centralisée du stockage des données ainsi qu'un accès rapide aux fichiers.

Le SAN constitue un réseau de stockage fiable et dédié, basé sur le Fibre Channel, qui offre une grande flexibilité tant en distance qu'en connectivité. Les SAN sont bien adaptés aux applications demandant un stockage dédié, comme les bases de données, qui demandent une mise à jour permanente, et le traitement des transactions en ligne (OLTP).

Les serveurs de stockage en réseau NAS, pour leur part, sont connectés directement à une infrastructure réseau existante et permettent le partage d'un même fichier entre de multiples serveurs et clients dans un environnement hétérogène. Les NAS sont adaptés aux applications qui font appel au service de fichiers comme la CAO, le développement logiciel, l'hébergement web ou le mail.

	NAS	SAN
Transfert des données	A travers un LAN ou un WAN	A travers le SAN vers un serveur vers un LAN ou un WAN
Disponibilité	Des alimentations et des ventilateurs redondants sont couramment utilisés	Des composants matériels et logiciels redondants donnent au système une haute disponibilité. Le système peut être configuré sans le moindre point de panne
Scalabilité	Plusieurs serveurs NAS peuvent être ajoutés au réseau, et du stockage peut être ajouté aux serveurs NAS intermédiaires	Des composants matériels et logiciels redondants donnent au système une haute disponibilité. Le système peut être configuré sans le moindre point de panne
Applications bien adaptées	Ideal pour servir les fichiers	Ideal pour les bases de données et le traitement des transactions en ligne
Fonction principale	Serveur spécialisé, qui sert les fichiers et les données stockées aux postes clients et aux autres serveurs à travers le réseau	Le stockage est accessible à travers un réseau qui lui est spécialement dédié. Sa principale fonction est de fournir aux serveurs un stockage consolidé basé sur le Fibre Channel
Ressources de stockage et de sauvegarde	Les sauvegardes peuvent être attachées directement à des appliances NAS intermédiaires ou être distribuées et attachées à un LAN ou un WAN	Les ressources de stockage et de sauvegardes peuvent être attachées directement au serveur ou à travers une structure Fibre Channel

Figure 6 : Comparatif des architectures NAS et SAN

Chapitre 3. Mesure de la performance

« On ne gère bien que ce que l'on peut mesurer »

3.1. Métriques

Quel que soit les utilisateurs des E/S , il existe des métriques communes qui permettent de juger un certain degré de performance :

3.1.1. Le débit du système

Il s'agit du nombre de requêtes exécutées par unité de temps, il est exprimé en mégabits par seconde (nombre de requêtes multiplié par la taille de celles-ci). Il est parfois appelé « bande passante ». Plus la valeur est importante, meilleure est l'E/S.

3.1.2. Le temps de réponse

Il correspond au temps nécessaire pour que le système traite une requête (durée d'attente en cache + durée d'exécution par le disque, voir 2.1.2) appelé également « latence », un temps de réponse plus court est meilleur.

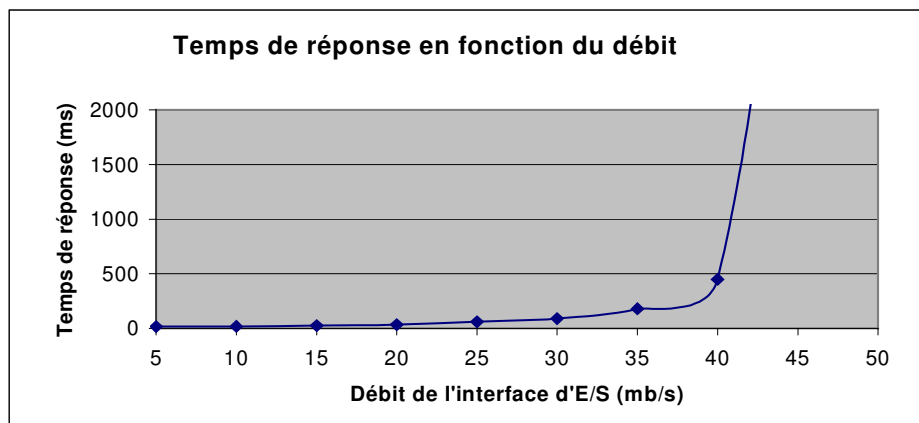


Figure 7 : Evolution du temps de réponse par rapport au débit

De façon mécanique, le débit d'un système croît au fur et à mesure que le nombre de requêtes qui lui est soumis augmente ; cependant, les phénomènes de saturation apparaissent dès que la capacité maximum de traitement de l'entrée-sortie est atteinte. Les nouvelles requêtes arrivantes ne pouvant qu'attendre que le système soit disponible, l'engorgement de l'interface d'entrées-sorties conduit inévitablement à une augmentation brutale du temps de réponse.

Le coût peut également être considéré comme une métrique. En effet, les systèmes d'entrées-sorties sont généralement la partie la plus coûteuse d'un système. On prend alors en compte un rapport coût/performance , tel que le coût total d'acquisition du système ou le coût /débit.

3.2. Méthodologie de mesure

La mesure de la performance a pour but de faire ressortir plusieurs indicateurs significatifs. Le premier concerne la capacité maximum de traitement du sous-système de stockage ou seuil de saturation. Il s'agit du nombre de requêtes maximum que l'entrée-sortie peut traiter avant qu'elle ne s'effondre.

Cet indicateur permettra d'établir un second indicateur qui sera le niveau maximum acceptable de charge, niveau au-delà duquel le temps de réponse deviendra inacceptable pour une qualité de service optimale, qui pourra par exemple être de 75% le seuil de saturation.

A partir de ces premiers indicateurs, il est possible d'obtenir le suivant : le temps de réponse minimum, il s'agit du temps de réponse lors d'une sollicitation moindre de l'entrées-sorties, par exemple à 10% de la charge maximum.

3.3. Moniteur de mesure

Un moniteur est un outil qui permet d'observer l'activité du système. Généralement un moniteur observe les performances d'un système, collecte les statistiques de travail, analyse les données et renvoi les résultats. Les moniteurs peuvent également avoir des fonctions plus évoluées comme l'identification de problèmes classiques et leur solutions adéquates.

3.3.1. Moniteurs matériels

Ce sont des instruments spéciaux ou des ordinateurs spécialisés qui sont utilisés pour prendre des mesures. Les moniteurs matériels ont l'avantage de consommer peu de

ressources et permettent d'obtenir des données plus précises, mais sont généralement très coûteux et rarement disponibles pour l'ensemble des systèmes d'E/S possibles.

