A Mean field model of work stealing in Large heterogeneous platforms

Nicolas Gast and Bruno Gaujal

INRIA Grenoble

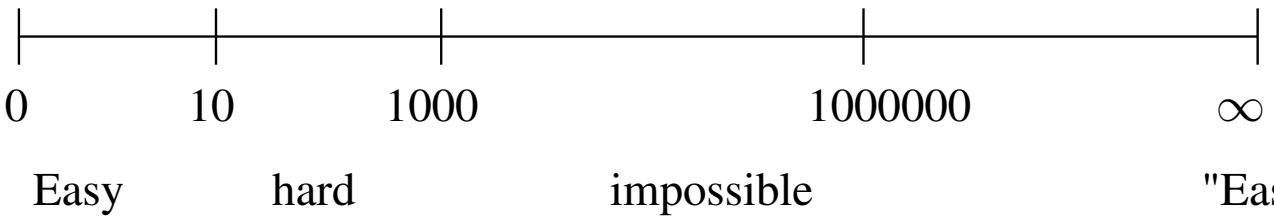
Aussois, June 2010

Introduction

System made of N objects (processors, tasks, buffers,...).
We want to compute the behavior of this system. This computation has a complexity $C(N)$ that grows with N (linear, polynomial, exponential).

Introduction

System made of N objects (processors, tasks, buffers,...).
We want to compute the behavior of this system. This computation has a complexity $C(N)$ that grows with N (linear, polynomial, exponential).



By changing our point of view (from microscopic to macroscopic)

Work stealing (WS)

Work stealing principle: idle processors steal work from busy ones.
A task (made of several jobs) is distributed over N processors. When one processor becomes idle, it chooses a victim processor (according to some rule, TBD), and steals a fraction (typically half) of its remaining work.
Some properties:

- Near optimal for the makespan (in the homogeneous case)

$$M = \frac{W}{N} + O(\gamma^{-1} C).$$

- Processor oblivious and stable.
- Application oblivious.
- Works well in practice.
- Implemented in many libraries (Cilk, TBB, Kaapi).

Goal of this work

Study the behavior of WS for parallel tasks, when used in large heterogeneous plateforms (grids), in the stationary regime.
In particular, we want to build a high level model of WS and optimize the key parameters:

- choose the best processor to steal from (tradeoff between communication speed and load balancing).
- choose the best steal fraction (tradeoff between local and global balance).
- distribute the incoming work among processors.

Mean field

Mean field principle: The behavior of a complex system made of N objects becomes simpler when N goes to infinity.

$O_1, \dots O_N$ objects in $\mathcal{S}$ (of size S).
The empirical measure of the system, for each state $s \in \mathcal{S}$, is

$$X_s^N = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{O_n = s}$$

Main feature: the dynamics of the system is invariant by permutation of the objects:

$$(O_1(k), \dots, O_N(k)) =_{db} \left(O_{\sigma(1)}(k), \dots, O_{\sigma(N)}(k)\right).$$

Under this invariance, $X^N(k) = (X_1^N(k), \dots, X_S^N(k))$ has the Markov property:
The value of $X^N(k+1)$ only depends on the value of $X^N(k)$.

Mean field (II)

The drift of the system at measure $x \in \mathcal{P}(\mathcal{S})$ is

$$F^N(x) \stackrel{\text{def}}{=} \mathbb{E}(X^N(k+1) - X^N(k) | X^N(k) = x).$$

The drift dictates the limiting behavior.
If $F^N \to 0$, the intensity $I(N)$ is its speed to 0:

$$f(\cdot) \stackrel{\text{def}}{=} \lim_{N \to \infty} F^N(\cdot)/I(N).$$

If F^N does not go to 0, its limit is $F(\cdot) \stackrel{\text{def}}{=} \lim_N F^N(\cdot)$.
In the first case, the mean field limit is the ODE $dx(t)/dt = f(x)$.
In the second case, the mean field limit remains in discrete time:
$x(k+1) = x(k) + F(x)$.

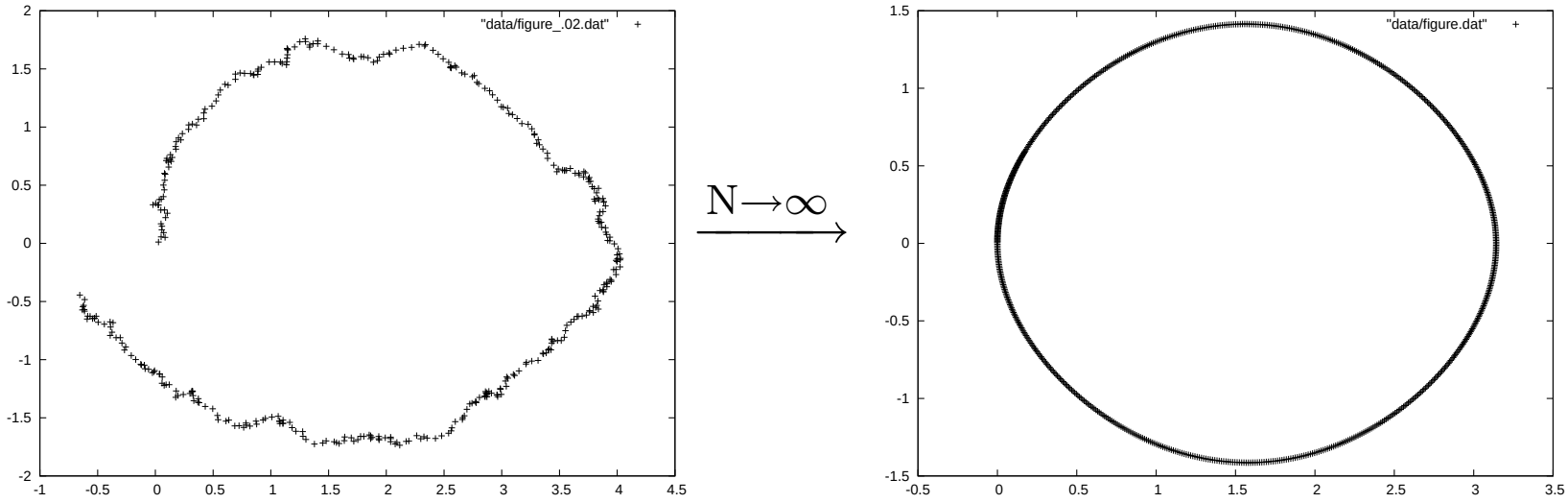
Theorem (Kurtz 86, Benaim 98)

case 1 $X^N(\lfloor tI(N) \rfloor) \to x(t)$ *uniformly on* $[0, T]$.

case 2 $X^N(k) \to x(k)$, *uniformly on* $[0, K]$.

In both cases, if the mean field has a unique attractor x^*, *the stationary measure* π^N *of* X^N *converges to the Dirac measure* δ_{x^*}.

