The load transfer model

Bernard Ycart
INRIA-Rocquencourt

November 18, 1996

[summary by Philippe Robert]

1. Introduction

This talk considers a problem of load sharing. A set of N processors are the vertices of some graph G and each of them has a buffer of finite capacity K. A flow of requests (which may be thought of as tasks to be processed) arrives at each vertex. In the case where the vertices do not cooperate in some way, this model is simply a set of N independent finite capacity buffers. In order to optimize the use of total capacity of the network, it is desirable to design a policy which transfers a fraction of the load of highly loaded nodes to under-used processors.

These problems are generic in the sense that they occur very frequently in many mathematical models as soon as finite capacity and possibility of cooperation is assumed. A classical example is the telephone network: If a call from A wants to reach B and all the links between A and B are used, it is possible to use the alternative routing strategy: a node C is chosen at random, a link between A and C an a link between C and B are reserved if possible. If this succeeds, the call between A and B is finally accepted. Notice however that this algorithm has a cost: two links are required instead of one [1]. Another example, again for routing, is Valiant's algorithm [2]. It is often quite easy to guess some reasonable load sharing algorithm, however it is more difficult to estimate its impact on the performances of the system. Moreover, in some cases, load sharing is worst than no sharing at all! [1, 4]

In this talk, our basic assumptions are:

1. The requests arrive at each node according to a Poisson process with parameter λ;
2. Each processor works at unit speed the requests which are assumed to be exponentially distributed with parameter 1;
3. A request arriving at a node which is already full (i.e. with K requests) is rejected.

To estimate the performances of a load sharing algorithm, one can use the following parameters:

- The rejection probability;
- The average use of each processor, or equivalently the average free space in the buffer;
- The mean response time.

2. The load sharing algorithm

We describe the state of the network as a vector $(\eta(x))$ on the set of vertices, $\eta(x)$ is the number of requests at node x. The algorithm considered here is defined as follows: If a request arrives at some node x, then

1. if $\eta(x) = K$, the request is lost;

2. If $\eta(x) < K$ and $\eta(x) \leq \inf\{\eta(y)/y \in N(x)\}$, where $N(x)$ denotes the set of neighbours of x, then the request is assigned to the node x and $\eta(x) \to \eta(x) + 1$;
3. If $K > \eta(x) > \inf\{\eta(y)/y \in N(x)\}$, then the request is sent to a node Y, chosen at random in the set $\{y \in N(x)/\eta(y) < \eta(x)\}$. At node Y, the request follows the same procedure and 3. is applied until case 2. is satisfied. Hence the request will finally be assigned to a processor which is a local minimum in terms of the number of requests in the buffer.

A departure from the node x has the same consequences: a request chosen at random among the nodes with a higher load level is transferred to the node x and so on.

The probabilistic assumptions make the state process $(\eta_t(x))$ at time t, a Markov process. The transition rates can be defined as follows: Let $\alpha(x, \eta)$ be the rate of the requests *through* node x when the state of the process is η, then $\alpha(x, \eta)$ is defined recursively:

1. If $\eta(x) \geq \sup\{\eta(y)/y \in N(x)\}$, then
   $\alpha(x, \eta) = \lambda$ if $\eta(x) < K$,
   $\alpha(x, \eta) = 0$, otherwise.
2. $\eta(x) < \sup\{\eta(y)/y \in N(x)\}$, then

$$\alpha(x, \eta) = \lambda + \sum_{y \in N(x), \eta(y) > \eta(x)} \frac{\alpha(y, \eta)}{\underline{n}(y, \eta)}, \tag{1}$$

where $\underline{n}(y, \eta) = |\{z/\eta(z) < \eta(y)\}|$.

These transitions are quite straightforward according to the description we gave. The term $\frac{1}{\underline{n}(y,\eta)}$ is a consequence of the fact that if y is not a local minimum, then the request is sent at random to some neighbour with a lower load.

The parameter $\alpha(y, \eta)$ is *not* the real rate of requests in the buffer of y. This rate is 0 if y is not a local minimum and $\alpha(y, \eta)$ otherwise. The departure rates are defined in a similar way, using the local minima of η.

3. **Results**

The Markov process we defined above has a local interaction, i.e. any event concerns only a bounded neighbourhood. The local interaction comes from the fact that a local minimum is at most at distance K from any node. This is a special case of interacting particle systems [3]. One of the first non trivial problems for this class of models is the existence of such a process. Given a generator, that is, the transitions, does there exist a semi-group, i.e. a Markov process, with this generator ? Standard techniques show that this is the case. The second problem is the asymptotic behaviour: does there exist an equilibrium measure, a unique one ? and if not what are the extreme points of the convex set of equilibrium measures ? In the case where the graph is the lattice $\mathbb{Z}^d$, the result is the following

Theorem 1. *There is a unique equilibrium measure and the state of the network converges exponentially fast to that equilibrium measure.*

The proof of this theorem is based on a coupling argument: if η, ζ denote two possible states for the network such that $\eta \leq \zeta$, i.e. $\eta(x) \leq \zeta(x)$ for all the nodes x, then it is possible to construct two processes η_t, ζ_t with initial state η, ζ such that $\eta_t \leq \zeta_t$ for all $t \geq 0$. The proof uses this coupling to prove that all the processes stick to the stationary process after some time T with an exponential moment.

3.1. **The case of the complete graph.** when the nodes are all connected, it is then easy to see that the total number of requests $X(t)$ in the system is a birth and death process with birth rate $n\lambda$ if $X(t) \leq (K-1)N$ and $(KN-j)\lambda$ if $(K-1)N < X(t) \leq KN-1$. The death rates are given by $X(t)$ if $X(t) \leq N$, and N otherwise. Asymptotic results are easily derived when $N \to +\infty$.

References

[1] Kelly (F. P.). – Loss networks. *Annals of Applied Probability*, vol. 1, n° 3, 1991, pp. 319–378.

[2] Leighton (F. T.). – *Parallel algorithms and architectures: array-trees-hypercubes.* – Morgan Kaufman, 1992.

[3] Liggett (T. M.). – *Interacting Particle Systems.* – Springer Verlag, New York, 1985, *Grundlehren der mathematischen Wissenschaften.*

[4] Malyshev (V. A.) and Robert (P.). – Phase transition in a loss load sharing model. *Annals of Applied Probability*, vol. 4, n° 4, 1994, pp. 1161–1176.