3.3.2. Moniteurs logiciels

Les moniteurs logiciels sont généralement implémentés au niveau du système de fichier du système d'exploitation. Ils possèdent certains avantages par rapport au moniteurs matériels :

- Facilité de développement et d'évolutions
- Stockages des données et des résultats

A contrario, certains défauts leur incombent de part son implantation moins proche du matériel :

- Capacité de traitement des débits moindre
- Moins bonne finesse d'analyse
- Impact de la mesure sur les performances

3.3.2.1. Zoom : la commande iostat sous unix

Pour mesurer et contrôler la charge des unités d'entrées/sorties sur un système unix, il est possible d'utiliser la commande **iostat**. Elle permet notamment de :

- voir, avec l'option -d le débit de chaque disque (Kbps), le temps de service moyen pour chaque commande (serv) ainsi que le taux de transferts par seconde (tps).
- voir, en utilisant l'option -D, le nombre de lectures (rps) et d'écritures (wps) par seconde sur chaque disque ainsi que le % d'utilisation du disque (util). Voici un exemple de sortie avec cette option :
- voir des statistiques détaillées sur chaque disque. Cela est possible avec l'option -x. Parmi les données disponibles, le nombre moyen de commandes en attente (wait), le nombre moyen de commandes en traitement (actv), le temps de service (svc_t), le pourcentage de temps que le disque est occupé (%b) et le pourcentage du temps où il y a des demandes en attente (%w). Les autres données sont des versions plus détaillées des données des autres options (divisées en lecture et écriture).

- Voici un exemple de sortie :

```
> iostat -x
```

	extended disk statistics								
disk	r/s	w/s	Kr/s	Kw/s	wait	actv	svc_t	%w	%b
fd0	0.0	0.0	0.0	0.0	0.0	0.0	730.5	0	0
sd1	0.2	0.0	0.7	1.1	0.0	0.0	53.0	0	0
sd3	0.4	1.5	2.9	11.2	0.0	0.4	192.7	0	2
sd6	0.0	0.0	0.0	0.0	0.0	0.0	83.7	0	0
sd15	0.9	1.1	5.0	8.4	0.0	0.1	46.1	0	2
sd16	0.9	1.1	5.0	8.4	0.0	0.1	44.5	0	2
sd17	0.7	0.8	4.4	6.9	0.0	0.2	163.4	0	1
sd18	0.3	0.2	2.5	3.2	0.0	0.0	31.8	0	1
sd31	0.5	0.2	2.3	1.2	0.0	0.0	91.3	0	1
sd33	0.8	5.5	10.7	34.6	0.0	0.3	47.6	0	7
sd36	0.0	0.0	0.1	0.0	0.0	0.0	137.9	0	0

Figure 8 : Sortie d'écran type de la commande iostat

3.4. Intérêts et limites

Le principal avantage de la mesure directe des performances est d'obtenir des informations d'une grande précision en situation concrète. En effet, nous avons à disposition le système à mesurer ainsi que la charge de travail réelle qui lui sera soumise.

Les indicateurs obtenus via la mesure des performances ont l'énorme avantage d'avoir une correspondance directe avec l'usage réel de l'entrée-sortie : pour une application client serveur, le seuil de saturation exprimé en Mb/s équivaut à un nombre connu de requêtes par seconde, donc un nombre défini d'utilisateurs simultanés.

Cependant le principal inconvénient est justement la nécessité d'avoir à sa disposition le système et le mettre en place, ce qui restreint l'usage des mesures à un objectif de contrôle de la performance : il n'est donc pas question de l'utiliser à des fins comparatives, sauf à acquérir tous les systèmes candidats.

Le second inconvénient de la mesure des performances est la difficulté de mise en place, que ce soit via les moniteurs logiciels, ou matériels, ils nécessitent généralement des moyens complexes et une méthodologie rigoureuse.

Chapitre 4. Les étalons de performance ou benchmarks

4.1. Vue d'ensemble

4.1.1. Définition et objectifs

Un benchmark ou étalon de performance est un outil qui permet de comparer les performances de différents systèmes en vue de choisir le plus adapté, et en l'occurrence le sous-système d'entrées-sorties le plus performant.

Il doit pour cela :

- Permettre la compréhension du système
- Pouvoir mesurer une large palette de matériels
- Assurer une comparaison équitable

4.1.2. Architecture d'un benchmark

Un benchmark s'articule autour de 3 modules :

- Un programme permettant de générer une charge de travail à soumettre au système.
- Un programme capable de récolter les données brutes issues de l'activité du système.
- Un programme qui analyse les statistiques et les synthétise en vue d'une présentation permettant des comparatifs.

4.1.3. Positionnement des benchmarks

Il existe une large gamme de benchmark. Leur positionnement dépend de trois critères, indépendants les uns des autres. Le choix d'un benchmark dépend des critères

d'évaluation que l'on souhaite. Le benchmark idéal permettrait d'ajuster chacun des critères suivant les besoins de l'évaluation des performances.

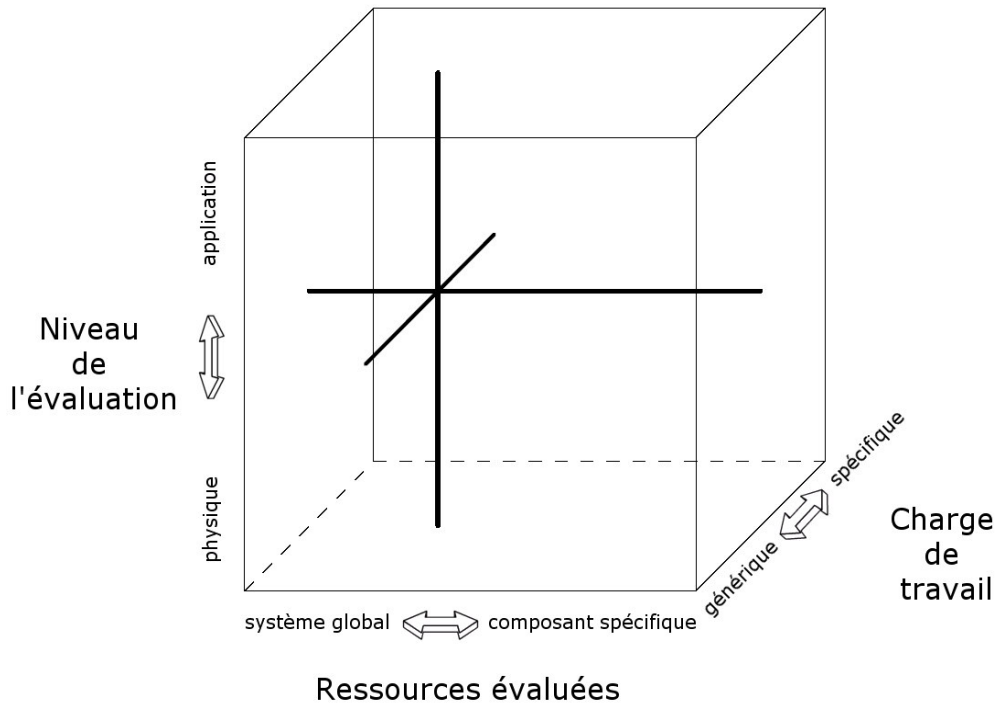


Figure 9 : Positionnement d'un benchmark

4.1.3.1. Ressources évaluées : Système global ↔ Composant spécifique

Un benchmark peut, soit mobiliser toutes les ressources du système (Processeur, mémoire, réseau et E/S) et agréger les résultats, soit se centrer sur une partie unique tel que les E/S.