Mean field (III)



Mean Field: A simple example

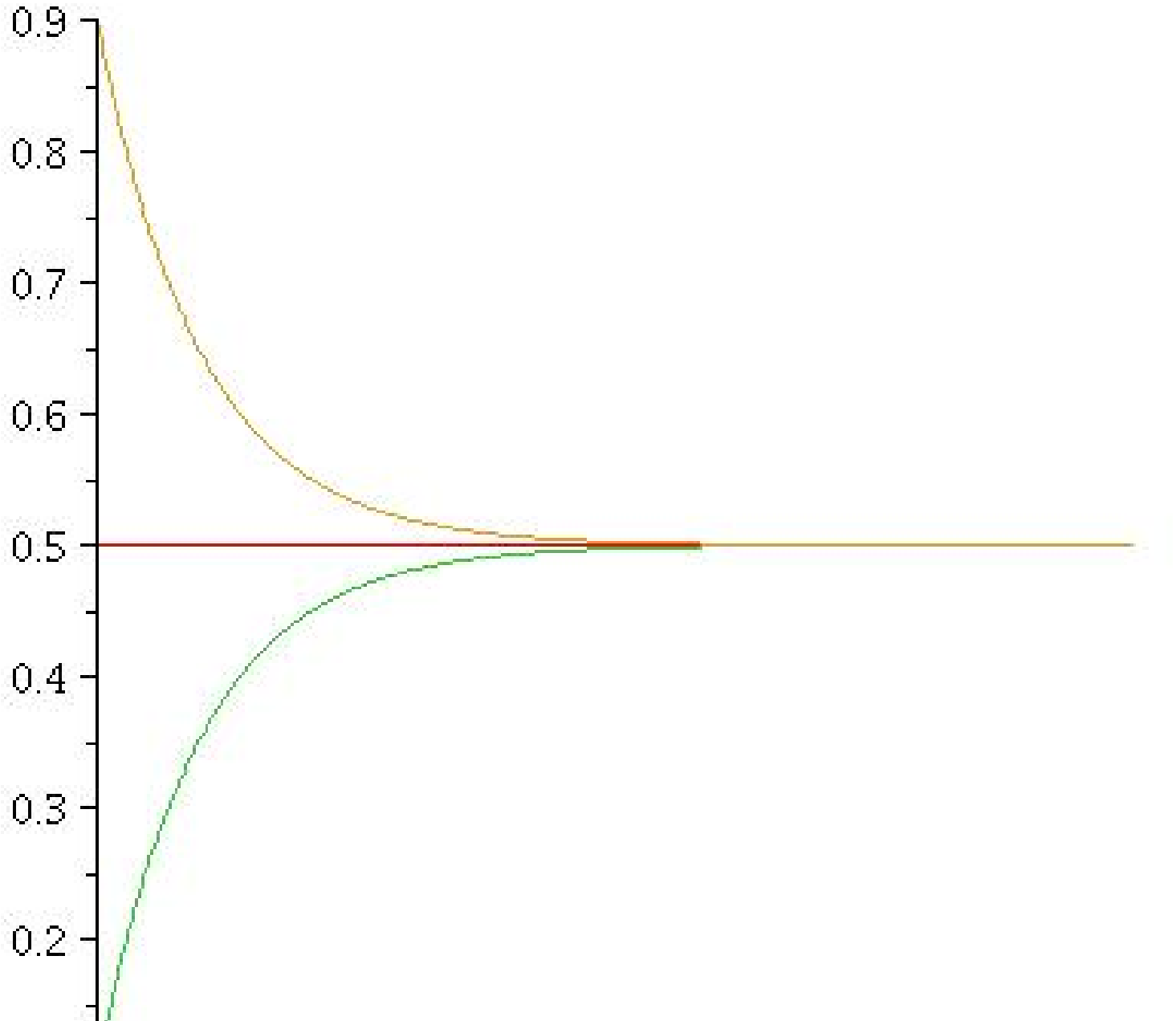
The state space $\mathcal{S} = \{0, 1\}$. At each step, one object is selected at random and flips a coin.

The empirical measure X_1^N is the proportion of ones.
The drift is $F^N(x) = \frac{1}{N}((1-x)/2 - x/2)$, so that $I(N) = 1/N$ and $f(x) = 1/2 - x$.

We are in case 1:
The limit system satisfies the differential equation $dx/dt = 1/2 - x$. The solution is $x(t) = \frac{1}{2} - \left(\frac{1}{2} - x_0\right) e^{-t}$.

Its unique attractor is $x^* = 1/2$.

Simple example (II)



