RICM4 : Évaluation de Performance

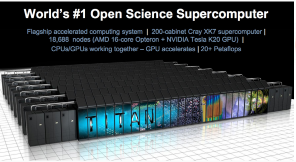
Arnaud Legrand (CNRS/Univ. Grenoble Alpes – LIG)

6 Décembre 2018

Digital infrastructures

Notre société (citoyens, compagnies, scientifiques, …) repose sur de gigantesque infrastructures digitales

- HPC/cloud/…
- Wireless networks
- Smart grids
- Transportation systems



Comment modéliser/concevoir/optimiser ces monstres ?

- Évaluation de performance
- Dimensionnement
- Tolérance aux pannes
- Consommation énergétique

L'évaluation de performances, à quoi ça sert ?

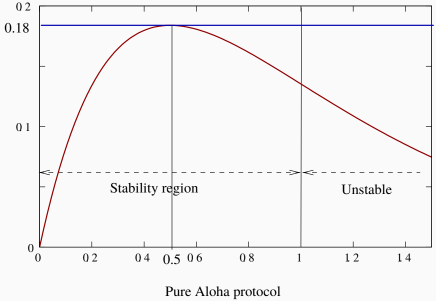
1. De quel débit fixe a-t-on besoin sur un réseau étendu pour garantir un temps de réponse inférieur à 100ms ?
2. Prouver que votre algorithme est plus efficace que celui actuellement utilisé.
   - Sous quelles conditions ?
3. Quels sont les goulots d'étranglement des réseaux 802.11p ?
4. Une application parallèle est capable de traiter 40 clients par secondes sur un cluster de 10 machines. Pour diviser ce temps par deux, faut-il doubler le nombre de machines ?

Compétences

- Prédire le comportement d’un système
- Optimiser un système en conception ou existant

Une science, un art ?

- Lies, bloody lies…
- simulation caveats
- “voodoo” constants



Mais c'est quoi l'éval de perf ?

Trois grands domaines :

Measurements	Simulation	Queuing Theory
• Experimental design • Statistics	• Discrete-event simulation	• Stochastic Process

Mais c'est quoi l'éval de perf ?

Trois grands domaines :

Measurements	Simulation	Queuing Theory
• Experimental design • Statistics	• Discrete-event simulation	• Stochastic Process
	Un compromis	
🙂 Réaliste	🙁 Modèle	🙁 Hypothèses restrictives
🙂 Convaincant	🙁 Bugs	
🙁 Espace monstrueux	🙂 Contrôlé	🙁 Intractable
🙁 Comparaison	🙂 *What if...*	🙂 Forme analytique
🙁 Chronophage	🙁 Long	🙂 Très rapide
🙁 $$$	🙁 $$	🙂 $

Il faut les comprendre pour les utiliser à bon escient

Toujours confronter l'une avec l'autre : (in)-validation

Réflexions sur et pratique de

- la mesure
- la modélisation
- la simulation

Des outils théoriques

1. Chaînes de Markov à temps discret
2. Modèles de trafic
3. Chaînes de Markov à temps continu
4. Files d'attente classiques
5. Réseaux de files d'attente ?

Compétences visées

- Modéliser un problème d'évaluation de performance
- Concevoir et développer un outil qui produit des indicateurs de performance
- Analyser les résultats obtenus pour prédire un comportement ou optimiser un système
- Savoir critiquer une étude de performance

Team

- Arnaud Legrand, Florence Perronnin, and...
- yourself ☺

Modalités

Pré-requis

- Probabilités et Simulation ☺
- Maths : produit matriciel, limites, dérivées/primitives usuelles

Attitude

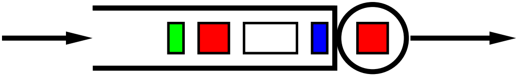
- Poser des questions (que vous ayez compris ou non...)
- Curiosité
- Esprit critique

Évaluation

- CC = Quicks + Projet (15-20h min) : 50%
- Examen : 50% (coefficients à confirmer)

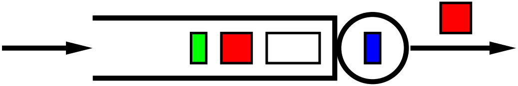
Queues

Queues are among simplest dynamic systems, but are still the source of many open problems.
Tasks do not have any constraints, sizes and arrival times are often independent.



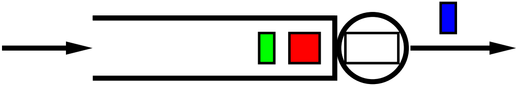
Queues

Queues are among simplest dynamic systems, but are still the source of many open problems.
Tasks do not have any constraints, sizes and arrival times are often independent.



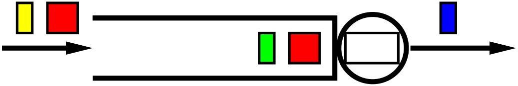
Queues

Queues are among simplest dynamic systems, but are still the source of many open problems.
Tasks do not have any constraints, sizes and arrival times are often independent.



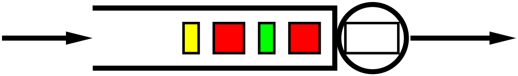
Queues

Queues are among simplest dynamic systems, but are still the source of many open problems.
Tasks do not have any constraints, sizes and arrival times are often independent.



