

Module : Evaluation de Performances Master 2 Informatique Haute Performance (MIHP)

Soraya Zertal

Li-PaRAD-Université de Versailles
`soraya.zertal@uvsq.fr`

Cours 2 : Workloads

1 Introduction

2 Sélection

- Service délivré
- Niveau de détail
- Représentativité
- Alignement dans le temps

3 Caractérisation

- Moyennes et facteurs de dispersion
- Quantiles et histogrammes

4 Workloads synthétiques

- Timing
- Addressing
- Nombre de sources

5 Workloads réels

- Benchmarks
- Applications utilisateurs

Workloads : Introduction

Il existe deux classes de workloads (charges de travail) : charges **réelles** et **synthétiques**.

Les charges réelles

- **Observées** ou mesurées sur un **système réel** pendant une exécution.
- Uniques et ne peuvent être ré-itérées.

Les charges synthétiques

- **Créées** selon des caractéristiques bien déterminées pour **représenter un contexte/comportement** spécifique.
- Peuvent être ré-itérées à l'infini.

Workloads : Introduction

Les charges réelles

Avantages : représentativité optimale, riches en détails

Inconvénients : grande taille, pas de répétition, mode d'extraction parfois difficile

Les charges synthétiques

Avantages : rapidité de développement, répétition à l'infini, taille réduite, portabilité

Inconvénients : représentativité moindre, beaucoup d'approximations

Comment sélectionner les charges ?

La **sélection de la charge** est cruciale pour l'évaluation des performances puisqu'elle détermine **l'utilité** et **l'exactitude** de l'impact des résultats obtenus par l'analyse.

Quatre points majeurs à considérer pour sélectionner une charge :

- 1 le service délivré par le système
- 2 le niveau de détail demandé
- 3 la représentativité des charges
- 4 l'alignement temporel au sein des charges

Sélection : Service délivré

Service délivré

Le système à analyser est perçu comme un fournisseur d'un ensemble de services (voir approche). Il faut considérer le **service concerné par l'étude** et bien le séparer des autres services que le système peut fournir.

Remarque

Un service indépendant du contexte est plus représentatif, donc plus approprié pour l'analyse puisqu'il ne restreint pas la validité des résultats à un contexte bien particulier.

Sélection : Service délivré

Exemple 1

L'analyse porte sur le calcul:

Le système analysé est le CPU mais on considère l'UAL puisqu'elle est le composant qui délivre le service analysé.

Le métrique utilisé pour l'analyse (ex. temps d'exécution) doit être au niveau système et non au niveau UAL.

Sélection : Service délivré

Exemple 2

L'analyse porte sur le stockage de données:

Le système analysé est le système de stockage mais on considère le disque puisqu'il est le composant qui délivre le service analysé.

Le métrique utilisé pour l'analyse doit être au niveau système et non au niveau disque. Plutôt le temps de réponse moyen, débit en TPS...etc que le temps d'accès disque.

Sélection : Niveau de détail

Relation avec la méthode d'évaluation

La **modélisation** mathématique nécessite moins de détails (utilise souvent **moyennes** ou **distributions**).

La **simulation** et la **mesure** utilisent des charges estampillées par le temps : **traces** ou (charges exécutables).

Sélection : Niveau de détail

Les traces sont souvent de **grandes tailles** et **riches en détails**

On utilise des traces de haut niveau afin de réduire la quantité de données à analyser.

On adapte la finesse du détail des traces à analyser selon les résultats obtenus et ceux souhaités.

Sélection : Représentativité

Représentativité

Un workload de test doit être représentatif d'une certaine application/comportement ou une classe d'applications avec des paramètres communs.

Pour qu'une **charge soit représentative d'une application**, il faut que les deux correspondent en termes de :

- Taux d'arrivée
- Ressources demandées
- Profil d'utilisation de la ressource

Sélection : Alignement dans le temps

Alignement dans le temps

Les workloads doivent suivre et représenter les changements temporels dans le comportement du système.

On distingue plusieurs niveaux:

- **Faible** : donnant juste **la tendance** du comportement dans le temps,
- **Fort** : retraçant avec exactitude **tous les événements** survenus dans le système,
- **Intermediaire** : retraçant les **changement essentiels** qui forment la tendance du comportement du système.

Sélection

Autres considérations pour la sélection

De bon usage pour la calibration de résultats

- **Niveau de charge** reflète les régimes sous lesquels un système peut opérer : l'utilisation du système à sa capacité maximale (**meilleur cas**), en dessous de sa capacité (**pire cas**) ou à un niveau habituellement observé (**cas typique**).
- **Répétition**: pour obtenir les "mêmes" résultats à chaque exécution avec les mêmes charges.