4.1.3.2. Charge de travail : Générique ↔ Spécifique

Un benchmark peut soumettre au système une charge de travail générique qui serait un profil d'utilisation moyen, ce qui permet de confronter les différents systèmes d'entrées-sorties à un étalon standard et obtenir également un aperçu général des performances de celle-ci, ce qui peut être très utile pour des comparaisons d'ordre marketing.

Un benchmark peut également reproduire l'utilisation d'un type d'application très spécifique (un serveur web par exemple) et donc permettre un choix plus fin et plus réaliste.

4.1.3.3. Niveau de l'évaluation : Physique ↔ Application

Le choix du benchmark dépend de la « couche » que l'on souhaite évaluer dans le systèmes d'E/S : au niveau matériel et donc avoir une information fiable sur le potentiel maximum d'une E/S ou alors mesurer le système au niveau du système de fichier pour avoir une comparaison plus réaliste des résultats qui serait obtenus par l'application utilisatrice (par exemple de mesurer l'adéquation du système de fichiers avec le matériel (E/S) qu'il gère).

4.2. Les benchmarks sur le marché

4.2.1. Survol des benchmarks disponibles

4.2.1.1. TPC

Le TPC (Transaction Processing Council) est une association regroupant de nombreux constructeurs et éditeurs de solutions de gestion de bases de données. Elle existe depuis le début des années 90 et a notamment pour objectif de faire émerger des benchmarks standards pour évaluer les performances globales des systèmes, et ceci pour les usages-type.

Nom	Année	Usage
TPC-C	1992	Transactions multiples et complexes : le plus connu et le plus utilisé, simule un usage varié et robuste du système
TPC-H	1999	Transactions décisionnelles
TPC-R	1999	Equivalent au TPC-H mais avec une gestion plus fine des transactions en vue d'optimiser le paramétrage du systèmes
TPC-W	2000	Simule des transactions type d'un site de commerce électronique
TPC-A	1989	(obsolète) Premier benchmark, usage simple du système
TPC-B	1990	(obsolète) Comme le TPC-A mais pour des transactions type SGBD
TPC-D	1995	(obsolète) Précurseur du TPC-H

Figure 10 : Vue d'ensemble des benchmarks du TPC

Les métriques utilisées sont de deux ordres :

Premièrement les *performances* globales du système avec des indices démontrant la capacité de traitement maximum d'un système pour les opérations type :

- TPC-C : tpmC (Transaction Per Minute)
- TPC-W : WIPS (Web Interaction Per Second)
- TPC-H : QphH (Query Per Hour)

Mais également *un rapport prix/performances* (Prix / tpmC ...): en effet de nombreux constructeurs informatiques utilisent les benchmarks du TPC pour affirmer leur supériorité technique et bénéficier de retombées publicitaires. Pour cela, ils n'hésitent pas à créer des architectures de plusieurs centaines de processeurs, de disques et de giga-octets de mémoire, uniquement dans le but d'apparaître en haut des classements. Cependant, ces architectures de concours coûtant plusieurs centaines de milliers de dollars restent peu accessibles au commun des utilisateurs, et le calcul du rapport prix / performance est un très bon indicateur pour des usages réels.

Chacune des applications que représentent les benchmarks du TPC fait un usage global du système (processeur, mémoire, stockage, voir réseau)

Le degré de sollicitations des entrées-sorties dépend du benchmark choisi, très fort dans le cas du TPC-C et TPC-H , et permet donc de le tester dans un environnement global très proche de l'usage réel. L'avantage des benchmarks du TPC par rapport à d'autres benchmarks globaux est qu'ils sont reconnus comme des standards par la grande majorité des constructeurs et éditeurs de logiciels et il est donc aisé de trouver les mesures de performance d'un système que l'on souhaite acquérir et de le comparer aux standards du marché.

4.2.1.2. IOzone

IOzone est un benchmark qui effectue ses tests et mesures au niveau du système de fichiers. Il prend en compte un nombre important d'opérations, les lectures, les écritures, mais également des opérations spécifiques au système de fichier : ouverture, recherche etc.

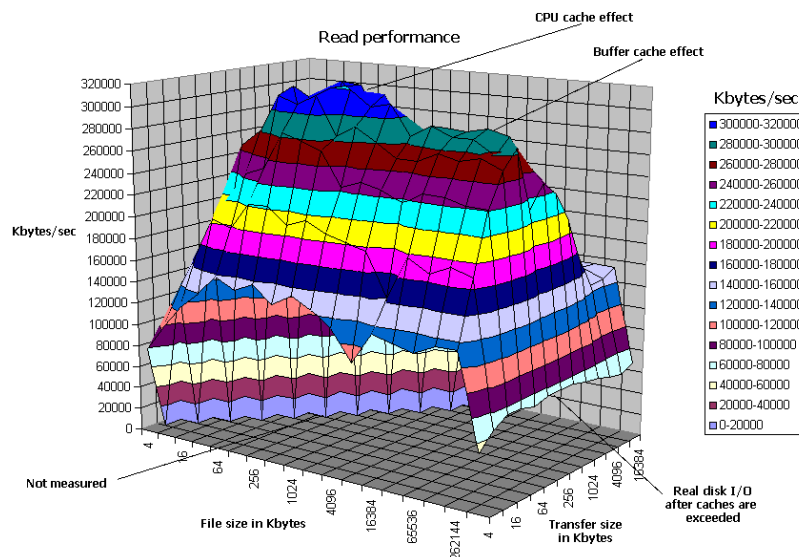


Figure 11 : Présentation des résultats de IOZone

IOzone adopte une présentation tridimensionnelle de ses résultats, elle permet une analyse fine et détaillée de l'impact des différents mécanismes du sous-système de stockage sur les performances, il est donc possible de faire sortir les forces et faiblesses des entrées-sorties testés en fonctions des types de requêtes, et donc d'ajuster au mieux les applications qui pourront solliciter le système.

4.2.2. Zoom : SPC-1

4.2.2.1. Présentation / origine

Le *SPC Benchmark-1* est le premier benchmark spécifique aux E/S reconnu par l'ensemble des professionnels et acteurs du secteur.

Il a pour origine le Storage Performance Council (SPC), une entreprise à but non lucratif créée dans le but de définir, normaliser et promouvoir les références des sous-systèmes de stockage, ainsi que pour diffuser, auprès du secteur informatique et de ses clients, des données objectives et vérifiables de performances.

Le SPC a parmi ses membres IBM, HP, Seagate, Sun, Adaptec, Nec et Veritas ainsi que de nombreux autres constructeurs et utilisateurs de solutions de stockage.

Le *SPC Benchmark-1* a pour objectif de fournir des informations objectives, pertinentes et vérifiables pour tous les acquéreurs de systèmes d'E/S. Il a été conçu de manière à pouvoir tester un large panel d'architectures de sous-systèmes d'E/S différents.