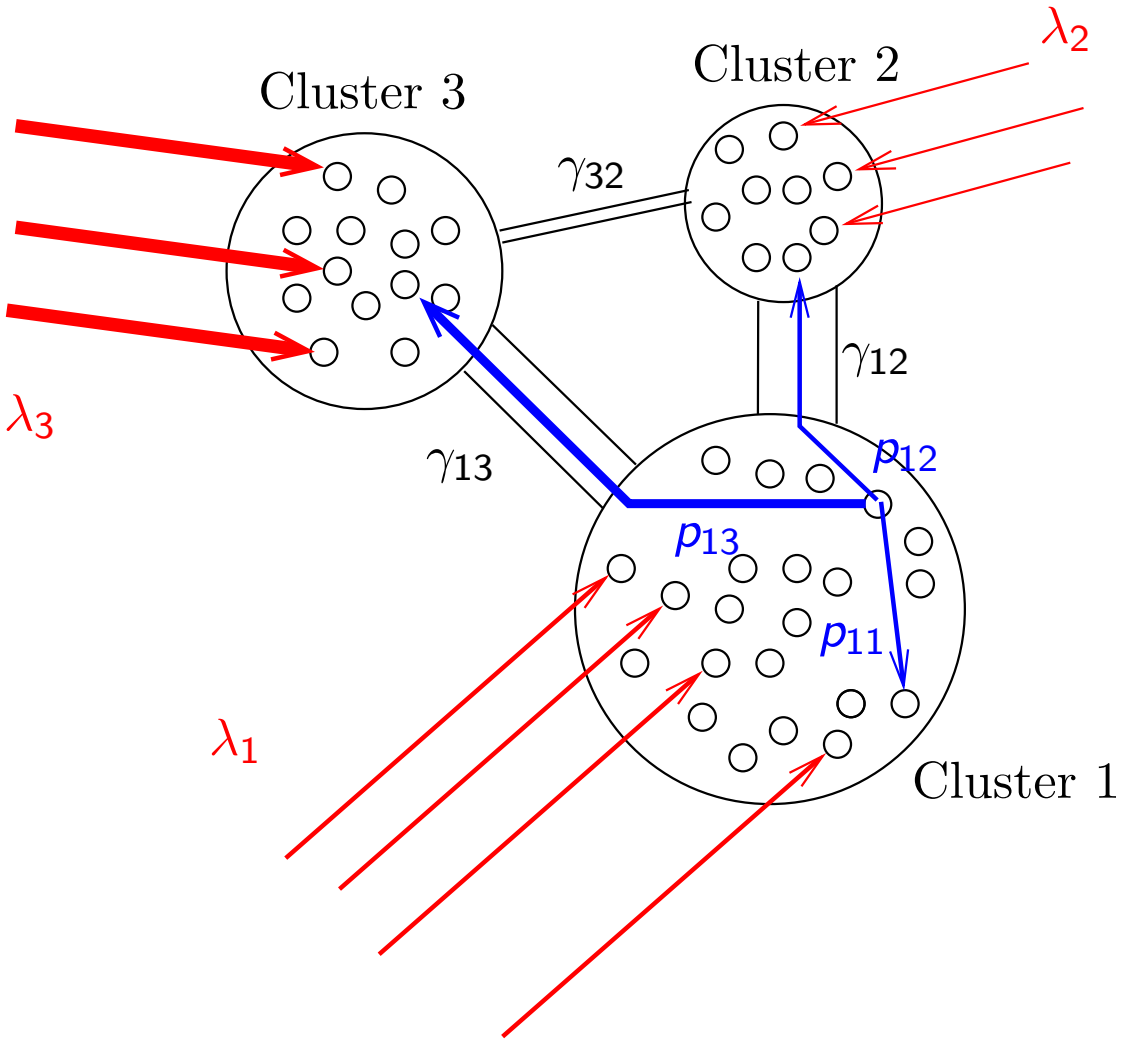
Work stealing model

Let $\{1, \ldots, C\}$ be the set of clusters and K be the buffer capacity of each processor.

If processor p belongs to cluster c_p and has j_p jobs in its buffer, its state is (c_p, j_p).
If p has 0 job and tries to steal from a processor q in cluster c_q its state is $(c_p, 0, c_q)$.

- Stealing rate of c in c' is $\gamma_{cc'}$ (does not depend on the number of stolen jobs).
- The arrival rate per proc. in cluster c is λ_c.
- The speed of a proc. in c is μ_c (equals one by default).
- A proc. in c steals from c' with probability $p_{cc'}$.

Work stealing model (II)

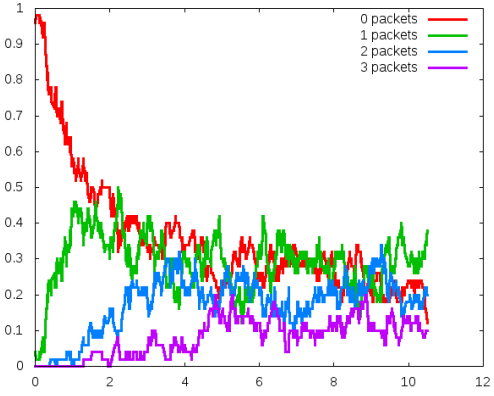


- Stealing rate of c in c':$\gamma_{cc'}$
- The arrival rate per proc. in c: λ_c.
- The speed of a proc. in c: μ_c
- A proc. in c steals from c' with prob. $p_{cc'}$.

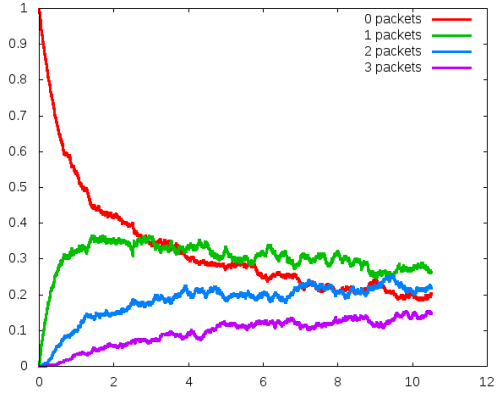
Mean field limit of WS

$$
\begin{aligned}
\dot{x}_{c0c'} &= \mu_c x_{c1} p_{cc'} - (\lambda_c + \gamma_{cc'}) x_{c0c'} + \sum_{c''} \gamma_{cc''} x_{c0c''} \frac{x_{c''0} + x_{c''1}}{x_{c''}} p_{cc'} \\
\dot{x}_{c1} &= \mu_c x_{c2} - (\mu_c + \lambda_c) x_{c1} + \sum_{c'} \lambda_c x_{c0c'} + \sum_{c'} \gamma_{c'c} x_{c'0c} x_{c2} / x_c \\
&\quad + \sum_{c'} \gamma_{cc'} x_{c0c'} (x_{c'2} + x_{c'3}) / x_{c'} \\
\dot{x}_{cj} &= -(\mu_c + \lambda_c \mathbf{1}_{j<K}) x_{c,j} + \mu_c x_{c,j+1} + \lambda_c x_{c,j-1} \\
&\quad + \sum_{c'} \gamma_{c'c} x_{c'0c} (x_{c,2j} + x_{c,2j-1}) / x_c \\
&\quad + \sum_{c'} \gamma_{cc'} x_{c0c'} (x_{c',2j} + x_{c',2j+1}) / x_{c'} \\
&\quad - \sum_{c'} \gamma_{c'c} x_{c'0c} x_{cj} / x_c,
\end{aligned}
$$

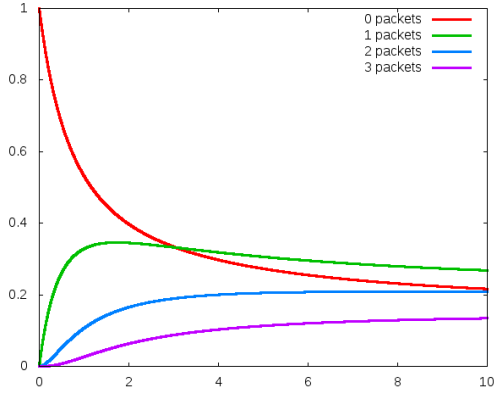
Simulations



50 procs



1000 procs



ODE

One cluster

If all processors belong to the same cluster, the ODE has a unique stationary point, that can be computed using the following iteration.

$x_0 \leftarrow 1 - \lambda/\mu$
$x_1 \leftarrow \lambda(1-\lambda)\frac{\gamma+\mu}{(1-\lambda)\gamma+\mu^2}$
$\forall j \geq 2 : x_j \leftarrow 0.$
repeat
$\forall j \geq 2$
$x_j \leftarrow \frac{1}{\lambda+\mu+\gamma x_0}\left(\lambda x_{j-1} + \mu x_{j+1} + \gamma x_0(x_{2j-1}+2x_{2j}+x_{2j+1})\right)$