Sélection

Remarque

Les charges qui introduisent une grande variabilité des résultats sont peu appréciables car elles génèrent des **intervalles de confiance** assez grands.

Intervalle de confiance

C'est l'intervalle de valeurs valides pour les résultats avec une certaine probabilité.

Plus la probabilité est grande (ex. 95%) et l'intervalle de confiance réduit, meilleure est la précision des résultats

Sélection

Calcul des intervalles de confiance

Pour une distribution Normale et une taille de l'échantillon n assez grande, un écart type s et une moyenne estimée \bar{X} , on a :

$$[\bar{X} - t_{\alpha} \frac{s}{\sqrt{n}} .. \bar{X} + t_{\alpha} \frac{s}{\sqrt{n}}]$$

Avec :

$t_{\alpha} = 1$ ou 2 ou 3 pour un intervalle de confiance à 68% ou 95% ou à 99% respectivement.

Workloads : Caractérisation

Quelque soit le type de workload, on en extrait les **caractéristiques clés** pour représenter le contexte de l'étude.

Intérêt

- * Représenter/identifier le contexte de l'étude,
- * Générer des workloads synthétiques associés aux workloads réels, mais qu'on pourrait répéter à l'infini.

Implémentation

Se base sur des techniques statistiques comme: moyennes, facteurs de dispersions, histogrammes simples et multiples...etc

Techniques de caractérisation de workloads

Moyennes

Technique la plus simple : caractériser les charges avec **une seule valeur** qui résume toutes celles que peut prendre le paramètre étudié dans la charge et qui renseigne donc sur sa **tendance**.

Il existe **plusieurs moyennes** (arithmétique, géométrique, harmonique...etc) et l'usage de chacune est plus approprié selon le paramètre analysé.

Techniques de caractérisation de workloads

Moyennes

Soit (x_1, x_2, \dots, x_n) les valeurs que peut prendre le paramètre X dans la charge analysée.

Moyenne arithmétique

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$$

Moyenne géométrique

$$\bar{X} = (\prod_{i=1}^n x_i)^{1/n}$$

Techniques de caractérisation de workloads

Moyennes :

Moyenne harmonique

$$\bar{X} = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

Exemples:

Temps de réponse moyen (arith.), taux de cache-miss moyen (geo.) et vitesse moyenne (harm.).

Techniques de caractérisation de workloads

Moyenne pondérée

Une moyenne pondérée est une moyenne où chacune des différentes valeurs dans la charge considérée est associée à un coefficient qui détermine son poids.

Moyenne arithmétique pondérée :

$$\bar{X} = \frac{\sum_{i=1}^n p_i x_i}{\sum_{i=1}^n p_i}$$

P_i : non négatifs.

Techniques de caractérisation de workloads

Facteur de dispersion

La moyenne seule ne suffit pas, surtout lorsqu'il y a une grande dispersion (variabilité) dans les valeurs que peut prendre le paramètre.

La dispersion est estimée par la **variance**

$$\text{var}(X) = \sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

Techniques de caractérisation de workloads

Facteur de dispersion

On en déduit **l'écart type**

$$\sigma = \sqrt{\text{var}(X)}$$

On utilise souvent le **coefficient de variation** ou écart type relatif

$$\sigma / \bar{X}$$

Techniques de caractérisation de workloads

Remarque 1

Si l'**écart type** est **nul**, cela signifie que le **paramètre est constant**

Remarque 2

Si l'**écart type** est **élevé**, il est plus approprié d'organiser les valeurs du **paramètre en classes**.

Techniques de caractérisation de workloads

Quantiles : définition

Les **quantiles** sont des points à intervalles réguliers qui **divisent la liste ordonnée** en q sous-parties d'effectifs égaux.

Les quantiles les plus utilisés

- Les 100-quantiles ou centiles : diviser l'échantillon en 100
- Les 10-quantiles ou déciles : diviser l'échantillon en 10
- Les 4-quantiles ou quartiles : diviser l'échantillon en 4
- Les 2-quantiles ou médiane : diviser l'échantillon en 2

Techniques de caractérisation de workloads

Quantiles : calcul

Un **q-quantile** est la plus petite valeur dans l'échantillon considéré, pour laquelle au moins q éléments de l'échantillon sont en dessous de cette valeur.

Exemples :

- Le premier décile (10-quantile) est la plus petite valeur de l'échantillon dont 10% de ce même échantillon sont en dessous.
- Le troisième quartile (4-quantile) est la plus petite valeur de l'échantillon dont 75% de ce même échantillon sont en dessous.