4.2.2.2. Architecture

Le cœur du *SPC Benchmark-1* est un générateur de charge de travail, qui simule le travail de 3 applications distinctes, chacune d'elle générant des flux de commandes et de requêtes de natures différentes.

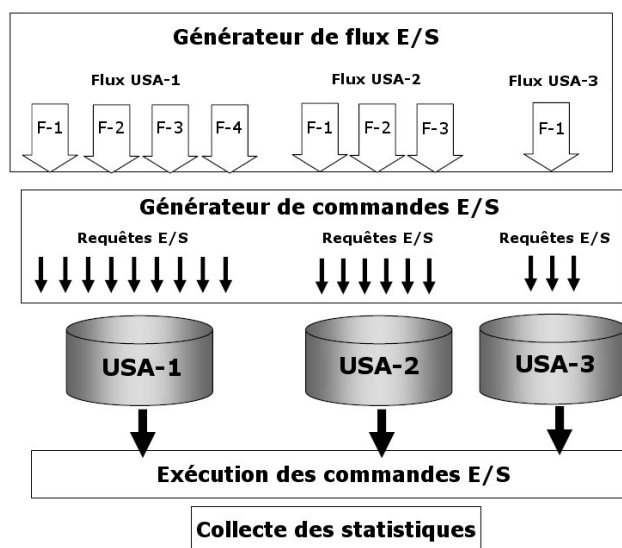


Figure 12 : Architecture de SPC-1

Unité de stockage applicatives (USA) : il s'agit d'une vue logique de l'espace de stockage nécessaire à une application. Chaque unité est une interface d'abstraction vers un système d'E/S qui peut être différent (Disque attaché, système RAID, ou NAS)

Flux E/S : il s'agit d'une suite de commandes et de requêtes obéissant à certains paramètres spécifiés (type d'opération, taille, vitesse, quantité etc.)

Requête / Commande E/S : unité de base de chaque flux, comprend une entête spécifique à son rôle plus une série de caractères générés aléatoirement.

Chaque USA est composé de un ou plusieurs flux et représente un type d'usage différent :

USA-1 : charge de travail de type magasin de données composée de 4 flux :

3 flux 50%lecture, 50% écriture de tailles et plus ou moins aléatoires

1 flux de lecture de taille variable

USA-2 : charge de type données utilisateur 3 :

2 flux 70%lecture, 30% écriture de tailles et plus ou moins aléatoires

1 lecture

USA-3 : 1 flux d'écriture séquentiel. (type log)

4.2.2.3. Fonctionnement

L'évaluation consiste dans un premier temps à augmenter au fur et à mesure la charge jusqu'à atteindre la saturation et obtenir la première valeur (le débit maximal appelé **SPC-1 IOPS**)

Ensuite il s'agit de faire fonctionner le générateur de charge à faible régime (10% de la charge de saturation) pour mesurer le temps de réponse minimum (**SPC-1 LRT**)

4.2.2.4. Evolutions

Le SPC prépare également un second benchmark appelé le SPC Benchmark-2 (SPC-2) qui a pour principal attrait le fait de pouvoir spécifier le type de charge de travail à tester :

- Traitement de fichiers de grande taille
- Longues requêtes de bases de données
- Flux vidéos

4.3. Critique des benchmarks

Le principal attrait des étalons de performances est de fournir dans un outil unique un moyen permettant des comparaisons efficaces des performances de différents sous-systèmes d'entrées-sorties. De plus ces outils sont largement répandues au sein des constructeurs et des utilisateurs, et permettent donc d'accéder facilement aux résultats sans avoir à les pratiquer.

Les benchmarks permettent à la fois d'évaluer les performances des entrées-sorties dans système le complet (TPC) mais également de ne pratiquer que des mesures du sous-système de stockage indépendamment de ce qu'il y a autour (SPC ou I/Ozone). Enfin les benchmarks permettent d'avoir une idée précise du potentiel maximum que peut offrir chacun des sous-systèmes.

Cependant, malgré les avantages qu'ils procurent, leur principal point faible provient de leur volonté à offrir un socle de mesure le plus large possible, en effet, dans de nombreux cas réels, il est nécessaire d'avoir des résultats plus précis de ce que pourrait être les performances réelles d'une application. Or les charges de travail synthétiques utilisés par les étalons ont pour vocation de simuler une utilisation « classique ».

Chapitre 5. La charge de travail

La charge de travail d'un système est l'ensemble des demandes de traitement, provenant des programmes, données ou commandes.

5.1. Importance de la caractérisation de la charge de travail

Les performances d'un système d'entrées-sorties sont aussi bien déterminées par ses propres caractéristiques que par la nature de la charge de travail qu'il traite. Les comparaisons de performance entre les systèmes ne sont significatives que si les charges de travail traitées sont identiques.

La compréhension de la charge de travail est donc indispensable pour déterminer la performance d'un système et l'améliorer.

La caractérisation d'une charge de travail consiste à une description de celle-ci en terme quantitatif via des paramètres et des fonctions, l'objectif étant de faire émerger un modèle capable de décrire, voire reproduire le comportement de la charge au travers de ses caractéristiques principales.

5.2. Applications et utilisation E/S : des profils variés

Le paysage applicatif a énormément évolué depuis les années 70 : au début la plupart des ordinateurs étaient des *mainframes* et leur charge de travail consistait généralement en des traitements par lots de transactions homogènes. L'avènement des réseaux, des interfaces graphiques et le doublement régulier des performances des processeurs a permis l'émergence d'une variété extraordinaire d'applications fondamentalement différentes les unes des autres et ayant des besoins très spécifiques en terme de ressources entrées/sorties.

Quelques exemples d'utilisation des E/S :

Bases de données : sollicitation extrêmement diverses des E/S, lectures et écritures aléatoires de données, opérations très nombreuses généralement de faible taille mais quelques fois de très grande taille, nécessite un système d'E/S robuste et complet à tous les points.

Serveurs de fichiers : capacité de stockage très importante et système d'E/S capable de gérer de très grands flux de données.

Logs et archivage : de multiples écritures séquentielles et très rapides de taille constante.

Serveurs multimédia : accès simultané et long à de nombreux fichiers de taille importante.

Calculs scientifiques : de nombreuses applications de calculs scientifiques de part les quantités gigantesques de données qu'elles manipulent nécessitent des performances optimales des systèmes de stockage. Le profil d'utilisation des entrées-sorties est très varié et dépend des calculs effectués.

5.3. Caractéristiques des charges de travail

5.3.1. Choix des paramètres

Il existe un nombre important de paramètres mesurables lors d'une sollicitation du système d'E/S. Cependant il est important de se limiter à un nombre restreint de paramètres. Il faut que le système puisse être modélisable pour pouvoir être simulé par la suite. Les paramètres doivent répondre à 3 critères :

- Dépendre de la charge de travail et non pas du système.
- Etre spécifique aux Entrées/Sorties.
- Avoir un impact non négligeable sur les performances.