Numerical results: Sojourn time

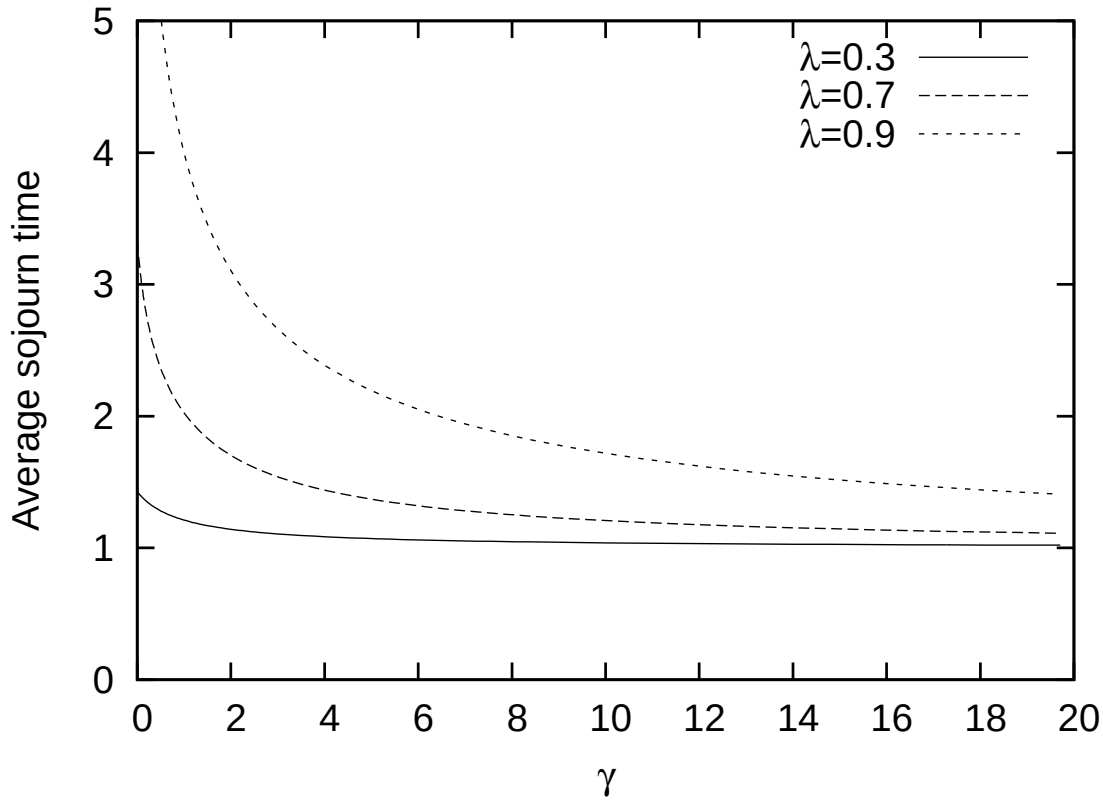


Figure: Average sojourn time as a function of the rate of stealing γ for various values of λ (.3, .7 and .9). As expected, the average sojourn time is decreasing from $S(0) = 1/(1-\lambda)$ to $S(\infty) = 1/\mu = 1$. When γ is small, the average number of jobs in the system decreases drastically.

Numerical results: Number of steals

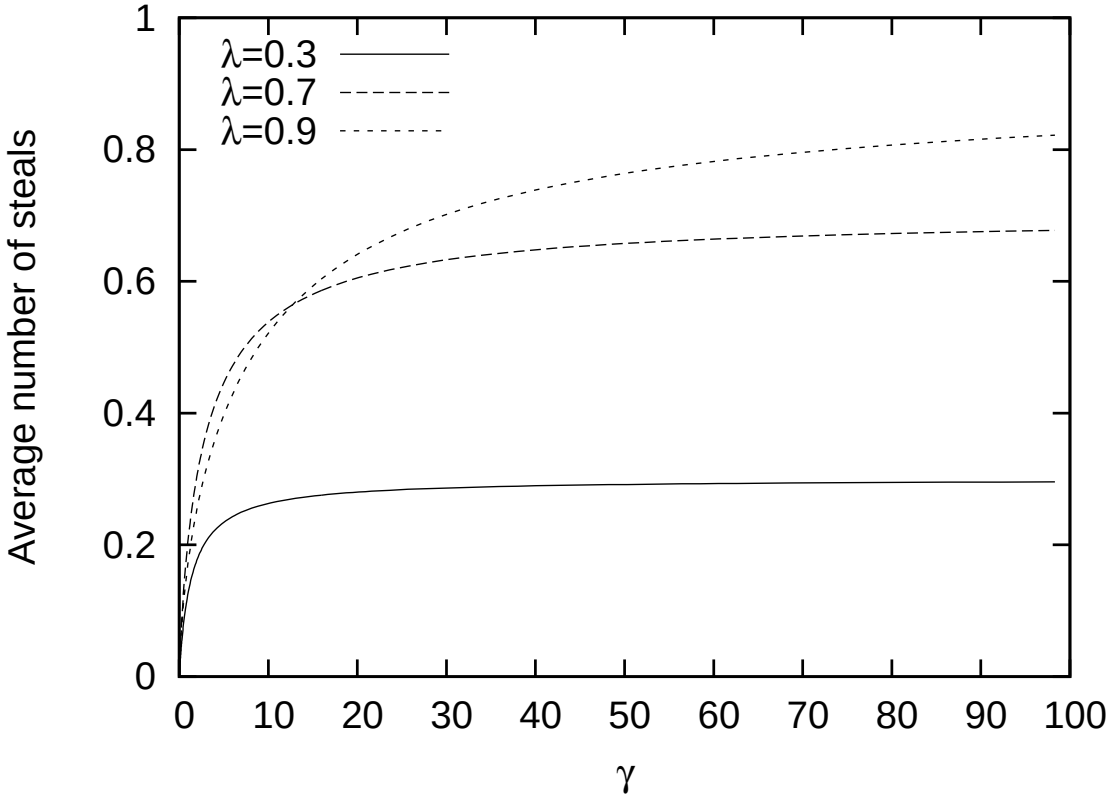


Figure: Average number of successful steals per job $V_\lambda(\gamma)$ viewed as a function of γ for different values of λ (.3, .7 and .9).
The number of steals per job ranges from $V_\lambda(0) = 0$ to $V_\lambda(\infty) = \mathbb{P}(\mathit{non-empty\ buffer}) = \lambda$.