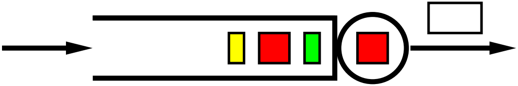
Queues

Queues are among simplest dynamic systems, but are still the source of many open problems.
Tasks do not have any constraints, sizes and arrival times are often independent.



Queues

Queues are among simplest dynamic systems, but are still the source of many open problems.
Tasks do not have any constraints, sizes and arrival times are often independent.



Quizz from Harchol-Balter(2013)

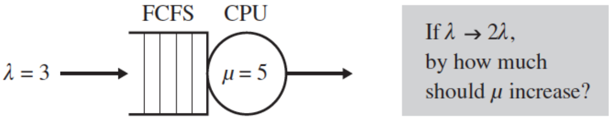


Figure 1.2. A system with a single CPU that serves jobs in FCFS order.

Answers

(a) Double the CPU speed

(b) More than Double the CPU speed

(c) Less than double the CPU speed

Performance Modeling and Design of Computer Systems: Queueing Theory in Action Mor Harchol-Balter, Cambridge University Press, 2013

Quizz (2)

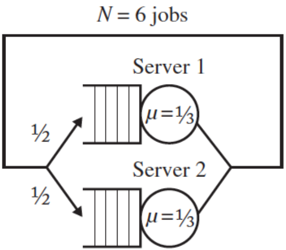


Figure 1.3. A closed batch system.

Question You replace server 1 with a server that is twice as fast (the new server services jobs at an average rate of 2 jobs every 3 seconds). Does this "improvement" affect the average response time in the system? Does it affect the throughput?

Quizz (3)

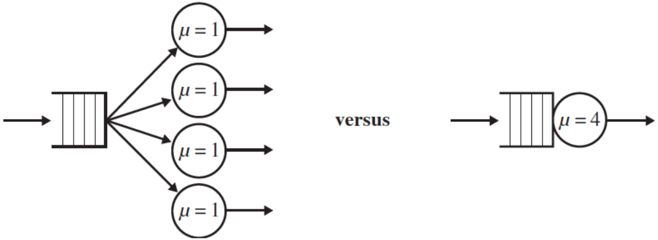


Figure 1.5. Which is better for minimizing mean response time: many slow servers or one fast server?

Quizz (4)

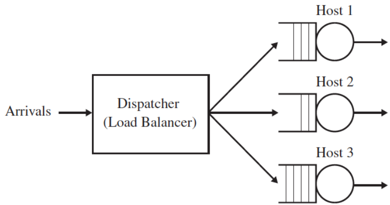


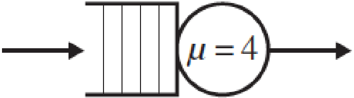
Figure 1.6. A distributed server system with a central dispatcher.

Task assignment policies

- **Random**
- **Round-Robin**
- **Shortest-Queue**
- **Size-Interval-Task-Assignment (SITA)**
- **Least-Work-Left (LWL)**
- **Central-Queue**

Question: Which of these task assignment policies yields the lowest mean response time?

Quizz (4)

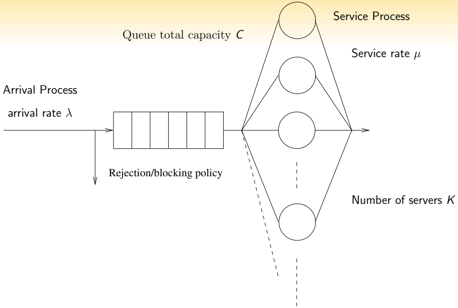


Scheduling strategy

- **Random**
- **Non-Preemptive Last-Come-First-Served (LCFS)**
- **First-Come-First-Served (FCFS)**

Question: Which of these non-preemptive service orders will result in the lowest mean response time?

Kendall's notation



Notation : $A/S/K/C/Disc$

- A : arrival process
- B : service process
- K : number of servers
- C : total queue capacity (including currently served customers)
- *Disc* : Service discipline (FIFO, LIFO, PS, Quantum, Priorities,...)

Process types

Arrival or service process

- M : memoryless (exponential distribution);
- D : deterministic (constant);
- U : uniform distribution;
- Erl_k : Erlang distribution (sum of exponentially distributed RV, $\gamma(k, \lambda)$);
- H_k : Hyper-exponential distribution
- GI : general independent (given arbitrary distribution) independence between inter arrivals or services
- G : general

Usually service times and inter arrival processes are independent

Service discipline

- *FIFO* : First In First Out
- *LIFO* : Last In First Out
  - pre-emptive or non pre-emptive
  - resume, restart with same service, restart with new service
- Quantum : round-robin policy
  - PS : asymptotic
- Priority
  - pre-emptive or non pre-emptive
  - resume, restart with same service, restart with new service
- Adaptive discipline

David George Kendall

Born: 15 January 1918 in Ripon, Yorkshire, England
Died: 23 October 2007 in Cambridge, England
One highlight was his pioneering work of 1949 on stochastic (or random) processes for population growth. Another was his classic 1951 paper on queuing theory, which was motivated by the scheduling problems of aircraft and runways during the Berlin air lift of 1948-49. A third was a series of penetrating studies, with G.E.H. Reuter, of Markov processes (roughly, random processes without memory).
The Independent/MacTutor History
Another biography by G. Grimmet

System variables

User variables

- Input rate λ or inter-arrival δ
- Service time σ or S (service rate μ)
- Waiting time W
- Response time R (in some books W)
- Rejection probability

Resource variables

- Resource utilisation (offered load) ρ
- Queue occupation N
- System availability