5.3.2. Paramètres relatifs aux E/S

Type d'opération : Il s'agit de la nature du travail effectué par l'E/S, on peut le classer en 2 catégories (vue physique) :

- Lecture
- Ecriture

Ou alors affiner les paramètres pour une meilleure compréhension des réactions du système et de ses goulets d'étranglement :

- Lecture
- Ecriture
- Recherche
- Ouverture de fichier
- Fermeture de fichier

Taille des requêtes : ce paramètre est très important, connaître la distribution de la taille des différentes requêtes est un facteur clé pour des optimisations possibles du système, par exemple, une surcharge à cause d'un nombre très important de petites lectures peut être diminuée en effectuant une lecture prédictive des données susceptibles d'être demandées.

Séquentialité des accès : des accès non séquentiels signifient que les adresses des secteurs demandés ne se suivent pas, obligeant la tête de lecture de se repositionner à chaque requête.

Taux d'arrivées : il s'agit de la fréquence des arrivées, elle peut être exprimée simplement en nombre de requêtes par unité de temps, mais est généralement plus complexe à exprimer car les accès E/S ne sont pas toujours réguliers et arrivent souvent en rafales plus ou moins brusques et soutenues ; ce qui a des répercussions complètement différentes sur le système.

5.4. Caractérisation : méthodologie

5.4.1. Collecte de données

La collecte des données peut se faire via des moniteurs de performances (voir 3.3) ayant des fonctions avancées de collecte d'informations. Elle se fait via un fichier de trace listant l'ensemble des opérations effectués par le système.

5.4.1.1. Zoom : « Pablo Performance Environment »

Pablo est un utilitaire issu du laboratoire du même nom de l'université de l'Illinois. Cet utilitaire permet de capturer les données issues de l'activité du système et de fournir une analyse quantitative de celles-ci. Il consiste en une librairie de fonctions qui s'exécutent sur le système en parallèle à l'application qu'il doit mesurer.

Au fur et à mesure de l'exécution du programme cible, la librairie génère un fichier de trace qui contient la liste de toutes les opérations et événements avec la date et l'heure précise de l'action ; la nomenclature du fichier – appelé SDDF - a été définie préalablement et le rend donc exploitable pour une analyse ultérieure.

Pablo possède également une extension spécifique dédiée aux entrées-sorties. Il est possible de capturer en détail les événements qui ont lieu pendant l'exécution d'une application : le type d'opération, la durée d'attente, le fichier ..

La librairie Pablo I/O se greffe à l'application à mesurer et fournit des fonctions analogues aux fonctions de traitement de fichier classique, par exemple (tracefopen) pour l'ouverture de fichier, ces fonctions encapsulent les fonctions standards (fopen) en y ajoutant le trace.

Exemple : quand une fonction de l'interface de trace est appelé (tracefopen) :

1. L'horodateur est vérifié
2. La fonction E/S est appelée (fopen)
3. L'horodateur est re-vérifié, et la durée de l'opération est calculée
4. L'opération est enregistrée dans le fichier de trace

L'appel d'une routine et d'une en-tête en début de programme provoque le remplacement à la compilation de toutes les fonctions de base par celles de trace, ce qui évite de modifier toutes les fonctions à la main.

```
1 #define IOTRACE
2 #include "IOTrace.h"
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 main()
7 {
8     FILE *fp;
9     char buffer[1024];
10    size_t cnt;
11
12    initIOTrace(); /* Initialize I/O Extension */
13
14    fp = fopen( "/etc/motd", "r" );
15    if ( fp != NULL ) {
16        cnt = fread( buffer, sizeof(char), 1024, fp );
17        fclose( fp );
18    }
19
20    endIOTrace(); /* Trace termination routines */
21    endTracing();
22}
```

Figure 13 : Programme exemple en C : remplacement automatique des appels de l'interface

5.4.2. Traitement des données statistiques

Les données recueillies pour chaque paramètre choisi peuvent être analysées d'une manière plus ou moins fine. Il existe de nombreuses méthodes statistiques pouvant être utilisées pour caractériser un paramètre : de la plus sommaire à la plus complexe.

Le choix d'une méthode dépend des objectifs de l'évaluation et de la nature de la charge de travail, l'objectif étant d'avoir un modèle le plus représentatif possible tout en essayant de le garder le plus simple possible.

Moyenne

La statistique la plus évidente à calculer sur un échantillon numérique, celle dont l'interprétation est la plus intuitive, est la moyenne. La moyenne d'un échantillon est la somme de ses éléments divisée par leur nombre.

$$\bar{x} = \frac{\sum n_i x_i}{\sum n_i}$$

Cette valeur est cependant trop simpliste et ne peut être utilisée que pour caractériser certains aspects de la charge de travail tel que le type d'opération.

Variance

Les notions de variance et d'écart-type servent à quantifier la dispersion d'un échantillon autour de sa moyenne. La définition est la suivante :

$$s^2 = \frac{1}{2} \sum_{i=1}^n (x_i - \bar{x})^2$$

L'écart type est utile pour mieux définir les caractéristiques d'un paramètre tel que la taille des requêtes, mais manque cependant de finesse pour les applications ayant un panel de requêtes plus large, tel que les SGBD.

Histogramme / quantiles

Un histogramme représente les fréquences relatives des différentes occurrences d'un paramètre, pour les valeurs continues, cela nécessite de séparer l'étendue des données en un nombre d'intervalles significatifs (quantiles, déciles ..). Les quantiles sont un bon moyen pour caractériser la taille des requêtes.

Modèles de Markov

En plus du nombre de requêtes de chaque type, il peut être utile de déterminer leur ordre d'arrivée, les nouvelles requêtes dépendant généralement des requêtes précédentes (ex : ouverture, lecture puis fermeture).

Un modèle de Markov est un processus stochastique qui prédit le comportement futur à partir de l'état actuel d'un système. Il est représenté via un ensemble d'états, appelés chaîne, où la probabilité de transition entre les différents états est définie et représentée grâce à une matrice de transition.

Dans la prédiction des requêtes, les états représentent les types d'opération, et les transitions le lien entre deux requêtes successives.

de/à	Ouverture	Lecture	Fermeture
Ouverture	0.2	0.6	0.2
Lecture	0.2	0.5	0.3
Fermeture	0.7	0.2	0.1

Figure 14 : Matrice simplifiée des probabilités de transition

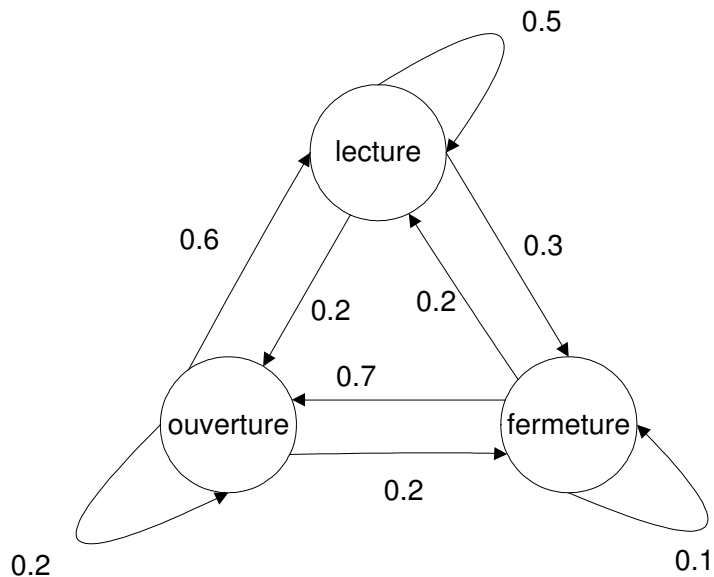


Figure 15 : Diagramme d'état transition d'un modèle de Markov

Regroupement (clustering)

Lors de traitement d'un grande quantité de données, il est difficile de résumer la charge de travail à une simple moyenne ou histogramme à paramètre unique, on peut alors utiliser les techniques statistiques de regroupement. Cela consiste à classer les résultats en petits groupes représentatifs. Il existe pour cela différentes méthodes et algorithmes pour faire isoler les caractéristiques intéressantes. Un exemple ci-dessous du regroupement de 30 requêtes en 5 groupes différents.