Stealing fraction

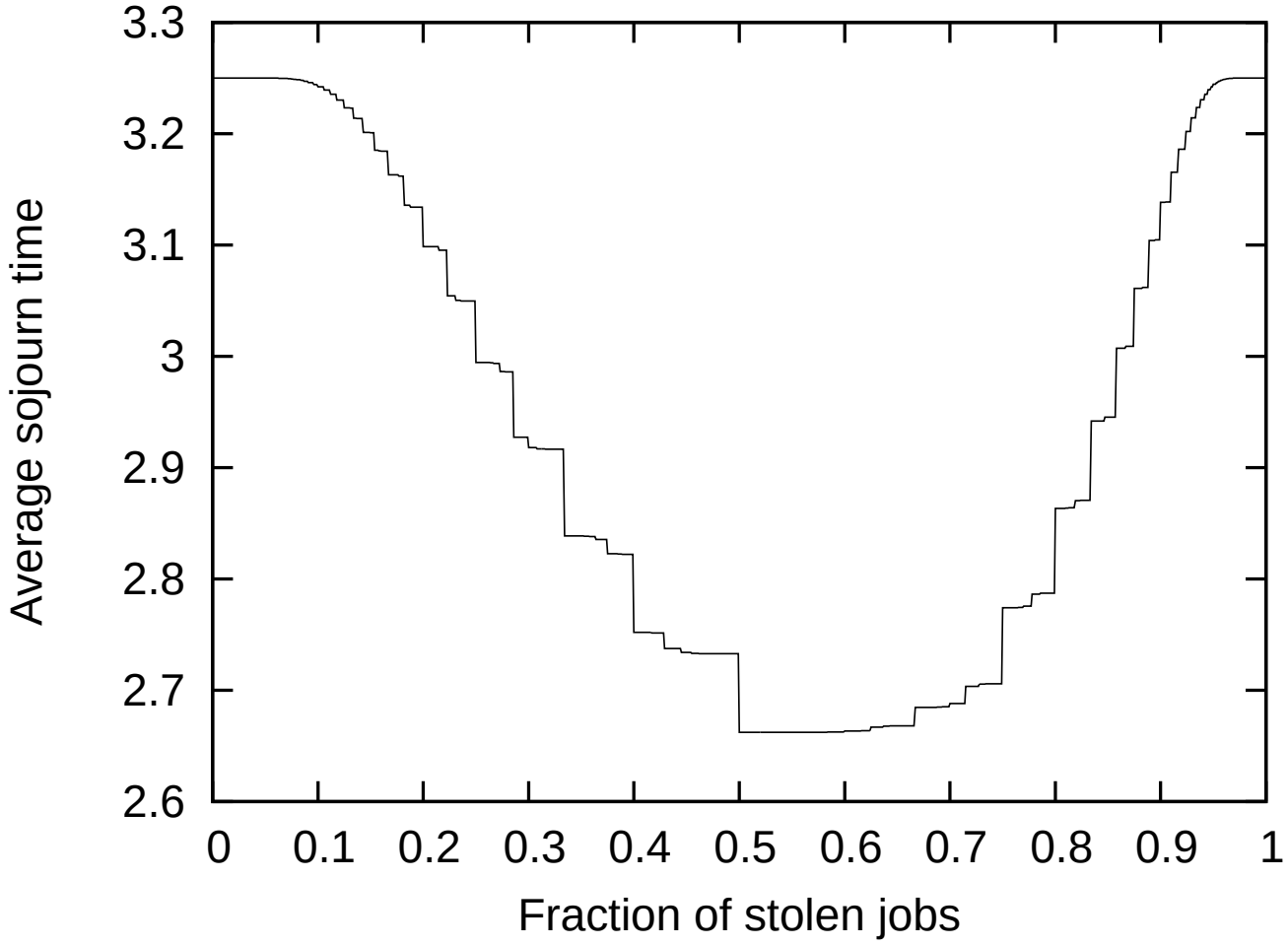


Figure: Average sojourn time as a function of the fraction of jobs stolen at each time for $\lambda = .9$ and $\gamma = 3$.

Batch arrivals

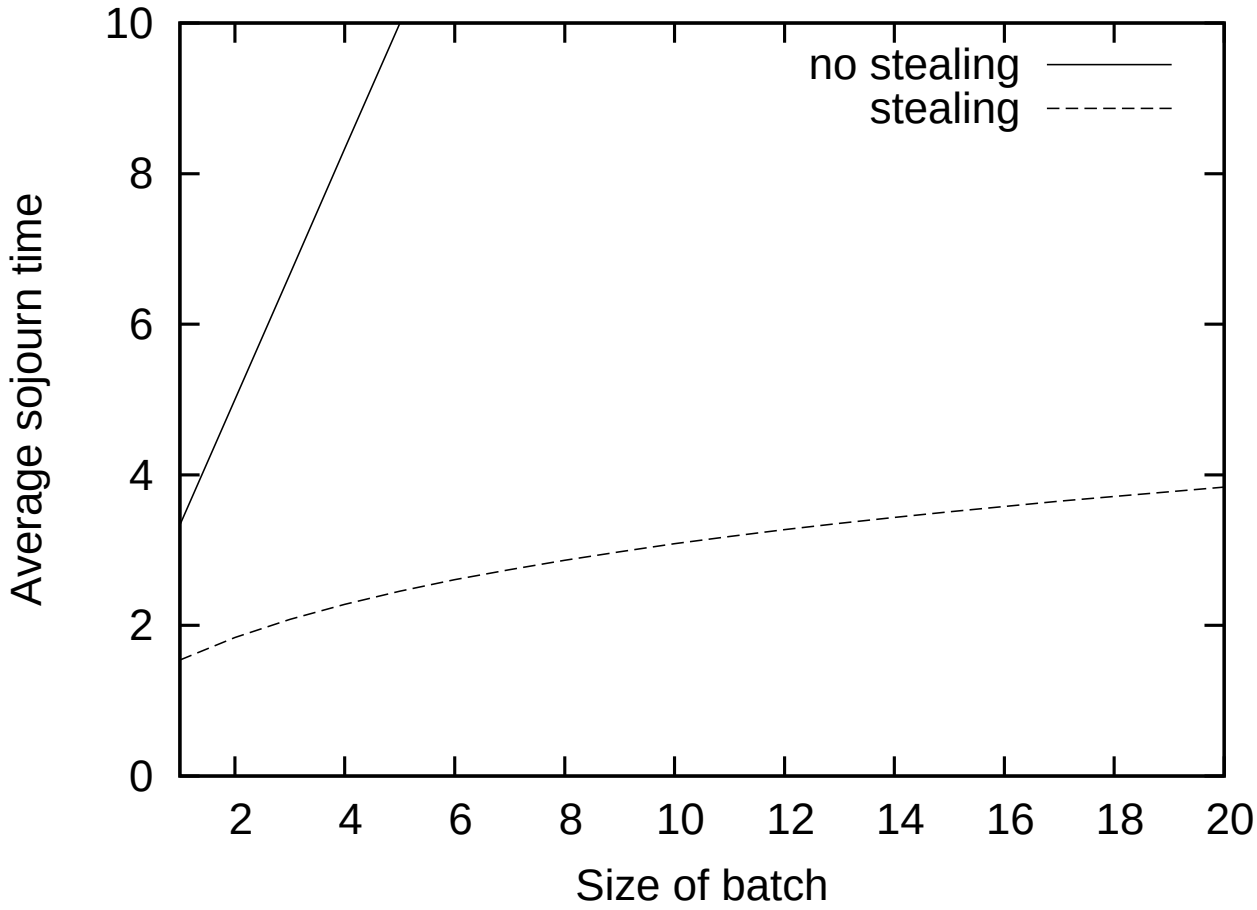


Figure: Average sojourn time as a function of the batch size for $\lambda = .7$. The higher curve represents a system without work stealing while the bottom one shows the results for $\gamma = 3$.

Beyond the empirical measure

Let $J^N(t)$ be the state of one particular processor at time t. For finite N, the behavior of the processor $J^N(t)$ is not independent of the behavior of $X^N(t)$: each transition in $J^N(t)$ changes $X^N(t)$.
The process $J^N(t)$ is not Markovian and is very complicated.
In the limit however, $J^N(t)$ goes to a non-homogeneous Markovian process.

Theorem

$(J^N(t), X^N(t))$ converges weakly to a continuous time jump and drift process $(Y(t), x(t))$ where $x(t)$ satisfies the ODE and $Y(t)$ is a non-homogeneous jump process of kernel $K(x(t))$.

