

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

Soraya Zertal

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

Cours 4 : La simulation et la mesure

Simulation

But

L'opportunité d'**observer le comportement d'un système**

- quelconque, même **inexistant** au moment de l'évaluation,
- dans n'importe quelles **conditions** (configurations),
représentant des **contextes d'applications divers**.

Intérêts

- Pas besoin de construire un système réel.
- Réalisation de **tests contrôlables** et répétitifs.
- Pas de limites pour les **scénarios de tests**.
- Possibilité de **reproduire les résultats** par les autres utilisateurs.

Simulation

Exemples d'utilisation

La simulation permet d'**apporter des réponses** aux questions telles que:

- Quel algorithme de placement est meilleur pour telle application sur une architecture donnée ?
- Quelle stratégie de cache est meilleure pour telle application ?
- Quelles sont les propriétés en termes d'équité et de débit associées à certaines stratégies de gestion de ressources partagées ?
-

Simulation

Simulation Vs. Mesure

La **mesure**.... oui pour la **précision** et la **représentativité** du **système réel** mais....

- Quelle est la part réelle des dysfonctionnements/surcharge du système dans les résultats ?
- Les expériences et donc leurs résultats sont limités à la plateforme analysée.
- Les extrapolations sont possibles mais très peu convaincantes.
- Les résultats sont difficilement reproductibles par d'autres utilisateurs.

Simulation : Définition et classification

Définition

Tentative de prédire différents aspects du comportement d'un système quelconque en créant une imitation.

Classification

Il existe différents types de simulation :

- Emulation
- Simulation statique
- Simulation à événements discrets

Emulation et simulation statique

Emulation

Imitation du comportement physique d'un système par un logiciel s'exécutant sur un autre, de manière à ce qu'il apparait comme le système imité.

Remarque : l'émulation produit des résultats assez précis mais génère une **surcharge (overhead) importante** en contre partie.

Simulation statique

Appelée surtout méthode **Monte Carlo**, est une simulation indépendante du temps et est exécutée jusqu'à l'atteinte d'un certain état (état d'équilibre). Voir plus loin dans ce cours.

Simulation à événements discrets

La simulation à événements discrets

Une représentation d'un système par des **événements**, des **transitions** qui leur sont associées et des **fonctions spécifiques** à exécuter à chaque fois qu'un de ces événements survient et qu'on effectue la transition associée. C'est donc une imitation du schéma fonctionnel, avec un changement d'état selon le temps et les événements externes ou générés par la simulation.

Simulation distribuée

Liée plutôt à la manière la simulation est exécutée : **distribuée**, exploitant les ressources de **différents sites**, collaborant et communiquant tout en respectant le schéma Workflow établi et les principes de causalité.

Simulation : Exemples

Simulation d'un CPU

Trois simulations possibles avec un niveau d'abstraction décroissant :

- **Simulation mathématique** : Flux macroscopique des tâches et composants (pipe) avec une simulation mathématique à gros grain.
- **Simulation à événements discrets** : Simulation au niveau cycle (niveau microscopique) au profit de la précision.
- **Emulation** : Virtualisation via un autre CPU, càd construction d'une machine virtuelle.

Simulation : Exemples

Simulation d'un réseau de communication

Trois simulations possibles avec un niveau d'abstraction décroissant :

- **Simulation mathématique** : Flux macroscopique des tâches et composants du réseaux (pipe) avec une simulation mathématique à gros grain.
- **Simulation à événements discrets** : Simulation au niveau du paquet et suivi microscopique à travers les différents composants du réseau.
- **Emulation** : de l'actuel flux à travers un réseau.

Simulation à événements discrets

Terminologie

- **Variables d'état** définissant l'état du système et dont les valeurs sauvegardées permettent à une simulation arrêtée, de redémarrer ultérieurement (taille de la file d'attente, avancement dans la consommation des tâches,...).
- **Événement** Un changement dans l'état du système (arrivée ou fin d'un job, panne d'un composant,...)
- **Temps continu Vs temps discret** indiquant si l'état du système est défini **tout le temps** ou pas.
- **Etat continu Vs état discret** indiquant si les variables du système sont continues (temps) ou discrets (nombre d'occurrence, taille).