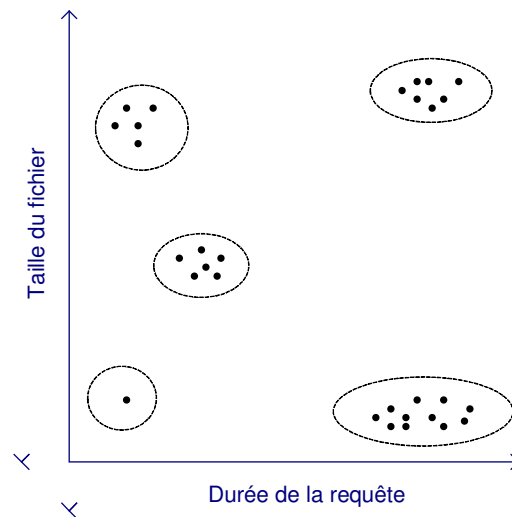


Figure 16 : Exemple de regroupement de données en 5 clusters

5.5. Analyse

L'analyse des données statistiques a permis d'une part de caractériser quantitativement la charge de travail et donc offre la possibilité de la reproduire lors d'une simulation (voir chapitre suivant). Elle permet également de faire remonter des informations essentiels qui permettront aux concepteurs d'application et de système de stockage d'optimiser le processus d'utilisation des entrées-sorties.

Quelques exemples :

- Une application nécessitant des lectures simultanées sur plusieurs fichiers devra effectuer des allers/retours incessants entre des parties éloignées du disque. Le choix d'une vitesse de rotation du disque plus élevée permettra de bénéficier de temps d'accès plus rapides.
- Pour une application nécessitant de multiples écritures séquentielles, la présence d'une mémoire tampon importante permettra d'effectuer des écritures par lots à des intervalles plus longs, rendant le système plus disponible pour d'autres opérations.
- La connaissance des probabilités de transition entre les différents types d'opération permettra d'optimiser les algorithmes d'ordonnancement (voir 2.2.3.1)

Chapitre 6. Modélisation et simulation

6.1. Définition et objectifs

La simulation consiste à imiter un processus physique par un programme. Le simulateur est donc chargé de reproduire les comportements des entrées-sorties. Un modèle de simulation est un système logiciel écrit pour représenter le comportement dynamique d'un système en représentant ses états et ses transitions d'état.

6.2. Modélisation

La démarche de modélisation d'un sous-système d'entrées-sorties consiste à modéliser les différentes parties entrant dans le processus d'utilisation des entrées-sorties :

6.2.1. La charge de travail

La génération d'une charge de travail synthétique, réaliste et paramétrable est indispensable pour correctement exploiter le modèle d'un disque. Cette charge de travail synthétique a pour socle un générateur de nombres aléatoires couplé à différentes techniques :

6.2.1.1. A partir d'un fichier de trace

Il est possible de générer une charge de travail à partir de la trace capturée de l'activité d'un disque, cela permet d'avoir d'une façon relativement aisée une charge de travail fidèle à la réalité ; cependant, il est dans ce cas nécessaire de traiter un grand volume de données et on perd la possibilité d'ajuster « au vol » les différents paramètres caractérisant la charge de travail.

6.2.1.2. A partir d'une distribution statistique type

A partir des caractéristiques statistiques recueillies lors de l'étude des charges de travail (voir 5.4.2), il est possible en simplifiant le modèle de faire apparaître une distribution

statistique connue, il est donc possible grâce à sa formule mathématique couplé au générateur de nombres uniformes d'obtenir des données correspondantes à la distribution et donc à la charge de travail souhaitée.

6.2.1.3. Ré- échantillonnage / Bootstrapping

L'idée de base du bootstrapping est de reproduire une charge de travail complète à partir d'un échantillon représentatif de celle-ci. Cela consiste en la reproduction d'une procédure statistique en considérant que le modèle initial (issu d'un trace représentative) est exacte et en générant un autre modèle à partir des données précédentes. La nouvelle trace ainsi obtenue, constituée de données appelées bootstrap, a la même taille que la trace générée par le modèle initial et peut être constituée de suites identiques résultant d'une double (ou triple ...) répétition de données. Un calcul statistique est réalisé pour chaque donnée, les nouvelles valeurs étant calculées pour chaque paramètre à estimer. La différence entre les paramètres calculés à partir des données initiales et la moyenne des paramètres calculées à partir des données issues du bootstrap détermine la tendance du calcul initial.

6.2.2. Le système de fichiers

La modélisation du système de fichiers peut paraître une tâche assez simple, la réalité en est tout autre, d'une part le travail effectué par les systèmes de fichiers ne se résume pas à une conversion des fichiers en données binaires – il suffit de voir la fragmentation de certains disques pour s'en convaincre – d'autre part il existe une multitudes d'architectures existantes.

Au lieu de créer un modèle approximatif qui ne reproduirait que certains aspects du système de fichiers, une approche intéressante consisterait à réutiliser le code d'un système de fichiers existant. Les développements du monde *open source* mettent à notre disposition de nombreux systèmes de fichiers de qualité avec le code source et une documentation riche. Il « suffit » donc de faire fonctionner le système de fichier en dehors de son environnement d'exécution normal (noyau du système d'exploitation), et de l'exécuter tel une application classique dans un processus standard, moyennant bien sur l'adaptation les interfaces d'entrée (charge de travail) et de sortie (disque physique). Cela correspondrai d'ailleurs à une configuration de *debuggage* classique tel qu'utilisée par les développeurs de ces même systèmes de fichiers.