Extreme values of the number of steals

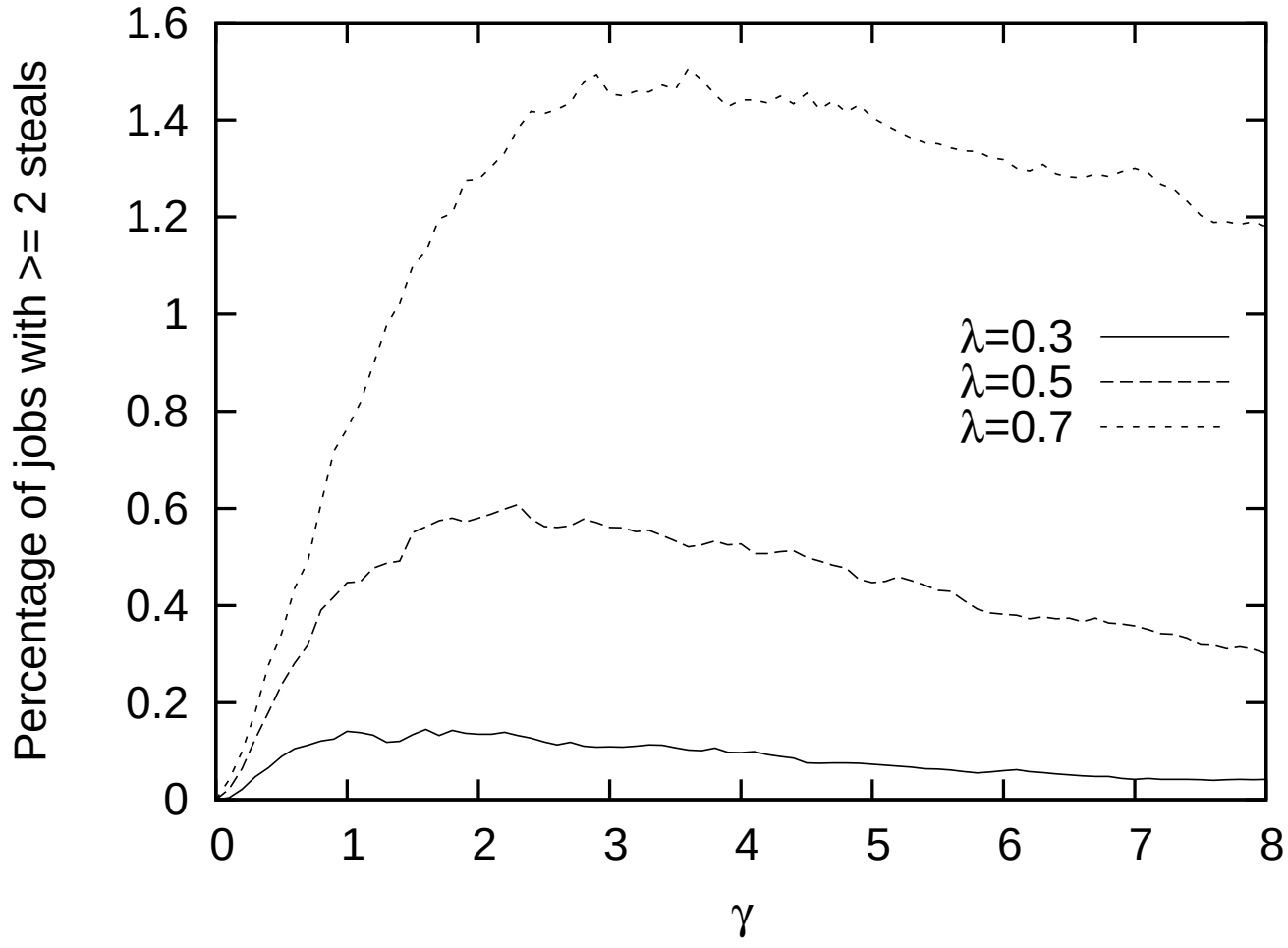


Figure: Fraction of jobs that are stolen twice or more as a function of γ.

Extreme values of Sojourn times

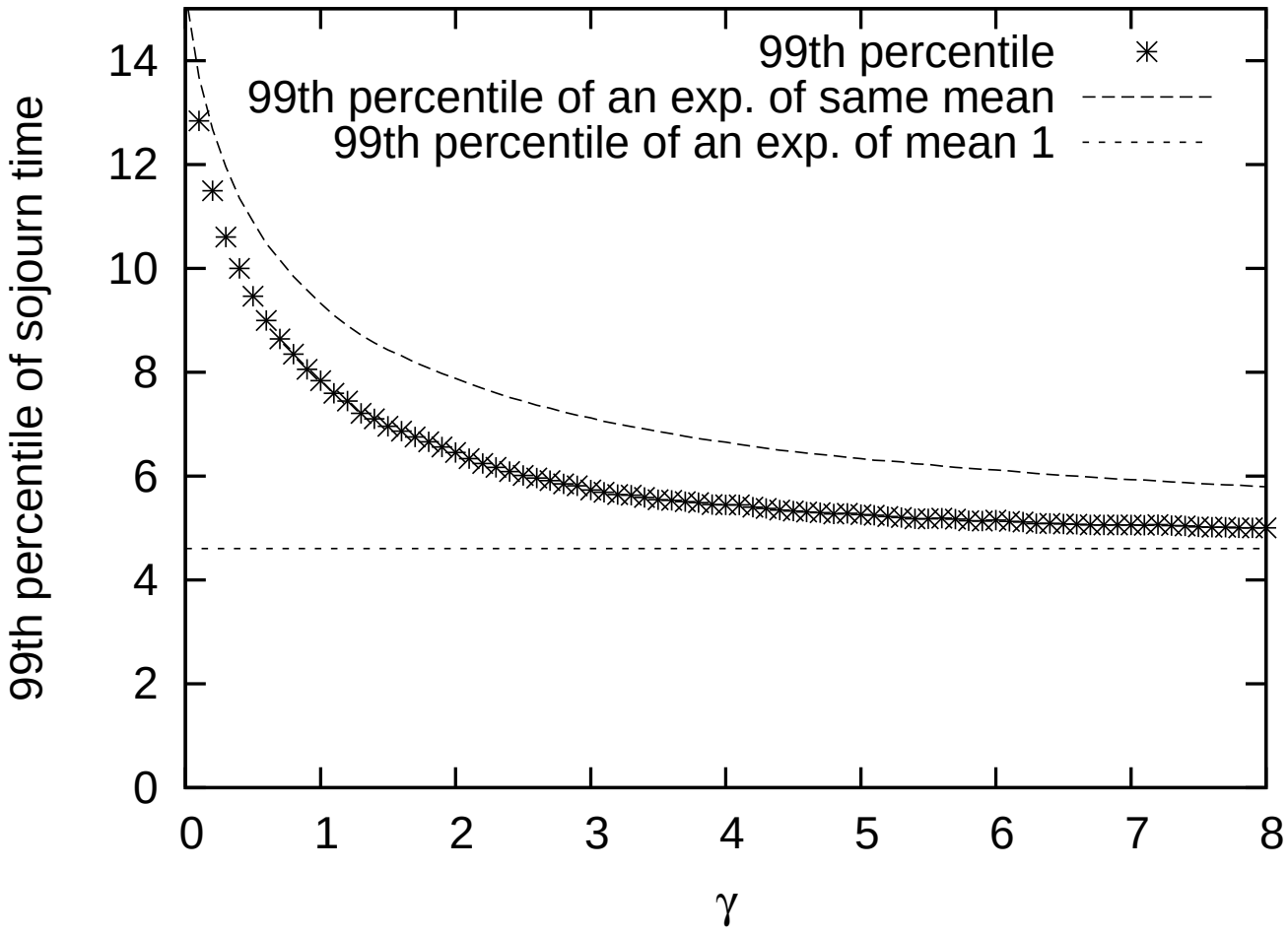


Figure: 99 percentiles of the sojourn time and of an exponential variable of the same mean, as functions of γ, for $\lambda = .7$.