Techniques de caractérisation de workloads

Exemple :

valeur	4	7	9	11	13	15	18
nbr occurrence	6	5	10	6	2	7	4

Le 1er quart-quantile = 7

⇒ 25% des valeurs sont en dessous

Le 3eme quart-quantile = 15

⇒ 75% des valeurs sont en dessous

Le 2-quantiles = 9

⇒ 50% des valeurs sont en dessous.

Techniques de caractérisation de workloads

L'écart inter-quartile

L'écart inter-quartile est la différence entre le premier et le troisième quartile.

Un écart inter-quartile important (resp. faible) indique une distribution hétérogène (resp. homogène).

Dans notre exemple précédent : l'écart inter-quartile est important (8), donc distribution hétérogène.

Techniques de caractérisation de workloads

Médiane : définition

Autre nom du 2-quantile, représente la valeur centrale d'un échantillon ordonné (ASC).

Elle n'est pas affectée par les valeurs aberrantes comme c'est le cas pour la moyenne.

Médiane : calcul

Si échantillon de $(2n + 1)$ termes, $\text{mediane} = u_{n+1}$

Si échantillon de $2n$ termes, $\text{mediane} = \frac{u_n + u_{n+1}}{2}$

Techniques de caractérisation de workloads

Exemple 1: Médiane, moyenne et mode

valeur	4	7	9	11	13	15	18
nbr occurrence	6	5	10	6	2	7	4

Moyenne pondérée : 10.45

Médiane : 9 la plus petite valeur auquel sont inférieures la moitié des valeurs.

Mode : 9 c'est la valeur la plus fréquente.

Techniques de caractérisation de workloads

Exemple 2: Médiane, moyenne et mode

valeur	32	41	48	58	60	97	124
nbr occurrence	15	10	15	15	15	2	6

Moyenne pondérée : 55.4 effet de valeur aberrante.

Médiane : 48 la plus petite valeur auquelle sont inférieures la moitié des valeurs.

Modes : 32, 48, 58, 60 c'est les valeurs les plus fréquentes.

Techniques de caractérisation de workloads

Historgrammes

Le type le plus utilisé est le **simple ou mono-paramètre**. Il met en évidence le nombre de valeurs similaires que peut prendre un paramètre.

L'historgramme est le lien entre les outils de **représentation des valeurs observées**/quantiles (avec tri, mixage de lots de mesures par exp.) et leur **description par une loi de probabilité**/distribution (loi normale par exp.)

Techniques de caractérisation de workloads

Exemple : Histogramme simple

valeur	3	6	9	10	11	14	16	18
nbr occurrence	1	3	5	6	5	3	1	1

Techniques de caractérisation de workloads

Historgrammes multiples

Utilisé lorsque plusieurs paramètres sont corrélés. Le plus souvent 2 à 3 paramètres.

Un "damier" avec des tours de différentes hauteurs en guise de pièces pour un histogramme à 2 paramètres.

Paramètres à générer

Les paramètres les plus utilisés pour caractériser les charges notamment en calcul haute performance sont :

- ① timing des charges
- ② adresses d'accès
- ③ nombre de sources des charges
- ④ temps d'exécution/service

et les valeurs qu'ils peuvent prendre sont générées en utilisant des lois de probabilités.

Génération : timing

Temps d'inter-arrivées

Les dates d'arrivées d'instructions/requêtes d'accès (caches, disques...)/tâches sont générées :

- A partir d'une **date de référence**. Cette date peut être nulle (amorce du système) ou correspondre à la date d'arrivée d'un événement.
- Selon une **fréquence** qui détermine le stress auquel est soumis le système à analyser.
- Selon un **profil** qui détermine le comportement des applications dédiées à ce système et par conséquent son utilisation.

Génération : timing

Taux d'arrivée ou **Fréquence/Débit**, noté λ , détermine le nombre d'instructions/requêtes/tâches... soumises au système en une unité de temps.

Utilisée dans la génération des temps inter-arrivées en combinaison avec le **profil d'utilisation** du système pour une meilleure représentativité des charges synthétiques.

- Utilisation fixe
- Utilisation ordinaire
- Utilisation en rafales
- Utilisations périodiques (ON/OFF), heavy tailed,...etc

Génération : timing

Utilisation fixe :

Les dates qui fixent le timing des charges sont séparées par un temps fixe δt

⇒ Les arrivées sont alors périodiques.