6.2.3. Le sous-système physique de stockage (disque)

La modélisation du disque est l'essence même de l'intérêt de la simulation, en effet l'objet de la simulation n'est pas de modéliser un disque figé qui fournirait des performances

standards largement attendues, le modèle doit permettre une amélioration des performances, en offrant la possibilité de détecter les problèmes, d'ajuster les paramètres et de simuler des solutions nouvelles. Le maître mot de la modélisation d'un disque est la modularité et le paramétrage, ce qui nécessite avant tout une méthode précise :

6.2.3.1. Démarche pour définir un modèle de disque

1. Segmenter les différentes parties du sous-système d'entrées-sorties, et répéter l'opération pour chacune des parties.
2. Analyser chacune des parties ou sous partie pour trouver le type de modélisation correspondant.
3. Etablir une hiérarchie de modèles.
4. Définir les modèles.
5. Identifier les paramètres nécessaires à l'ajustement des modèles et pour prendre en compte tous les cas réels.
6. Obtenir les paramètres type d'usage, par mesure ou par estimation.
7. Evaluation et ajustement des différents modèles un à un de façon verticale (de bas en haut).
8. Evaluation et ajustement du modèle global.
9. Valider le modèle en appliquant des paramètres réelles et vérifier leur correspondance avec la réalité.

6.2.3.2. Modèle de disque

L'approche consiste à modéliser chacun des composants du disque tel que les files d'attente, caches, contrôleurs et mécanismes de disque, et de les agréger pour former un modèle complet. (voir 2.1)

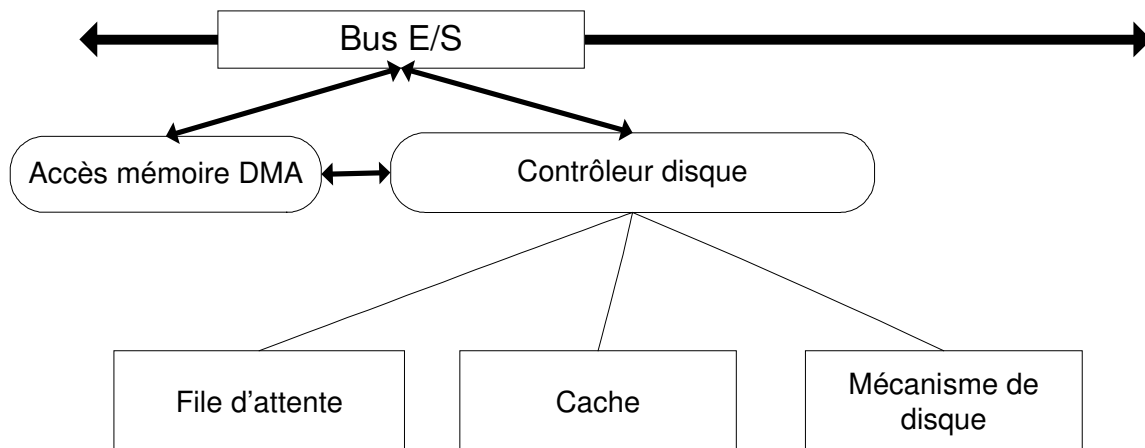


Figure 17 : Composants majeurs d'un modèle de disque

Les sous modèles à agréger doivent comporter un minimum de propriétés qui puissent être paramétrables, ci-dessous une liste minimale de paramètres à intégrer en vue de reproduire un comportement fonctionnel d'un disque.

Composant	Propriétés
Mécanisme du disque (voir 2.1.2)	Capacité Vitesse de rotation Temps de commutation entre cylindre Temps de commutation entre pistes Nombre de cylindres Nombre de pistes par cylindre Nombre de secteurs par piste
Cache (voir 2.2.3)	Taille du cache Algorithme de remplacement des données Politique de lectures prédictive Politique de mise en cache Politique d'écriture sur disque Durée de recherche en cache Taux de transfert du cache
File d'attente (voir 2.2.3.1)	Algorithme d'ordonnancement Taille maximum de la file d'attente

Figure 18 : Liste non exhaustive des propriétés à prendre en compte lors de la modélisation du disque.

6.3. Zoom : DiskSim

6.3.1. Présentation

DiskSim est un simulateur de système de stockage particulièrement configurable, il crée au sein de la « school of Computer Science » au sein de l'université de Carnegie Mellon à Pittsburg, il est écrit en langage C.

Le simulateur est décomposé en modules reproduisant chacun une partie particulière du sous-système :

- Ordonnanceur / file d'attente des requêtes
- Gestion des caches
- Organisation des données

Il permet également de simuler via des modules annexes des fonctions périphériques au système de stockage :

- Contrôleurs intérimaires
- Bus d'E/S
- Drivers de disque

Sa conception modulaire permet de pouvoir ne tester qu'une seule partie du sous-système, en bénéficiant d'un niveau de paramétrage maximum et d'une génération détaillée de la trace de l'exécution. DiskSim peut également s'interfacer avec différentes applications comme un générateur de charges de travail ainsi que s'intégrer dans un simulateur plus global du système.

6.3.2. Fonctionnement

L'utilisation de DiskSim s'effectue via le lancement d'un exécutable et de paramètres associés :

```
disksim <parfile> <outfile> <tracetype> <tracefile> <synthgen>
```

disksim : nom de l'exécutable

parfile : nom du fichier de paramètres

outfile : nom du fichier de sortie

tracetype : spécifie le format de la trace d'entrée (si celle-ci existe)

tracefile : nom du fichier de trace en entrée

synthgen : si oui ou non le générateur interne de charge de travail doit être utilisé

Le fichier de paramétrage du disque (parfile) contient plus de deux cents paramètres différents et permet donc de reproduire le comportement de la quasi totalité des disques sur le marché.

Le fichier de sortie (outfile) permet également d'obtenir une centaine d'indicateurs sur le fonctionnement de l'entrée-sortie (des statistiques du contrôleur disque, jusqu'aux états des différentes parties mécaniques).

La charge de travail peut provenir, soit de la trace d'une utilisation de disque réelle (en transmettant à diskSim la nomenclature du fichier de trace via tracetype), soit du générateur de charge de travail synthétique intégré au simulateur : on fournit donc les paramètres la caractérisant (synthgen).

6.4. Intérêts et limites

La simulation est une technique très valable pour évaluer la performance des entrées-sorties. Un modèle de simulation peut reproduire certains aspects du comportement d'un système même si celui-ci n'existe pas. On peut alors prendre des mesures réelles à l'aide d'outils de simulation et évaluer le comportement du système sous certaines circonstances.

La simulation, comme toutes les techniques de modélisation, offrent l'inconvénient d'utiliser des modèles qui sont inévitablement des représentations partielles et approximatives de la réalité. Ils fournissent donc des résultats qui peuvent être pris en compte de façon sérieuse uniquement si le modèle a pu être validé : en construisant le système.

La simulation permet de prévenir la construction de systèmes mal conçus en faisant ressortir leurs problèmes avant la construction.

Conclusion

La complexité et la diversité des technologies présentes dans le processus des entrées-sorties, et l'émergence récente de nouvelles architectures d'entrées-sorties parallèles coûteuses montrent la nécessité de disposer d'un support de stockage important, fiable et performant.