Several Homogeneous clusters

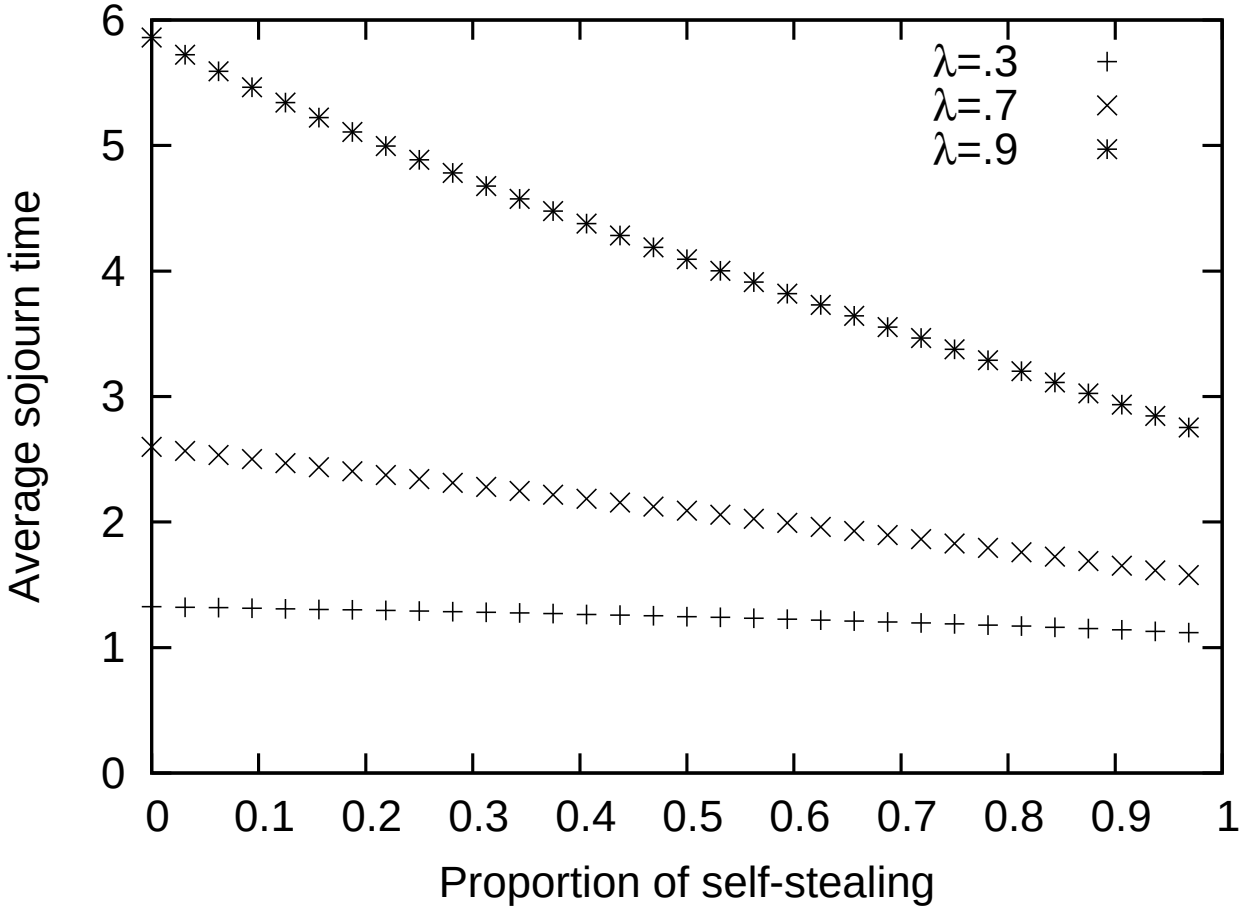
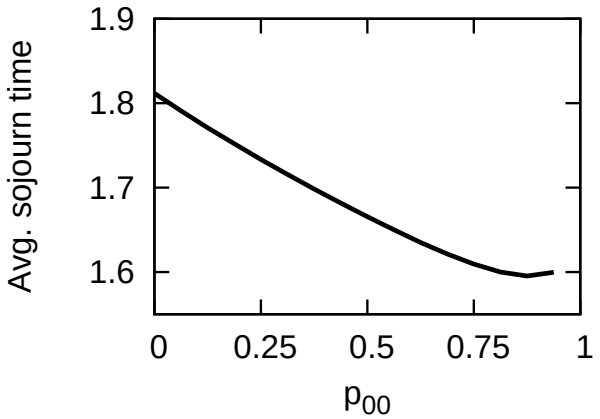
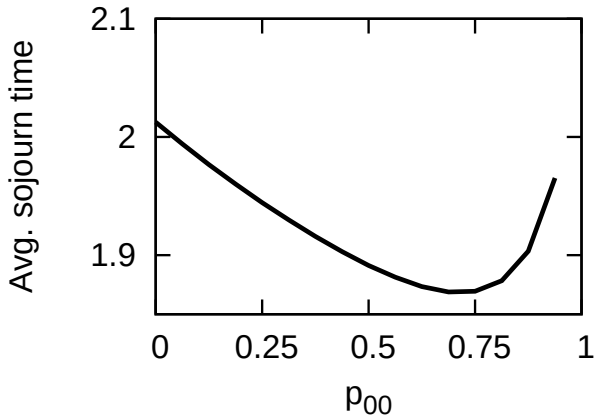


Figure: Average sojourn time in a system with two homogeneous clusters as a function of the probability for a processor to steal inside its cluster when inter-cluster communication is 10 times slower.

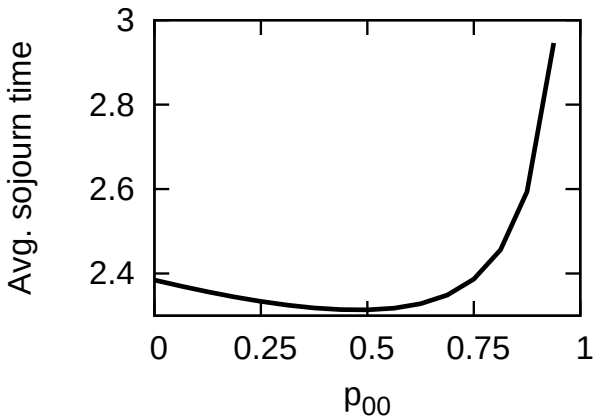
Heterogeneous clusters



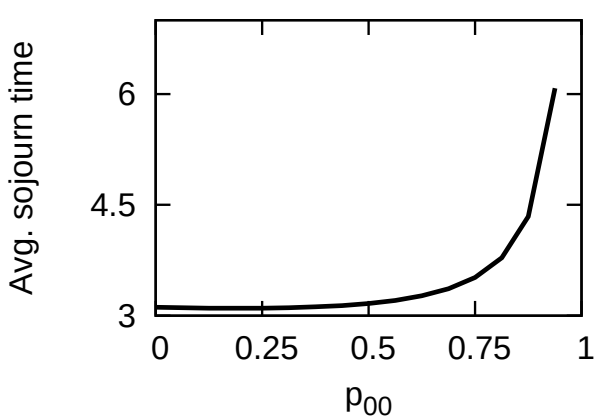
(a) $\lambda_1 = .8$



(b) $\lambda_1 = .9$



(c) $\lambda_1 = 1$



(d) $\lambda_1 = 1.1$

Figure: Average sojourn time as a function of p_{00} for the two heterogeneous model. The first cluster is lightly loaded ($\lambda_0 = .5$). The load of the second cluster is λ_1 (varying from .8 to 1.1).