Pas besoin donc de loi de probabilité pour la génération des dates. Celle-ci se fait par un processus incrémental d'un pas de δt .

$$date_i = date_{ref} + i \times \delta t$$

Génération : timing

Utilisation ordinaire (standard):

Cas typique, les **arrivées** suivent une **loi de poisson** et les temps inter-arrivées une loi exponentielle avec un paramètre λ (la fréquence)

La génération des dates d'arrivées :

- 1 Générer une valeur à une variable aléatoire u suivant une **loi Uniforme** sur $[0..1]$,
- 2 Utiliser cette valeur pour calculer le temps inter-arrivée :
$$\delta t = -\frac{1}{\lambda} \ln(u)$$
- 3 La date de l'instruction/requête/tâche actuelle est obtenue par celle de la précédente incrémentée de ce δt

Génération : timing

Utilisation en rafales :

Les arrivées se produisent en rafales (*bursts*) selon une loi de poisson et avec des **tailles de rafales** selon une **loi géométrique** de paramètre p correspondant à l'inverse de la taille moyenne d'une rafale.

Pour la génération de la taille de la rafale :

- 1 Donner une valeur à une variable aléatoire u suivant une **loi Uniforme** sur $[0..1]$,
- 2 Utiliser cette valeur pour calculer la taille de la rafale

$$G(p) = \lceil \frac{\ln(u)}{\ln(1-p)} \rceil$$

Génération : addressing

Les adresses d'accès aux caches/disques et autres supports peuvent être **uniformément distribuées** sur la totalité de l'espace d'adressage comme elles peuvent montrer une certaine **localité spatiale** avec une concentration sur certaines zones (hot spots).

- **Uniforme** : la génération des adresses se fait selon **une loi uniforme**.
- **hot spots** : la génération des adresses se fait selon **une loi uniforme** avec des **facteurs de concentration** sur les zones composant les hot spots.

Génération : addressing

La génération se fait comme suit :

- **Uniforme** : génération de valeurs selon une **loi uniforme** dans l'intervalle $[0..1]$ puis l'adapter à l'espace d'adressage du système à analyser.
- **hot spots** : subdiviser l'espace en sous-ensembles, en déterminer les plus actifs d'entre eux, associer un facteur de concentration (a) à ces zones actives.

Remarque : un facteur a est associé à chaque zone. Pour les zones actives, $a > 1$ et $a = 1$ pour les autres.

Génération : nombre de sources

Les charges peuvent provenir d'un seul utilisateur/application (**charge à source unique**) ou bien de plusieurs (**charge à sources multiples**).

- **Charge à source unique** : voir précédemment.
- **Charge à sources multiples** : les temps inter-arrivées sont générés par une loi d'Erlang a k étages (k sources)

Remarque :

$$\text{Erlang}(1, \lambda) = \text{poisson}(\lambda)$$

Génération : nombre de sources multiples

Génération de charge à k sources ($k > 1$) :

Les temps inter-arrivées suivent une loi **Erlang** à k étages avec k le nombre de sources à représenter.

- Donner une valeur à une variable aléatoire u suivant une **loi Uniforme** sur $[0..1]$,
- Utiliser cette valeur et celle du **taux d'arrivée** λ pour calculer le temps inter-arrivée par :

$$t = -\left(\frac{1}{\lambda k}\right) \times \ln(\prod_{i=1}^k u_i)$$

Workloads réels

Les charges réelles peuvent provenir de deux sources :

- Collecte au cours d'exécution d'une **application-utilisateur** sur le système observé.
- Collecte au cours d'exécution de **benchmarks** sur le système observé.

Workloads réels : benchmarks

Définition

Une portion d'un code utilisateur qui caractérise le mieux le comportement de cette application.

Exemples

Calcul : les benchmarks SPEC, Linpack , Livermore, NAS

Entrées/Sorties : Bonnie, IOzone

- Il s'agit d'exécuter l'application utilisateur à analyser ou celle représentant la classe à analyser.
- Collecter les charges générées pendant cette exécution.
- Les re-utiliser éventuellement dans des contextes d'exécution différents dans le futur ou bien,
- En déduire les principales caractéristiques et les représenter par des lois de distributions pour multiplier les cas d'analyse possibles.

Bien cibler **quoi mesurer** avant d'exécuter l'application utilisateur et préparer les **outils d'extractions des charges** qui en résultent.

Pour plus d'infos :
Voir Cours Mesure dans quelques semaines

Workloads réels : paramètres

Que la collecte concerne un benchmark ou une application-utilisateur:

Elle se concentre sur **les événements qui changent l'état du système**:

- changement de contexte,
- Seek (disque),
- réception ou envoi de messages...etc

Avec leurs datations, leurs nombres en une unité de temps, leurs tailles...etc