Simulation à événements discrets

Structure

Un simulateur à événements discrets doit comporter les modules suivants:

- 1 un **ordonnanceur** d'événements
- 2 une **horloge globale** et une méthode pour la mise à jour du temps
- 3 des **procédures de traitement** des événements
- 4 des **mécanismes de génération** d'événements
- 5 des **procédures d'enregistrement**/filtrage des données

Simulation à événements discrets

L'ordonnanceur des événements

L'ordonnanceur constitue le coeur du simulateur à événements discrets et a pour fonction de:

- Maintenir la **liste des événements** (non traités) dans un ordre temporel global.
- Procéder au **traitement de l'évènement approprié** de la liste par sa suppression de celle-ci et de le véhiculer à la procédure de traitement qui lui est associée.
- Insérer les **nouveaux événements** en respectant l'ordre temporel de la liste, ainsi qu'assurer les autres tâches de l'ordonnancement ou de la suppression d'événements.
- Assurer la **coordination avec l'horloge** globale.

Simulation à événements discrets

L'horloge globale

La référence temporelle de la simulation, mise à jour par l'ordonnanceur selon deux approches:

① Incrémentation par pas fixe

L'incrémentation se fait par l'ordonnanceur avec un pas fixe, qui ensuite vérifie les événements de la liste et lance ceux prêts. A la fin des exécutions, l'ordonnanceur re-incrémente l'horloge.

② Incrémentation événementielle

L'horloge change de manière non-uniforme et prend la valeur de la date de l'événement en tête de liste. C'est l'approche la plus utilisée.

Simulation à événements discrets

Traitement des événements

Chaque type d'événement est associé à **une procédure de traitement** qui simule le comportement du système lorsque ledit événement se produit.

Chaque procédure de traitement peut **changer l'état** du système et **générer d'autres événements**, qui se rajouteront à la liste tenue par l'ordonnanceur.

Le traitement implémenté dépend de la nature du système simulé.

Simulation à événements discrets

Génération des événements

Les générateurs d'événements se classifient par rapport à la **technique de génération utilisée**. Deux classes principales se distinguent:

- ① Classe orientée **trace** : séquence d'événements observés sur un système réel. Représentativité mais très grande Taille, point unique de validation et introduction de biais par la collecte.
- ② Classe orientée **loi de probabilité** : Similaire dans la forme mais générée par le simulateur selon des lois de probabilité (au début ou accompagnant la simulation).

Simulation à événements discrets

Enregistrement des données

Maintenir les variables d'états à savoir les compteurs d'événements et les variables de mesure du temps pour effectuer des statistiques et résumer les résultats.

Structures

- Listes ordonnées ou doublement chaînées
- Arbres

Se sont les structures les plus utilisées, pour accélérer les opérations (insertion, interrogation, modification) même si les accès en mémoire ne sont pas uniformes.

Simulation à événements discrets

Validation

La validation détermine le **degré de précision du modèle fonctionnel** en répondant aux questions:

- Le modèle capture-t-il les fonctionnalités essentielles du système ?
- Les suppositions sont-elles raisonnables ?
- Les données en entrée sont-elles représentatives des cas réels ?
- Les résultats obtenus sont-ils raisonnables ?
- Ces résultats sont-ils explicables ?

Simulation à événements discrets

Remarques ...

...concernant la construction du simulateur :

- Suivre de bonnes pratiques : programmation structurée, design modulaire, documentation, vérification d'erreurs, ...etc
- Exécution étape par étape pour vérifier le passage par TOUS les états du système simulé
- Vérification de la cohérence et la stabilité du simulateur
- Mettre l'accent sur la vérification des cas spéciaux (corner conditions)

Simulateurs populaires

Simulateur de Microprocesseur

Multitude de simulateurs standards au niveau cycle :

<http://pages.cs.wisc.edu/arch/www/tools.html>

Simulateur de système de stockage

Le pionnier, développé/maintenu au Parallel Data Lab (CMU).

<http://www.pdl.cmu.edu/DiskSim/>

SimGrid

Outils pour les environnements distribués et hétérogènes (réseau de stations aux grilles de calcul), développé par l'**inria**.

<http://simgrid.gforge.inria.fr/>

Monte Carlo

Historique

N. Metropolis et *S. Ulam* [1947-1949], utilisant un **procédé aléatoire** pour le **calcul d'une valeur numérique**. Faisant donc allusion aux jeux de hasard pratiqués à **Monte Carlo**.

Définition

Méthode pour l'évaluation **itérative** d'un **modèle déterministe** en utilisant des ensembles de **variables aléatoires** en entrée.

Un modèle **déterministe** donne les mêmes résultats autant de fois qu'il est recalculé dans les mêmes conditions, avec les mêmes entrées.