Hierarchical work stealing: master-worker paradigm

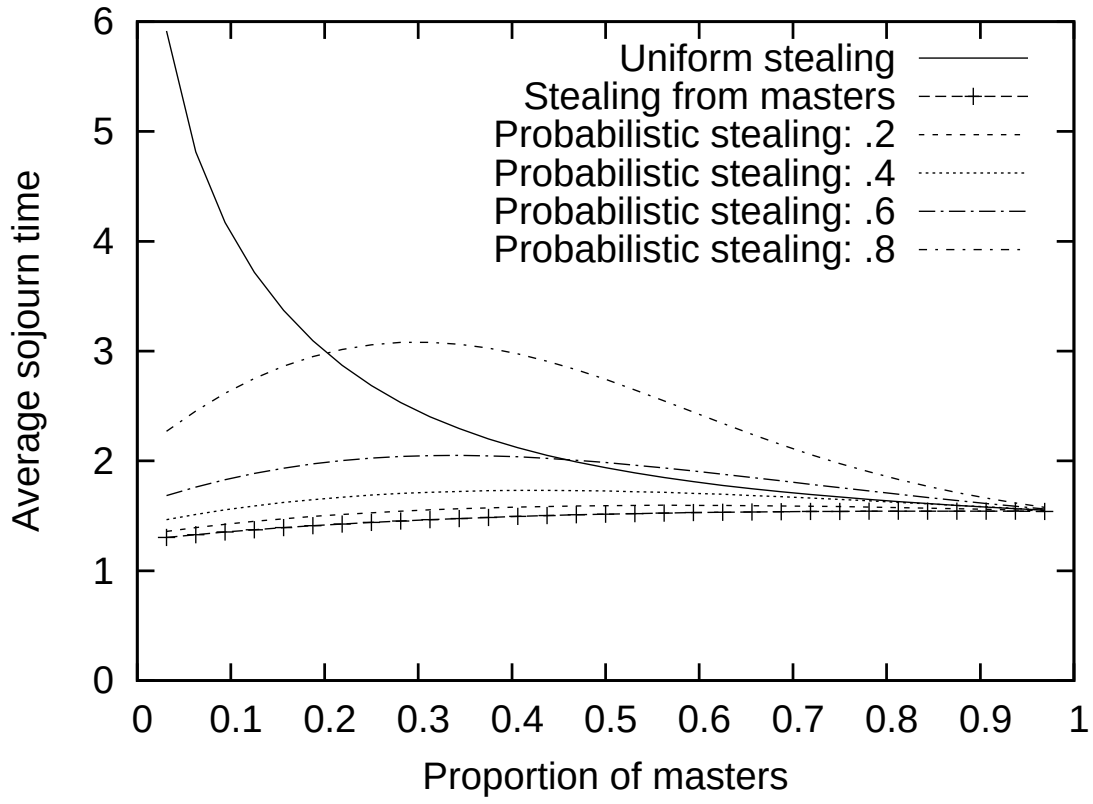


Figure: Comparison of the average sojourn time in the Master-Worker setting with one cluster. Average sojourn time when the batch size is 1.

Hierarchical work stealing (II)

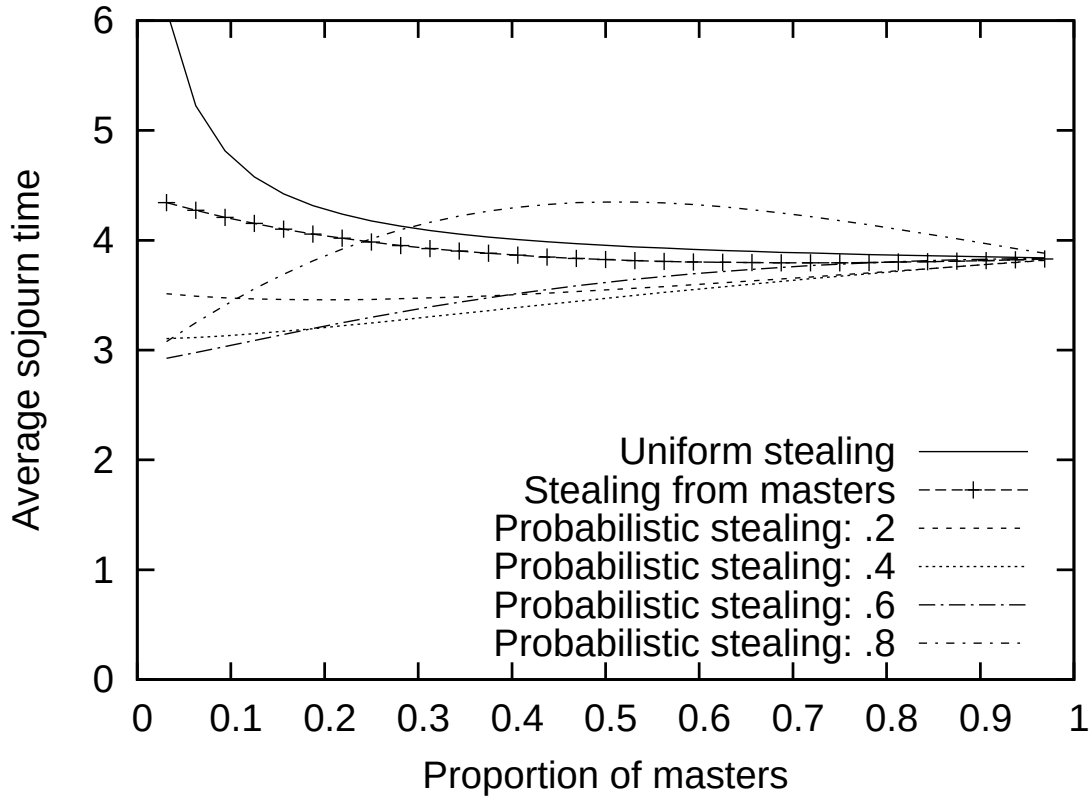


Figure: Comparison of the average sojourn time in the Master-Worker setting with one cluster. Average sojourn time when the batch size is 20.

Having the arrivals concentrated on masters improves the performance if the probabilities of stealing are correctly tuned.