Les performances des entrées-sorties, cruciales pour la performance des applications dépendent d'un grand nombre de paramètres. L'évaluation des performances au travers de ses différentes techniques offre un moyen de choix pour obtenir le meilleur des entrées-sorties. Que ce soit la mesure des performances, utile pour le contrôle, les étalons de performances pour la sélection, ou les méthodes de modélisation et de simulation pour la projection, ces méthodes allant de la plus simple à la plus complexe correspondent à un réel besoin, le choix de l'une d'entre elles dépendra des enjeux et des moyens humains, financiers disponibles.

Les entrées-sorties restent malgré tout le goulet d'étranglement des systèmes informatiques d'aujourd'hui, et ceci malgré les moyens colossaux mis en œuvre pour les améliorer. La tendance prise depuis plusieurs années est d'orienter l'architecture des applications de manière à n'utiliser les entrées-sorties que pour les tâches non réalisables en mémoire vive. L'évaluation des performances - via la caractérisation de la charge de travail - va également dans ce sens en offrant un moyen de mieux comprendre le fonctionnement de l'application et de l'adapter en conséquence.

Glossaire

Burst mode : méthode assurant une transmission continue des informations entre deux éléments d'un ordinateur, et permettant (principalement) au processeur de recevoir les données sans wait state.

Datawarehouse : entrepôt de données en français. Outil d'aide à la décision, basé sur une base de données fédérant et homogénéisant les informations des différents services d'une organisation. Le datawarehouse est la forme la plus sophistiquée des systèmes d'aide à la décision. Il coûte plusieurs millions de francs et peut stocker des centaines de Go de données.

Débuggage : action de déboguer, enlever les fautes et les erreurs dans un programme, de façon à ce qu'il réalise la tâche attendue.

DMA : Direct Memory Access. Accès direct à la mémoire. Méthode de transfert de données dans un ordinateur évitant d'avoir à utiliser le processeur et/ou une zone d'E/S standard. Le processeur lance donc le transfert DMA et peut continuer de faire des calculs indépendants (surtout grâce à sa mémoire cache), les périphériques ayant leur propre canal DMA évitent de faire la queue pour l'accès à la zone d'E/S.

EIDE : Enhanced IDE. Version améliorée de l'IDE, qui peut accueillir quatre périphériques, dont deux lents et deux rapides, en particulier des disque dur. L'un d'eux doit être ATAPI. On utilise le LBA pour les gros disques

Fiber channel : le « Fibre Channel » est une norme de transmission de données en série, permettant d'obtenir un débit de 100 Mo/s sur des liens de quelques kilomètres de long (10 km au maximum).

FIFO : First In , First Out. Premier entré, premier sorti. C'est-à-dire une queue, une file d'attente.

IDE : Intelligent Drive Electronic. Norme de connexion de périphériques. Dans cette norme, un contrôleur ne peut piloter que deux périphériques au plus, dont l'un est toujours prioritaire par rapport à l'autre (un maître et un esclave).

Interleave : utilisé essentiellement sur les disques. Les fichiers sont stockés dans des secteurs non contigus, ce qui permet une rotation plus rapide du disque.

Mainframe : gros ordinateur central, qu'on entoure de terminaux aux faibles capacités. Littéralement, c'est le cadre central, principal, en ferraille, sur lequel sont montées des cartes électroniques.

PCI : Peripheral Component Interconnect. Bus de 32 bits de large, dans l'univers PC. Il propose un débit de 132 Mo/s, à une fréquence de 33 Mhz

SGBD : Système de Gestion de Base de Données

Bibliographie

PETER M. CHEN, DAVID A. PATTERSON [1993]. « A new approach to I/O performance evaluation, self-scaling I/O benchmarks, predicted I/O performance »

JOHN L. HENNESSY, DAVID A. PETERSON [1992]. « Architecture des ordinateurs : Une approche quantitative » McGraw Hill – Chapitre 9 : Entrées-sorties

RAY JAIN [1991]. « The art of computer systems performance analysis, Techniques for experimental design, measurement, simulation, and modelling » John WILEY

RENE J. CHAVANCE [2003] « Slides Intégration des systèmes Client/Serveur – Performances – CNAM »

JAMES GRIFFIOEN, RANDY APPLETON [1994]. « Reducing file system latency using a predictive approach »

STORAGE PERFORMANCE COUNCIL (SPC) « SPC Benchmark-1 (SPC-1) , Official specification Revision 1.7.0 »

STORAGE PERFORMANCE COUNCIL (SPC) « SPC Benchmark-2 (SPC-2) , Public review draft specification Revision 0.8.0 »

TRANSACTION PROCESSING PERFORMANCE COUNCIL « TPC Benchmarks »
<http://www.tpc.org>

WILLIAM D. NORCOTT, DON CAPPS « Iozone Filesystem Benchmark »
<http://www.iozone.org>

GABRIEL GIRARD Université de Sherbrooke, Canada « IFT 628 : systèmes d'exploitation II »

MARIA CALZAROSSA, LUISA MASSARI, DANIELE TESSERA [2000]. « Workload Characterization Issues and Methodologies »

EVGENIA SMIRNI, DANIEL A. REED « Lessons from characterizing Input/Output behaviour of parallel scientific applications »

RUTH A. AYDT [1994] « A user's guide to Pablo® I/O Instrumentation »

ROGER J. NOE [1994] « Pablo® Instrumentation Environment User's Guide »

CHRIS RUEMMLER, JOHN WILKES [1994] « An introduction to disk drive modelling »

CHANDRAMOHAN A. THEKKATH, JOHN WILKES, EDWARD D. LAZOWSKA [1994] « Techniques for file system simulation »

DAVID KOTZ, SONG BAC TOH, SRIRAM RADHAKRISHNAN [1994] « A detailed simulation model of the HP 97560 Disk Drive »

JOHN WILKES [1995] « The Pantheon storage-system simulator »

JOHN S. BUCY, GREGORY R. GANGER, CONTRIBUTORS [2003] « The DiskSim Simulation Environment Version 3.0 Reference Manual »

Table des illustrations

Figure 1 : Objectifs et acteurs de l'évaluation des performances	8
Figure 2 : communication entre composants et E/S.....	11
Figure 3 : Structure logique des plateaux	11
Figure 4 : Structure physique du disque	12
Figure 5 : Evolution des performances des processeurs et des entrées-sorties au cours des dernières décennies	13
Figure 6 : Comparatif des architectures NAS et SAN.....	17
Figure 7 : Evolution du temps de réponse par rapport au débit.....	18
Figure 8 : Sortie d'écran type de la commande iostat	21
Figure 9 : Positionnement d'un benchmark.....	23
Figure 10 : Vue d'ensemble des benchmarks du TPC	24
Figure 11 : Présentation des résultats de IOZone.....	25
Figure 12 : Architecture de SPC-1	26
Figure 13 : Programme exemple en C : remplacement automatique des appels de l'interface.....	32
Figure 14 : Matrice simplifiée des probabilités de transition	34
Figure 15 : Diagramme d'état transition d'un modèle de Markov.....	34
Figure 16 : Exemple de regroupement de données en 5 clusters	35
Figure 17 : Composants majeurs d'un modèle de disque	38
Figure 18 : Liste non exhaustive des propriétés à prendre en compte lors de la modélisation du disque.....	39