Monte Carlo

Applications

Le domaine d'applications de la méthode Monte Carlo englobe : le calcul des intégrales multidimensionnelles (surface, volume), l'évaluation du risque financier, de la rentabilité ou la solvabilité dans le secteur bancaire et d'assurance, l'étude de la dynamique des molécule ou des mouvements des populations, ...etc

Remarque

La méthode Monte Carlo nécessite un **très grand nombre d'évaluations**, tâche facilitée par le **calcul parallèle**

Monte Carlo

Monte Carlo en étapes :

Les différentes étapes suivies pour effectuer une simulation Monte Carlo sont :

- 1 **Créer un modèle pour les paramètres** (X_1, X_2, \dots, X_k)
- 2 **Générer un ensemble de valeurs aléatoires**
 $(X_{i1}, X_{i2}, \dots, X_{ik})$
- 3 **Evaluer le modèle** et ranger le résultat dans Y_i
- 4 **Répéter** les étapes 2 et 3 pour $i = 1..n$ n très grand pour utiliser la loi des grands nombres.
- 5 **Analyser les résultats** en utilisant des histogrammes, moyennes et variances, intervalles de confiance...etc

Monte Carlo : Exemple

Calcul de π

Déterminer π en calculant la surface d'un cercle de rayon $= 1$.

- 1 Le **modèle** : $Surf(cercle) = \pi R^2 = \pi$ car $R=1$
La **surface du cercle incluse dans le carré** $= \frac{\pi}{4}$
C'est l'ensemble de points $X_i(x_{i1}, x_{i2})$ dans cet espace.
- 2 **Générer un couple** (x_{i1}, x_{i2}) , deux v.a. uniformément distribuées sur $[0..1]$, représentant un point sur le carré.
- 3 **Evaluer le modèle** : incrémenter le nombre de points NB du carré **ET** du cercle ssi: $\sqrt{(x_{i1}^2 + x_{i2}^2)} < 1$
- 4 Refaire 2 et 3, N (très grand) fois. $\frac{\pi}{4}$ estimé par $\frac{NB}{N}$

Mesure : Introduction

Quoi mesurer?

Le **nombre d'occurrence** d'événements, la **durée** entre deux événements, la **taille** pour contenir une information, ...

Définition

La **mesure** sert à suivre le fonctionnement du système par :

- ❶ **L'observation et la collecte** : effectuer des prises de mesure et rassembler les informations qui en découlent.
- ❷ **L'analyse** : traiter les données collectées avec des techniques statistiques
- ❸ **La présentation** : afficher les résultats sous les diverses formes appropriées.

Mesure : terminologie

Quelques termes usuels

- **Événement** : tout changement de l'état du système (ex. arrivée d'une requête/un paquet, un accès mémoire ...etc)
- **Trace** : liste ou journal d'événements
- **Surcharge** : (overhead) mesure indiquant une **consommation additionnelle** du système
- **Résolution** : niveau de détail de l'observation
- **Taille en entrée** : nombre de bits pour coder l'évènement
- **Portabilité** : quantité de modifications à effectuer afin d'implémenter l'outil de mesure.

Phase d'observation/collecte

Mécanismes d'observation

- 1 **"Espionnage implicite"** (Implicit spying):
Observer le système sans interférer avec son fonctionnement. Pas d'impact sur sa performance mais utilisation de filtres pour garder l'information utile.
- 2 **Instrumentation explicite** :
Incorporer des traceurs, des sondes en utilisant des compteurs matériels par exemple.

Classes d'observation/collecte

- **On-line** : résultats analysés pendant le monitoring.
- **batch** : données collectées mais l'analyse est reportée.

Phase d'observation/collecte

Stratégies de collecte

- **Event-driven** : l'information est enregistrée à chaque apparition d'un événement
- **Trace** : il y a l'enregistrement de l'état du système en plus. Donc surcharge en temps et en espace.
- **Echantillonnage** L'enregistrement de l'information se fait de manière **périodique**, donc réajustement de la surcharge.
- **Mesure indirecte** Utilisable lorsque le métrique à mesurer ne peut l'être directement. Déterminer un autre métrique mesurable et en déduire/dériver le métrique d'intérêt.

Outils simples de collecte

Exemples :

En plus de matériels de mesure et logiciels spécifiques, il y a des outils simples, MAIS efficaces, utilisés **souvent** pour la mesure.

On cite :

time, getimeoftheday (mesure du temps au niveau μs), top, gprof etc

⇒ **Sujet du dernier cours**

Phase d'analyse

But de toute évaluation, dépend du contexte et des métriques. Plus importante pour la mesure à cause de la grande taille des données collectées, de la richesse du détail et du bruit.

ANalysis Of Variance (ANOVA)

Lors des mesures sur un système, les données collectées présentent des écarts, pouvant être importants, liés juste aux re-itérations des mesures ou présentant des réelles erreurs de mesure.

La méthode statistique ANOVA détermine la part d'erreurs dans les données collectées.

Phase d'analyse

ANOVA - Calcul

On considère un échantillon de n mesures collectées pour chacune des k exécutions (alternatives) et l'ensemble compose les y_{ij} pour la i ème mesure de la j ème exécution.

Calcul de la moyenne :

- **Par exécution :** $\bar{y}_j = \frac{\sum_{i=1}^n y_{ij}}{n}$
- **Générale :** $\bar{y} = \frac{\sum_{j=1}^k \sum_{i=1}^n y_{ij}}{k \times n}$

La mesure se réécrit $y_{ij} = \bar{y}_j + e_{ij}$ avec e_{ij} l'erreur de mesure.

Phase d'analyse

ANOVA - Calcul

La moyenne des mesures par alternative se réécrit :

$\bar{y}_j = \bar{y} + \alpha_j$ avec α_j l'effet d'exécution et $\sum_{j=1}^k \alpha_j = 0$

ANOVA - Application

La variation liée aux exécutions par la **Sum of the Squares of Alternatives (SSA)**

$$SSA = n \sum_{j=1}^k (\bar{y}_j - \bar{y})^2$$

La variation liée aux erreurs de mesures par la **Sum of Squares of Errors (SSE)**

$$SSE = \sum_{j=1}^k \sum_{i=1}^n (y_{ij} - \bar{y}_j)^2$$

Phase d'analyse

ANOVA et F-test

F-test : **test de Fisher** sert à tester l'égalité de deux variances en calculant le rapport entre les variances de SSA et SSE, en vérifiant qu'il ne dépasse pas une valeur théorique de la table de Fisher.

Le rapport entre les variances de SSA et SSE se calcule par :

$$S_a^2 = \frac{SSA}{k-1} \quad S_c^2 = \frac{SSE}{k(n-1)} \quad F = \frac{S_a^2}{S_c^2}$$

Si $F_{[1-\alpha; (k-1), k(n-1)]} > F_{calculé}$, la variation liée aux alternatives est statistiquement significative et est supérieure à celle liée aux erreurs à un niveau de confiance de $1 - \alpha$.

Phase de présentation

Paramètres du choix de la présentation

Choix lié au système mais trois paramètres à considérer :

- ① La fréquence de la présentation (on-line)
- ② La résentation hiérarchique
- ③ Les points importants: seuils, événement anormal ...etc

Outils simples, populaires et représentatifs

Un ensemble de techniques statistiques simples : moyenne(s), variance et écart type, quantilesetc (voir cours Workloads) avec les intervalles de confiance.

Qualité de mesure

Définition

Les **erreurs ou bruit** dans les mesures, doivent être **quantifiés** afin de déterminer **la qualité de ces mesures** par :

- 1 **Justesse** (Accuracy) : indique le rapprochement de la valeur mesurée de la valeur standard.
- 2 **Précision** : est relative aux mesures répétées (mêmes outils et conditions), souvent représentée par son inverse (l'imprécision).
- 3 **Résolution** est le plus petit changement détectable et "affichable".

Qualité de mesure

Sources d'erreurs

Instruments de mesure, temps partagé entre utilisateurs, événements temps-réel (exp. interruptions avec les interfaces réseaux), synchronisation des horloges, événements non déterministes (cache misses, les exceptions système, les défauts de pages...etc)

Quantification

- 1 **Justesse** : difficile à quantifier
- 2 **Précision et Résolution** : liées aux erreurs aléatoires, peuvent être quantifiées par des mesures répétées en utilisant les **intervalles de confiance**

Qualité de mesure

Intervalles de confiance pour la moyenne : grands échantillons

- échantillon de ($n \geq 30$)
- approximation de la distribution des erreurs par **une loi Normale** avec une moyenne \bar{X} et un écart type s
- α est le niveau de signification, généralement 0.9 ou 0.95
- L'intervalle de confiance $[c_1..c_2]$ pour la moyenne \bar{X} avec

$$c_1 = \bar{X} - Z_{1-\alpha/2} \frac{s}{\sqrt{n}}$$

$$c_2 = \bar{X} + Z_{1-\alpha/2} \frac{s}{\sqrt{n}}$$

Les valeurs $Z_{1-\alpha/2}$ disponibles sur la table de la loi Normale centrée réduite.

Qualité de mesure

Intervalles de confiance pour la moyenne : échantillons réduits

- échantillon de ($n < 30$)
- la variance calculée s^2 peut montrer de grandes différences entre une série de mesure et une autre.
- L'intervalle de confiance $[c_1..c_2]$ pour la moyenne \bar{X} avec
$$c_1 = \bar{X} - t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}}$$
$$c_2 = \bar{X} + t_{1-\alpha/2;n-1} \frac{s}{\sqrt{n}}$$
Avec $t_{1-\alpha/2;n-1}$ la valeur de la distribution t pour $(n - 1)$ degré de liberté.

Quantification des erreurs

Exemple

Le temps mesuré d'écriture d'un fichier sur un disque 8 fois :

Num mesure	1	2	3	4	5	6	7	8
Tps d'ecr (X_i)	8.0 s	7.0 s	5.0 s	9.0 s	9.5 s	11.3 s	5.2s	8.5 s

$$\bar{X} = 7.94 \text{ s et } s = \sqrt{\frac{\sum_{i=1}^8 (X_i - \bar{X})^2}{(n-1)}} = 2.14$$

Calcul de l'intervalle de confiance pour quantifier la precision associée à ces mesures

Une intervalle de confiance de 90%, assure qu'il y a 90% de chance que le temps moyen soit dans $[c_1..c_2]$ avec :

$\alpha = 0.1$ et selon la table $t_{(1-\alpha/2, n-1)} = t_{0.95, 7} = 1.895$

$$c_1 = 7.95 - \frac{1.895 \times 2.14}{\sqrt{8}} = 6.5 \text{ et } c_2 = 7.95 + \frac{1.895 \times 2.14}{\sqrt{8}} = 9.4$$

Quantification des erreurs

Nombre d'alternatives de la mesure

Combien de fois il faut répéter la prise de mesure ? en assurant que les résultats soient dans une certaine intervalle de confiance avec un nombre minimal n de mesures effectuées.

$$n = \left(\frac{s \times Z_{1-(\alpha/2)}}{e \times \bar{X}} \right)^2$$

$e = c_2 - c_1$ l'étendue de l'intervalle de confiance
 α le niveau de signification.

Quantification des erreurs

Exemple

On considère l'exemple précédent et on voudrait calculer combien de mesures doit-on effectuer pour assurer à 90% que les temps d'écriture obtenus soient à 3.5% ($e = 0.07$) de la valeur moyenne estimée \bar{X} .

$$\left(\frac{s \times Z_{1-\alpha/2}}{e \times \bar{X}} \right)^2$$

$$\left(\frac{2.14 \times 1.895}{0.07 \times 7.94} \right)^2 = 212.95$$

⇒ il faut 213 mesures (écritures de ce fichier sur le disque)

Conception d'un scénario de mesure

But

- Minimiser le nombre d'expériences en maximisant l'information collectée.
- Isoler l'effet de chaque facteur en entrée
- Déterminer les effets dûs aux interactions entre les facteurs
- Quantifier la part de l'erreur dans les résultats

Conception d'un scénario de mesure

Plan d'expérience : définition

La répartition entre les expériences du nombre de facteurs et du nombre de niveaux (valeurs possibles) des facteurs.

Plan d'expérience : classification

- ❶ Plan simple : d'une configuration initiale, on varie un facteur (optimisation locale). Pas d'analyse d'interactions.
- ❷ Plan factoriel complet : considérer toutes les configurations (combinaisons). Impraticable car nombre Très grand.
- ❸ Plan fractionnaire : un sous ensemble des combinaisons. Réalisable, mais comment choisir les combinaisons ?

Conception d'un scénario de mesure

Plan d'expérience : Exemple

4 facteurs {A,B,C,D} avec 3 niveaux/facteur

Le plan factoriel complet donne 81 combinaisons.

Il existe d'autres plans, le plus "classique" est le plan en étoile.

Expérience	Fact1	Fact2	Fact3	Fact4
1	A1	B1	C1	D1
2	A1	B2	C2	D2
3	A1	B3	C3	D3
4	A2	B1	C2	D3
5	A2	B2	C3	D1
6	A2	B3	C1	D2
7	A3	B1	C3	D2
8	A3	B2	C1	D3
9	A3	B3	C2	D1