Taille des messages et quantite d'information dans les algorithmes distribues

C. Lavault

IRIA
CENTRE DE ROCQUENCOURT

Rapports de Recherche

N° 540

TAILLE DES MESSAGES ET QUANTITÉ D'INFORMATION DANS LES ALGORITHMES DISTRIBUÉS

Christian LAVAULT

Juillet 1986

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

TAILLE DES MESSAGES ET QUANTITE D'INFORMATION DANS LES ALGORITHMES DISTRIBUES

Christian LAVAULT

Résumé

Considérant un algorithme distribué A asynchrone résolvant un problème quelconque Π sur un système distribué S (réseau local de communication), on montre que, pour tout message de A, la quantité d'informations à transmettre n'est pas, en général, directement liée à la taille (en nombre de bits) du message. On démontre de ce point de vue deux théorèmes montrant que, sous certaines conditions (CNS pour le théorème 2), l'information est conservée qu'on multiplie la taille d'un message donné par un entier donné k ou bien qu'on la divise par tout entier k ($2 \leq k \leq |M|$ = nombre de messages de A).

La "complexité en nombre de bits" d'un algorithme distribué est mise en cause de ce point de vue pour l'évaluation des performances (complexité en message).

Abstract

Consider a distributed algorithm A asynchrony which solves a certain problem Π on a distributed computing system S (e.g. a local area network).

It is shown herein how the amount of information to be transmitted does not actually depend, in general, on how large the maximal number of bits in any mesage M of A on S. From the latter point of view two theorems are proved and show that, upon certain assumptions, ("iff" conditions in theorem 2) the information is preserved when multiplying the size of a given message by an increasing factor k as well as when dividing the size of a message by any discount factor k (k integer, $2 \leq k \leq |M|$ = # of messages in A).

With respect to the performance of distributed algorithms (message complexity mainly), the "bit complexity" of an algorithm can therefore be considered not relevant indeed.

I - INTRODUCTION

I.1. Généralités

Le plus souvent, un système distribué est considéré comme un réseau de sites (réseau distribué) sur lequel un algorithme distribué résoud un certain problème ; l'hypothèse fondamentale étant que les processus attachés aux sites du réseau ne peuvent communiquer entre eux que par l'intermédiaire de messages qui circulent le long des canaux (lignes de communication) du réseau en question. Par exemple, au sein d'un réseau local (Local Area Network, ou "L.A.N."), les messages en transit selon un protocole de communication bien déterminé ont une taille qui peut varier entre quelques bytes et plusieurs millions de bits suivant le type du LAN.

De manière générale, on fait le plus souvent des hypothèses "favorables" sur la taille maximale des messages transmis au cours de l'exécution d'un algorithme dans un système distribué, et on cherche à contrôler la quantité d'information - le nombre de bits - à chaque étape. Par exemple, la phase dite d'"écho" d'un algorithme distribué du type "shout-echo" (cf. [SA]) nécessite de nombreux messages. Si bien que l'on fait souvent l'hypothèse que la taille des messages d'un algorithme est inférieur à O(T(n)) (où T est une fonction petite de n, taille d'instance du problème, et, éventuellement de d, taille du réseau distribué). Ceci afin de n'avoir que quelques bits par message.

La quantité d'information (au sens large) contenue dans un message est presque toujours vue comme le nombre de bits de ce message. C'est ainsi que, pour des raisons de performance algorithmique, on cherche d'un côté à obtenir le "maximum d'information", tandis que, de l'autre, on tente de diminuer au maximum le nombre total de bits nécessités par l'algorithme distribué.

I.2. Comment le problème est-il posé ?

A notre connaissance, il existe peu de travaux concernant directement ce problème du compromis entre la quantité d'information et la taille des messages.

Bien entendu, le concept même de quantité d'information, d'entropie et, de manière générale, de la probabilité de l'information est connu depuis longtemps par les travaux de Shannon [SH] ou Rényi [RE]. De même, la "complexité à la Kolmogorov" utilisée par ce dernier, puis par Chaiting pour tenter de développer une mesure théorique des programmes séquentiels a été souvent utilisée ([CH], [HA], K. Ko, U. Schöning,

A. Selman, M. Sipser, etc). Mais, même dans la récente approche d'Hartmanis [HA] pour mesurer la "compression" d'une suite de bits (donc d'un message), le point de vue théorique est différent de celui développé ici, même s'il peut l'éclairer par bien des aspects.

Quoi qu'il en soit, la relation entre taille de messages et quantité d'information est d'un intérêt aussi bien théorique que pratique pour l'algorithmique distribué.

D'un point de vue théorique par exemple, [SA] fait l'hypothèse très forte que "the number of values contained in a message is bounded by a (small) constant independant of the network size". De même, la taille des messages dans les réseaux locaux ou autres systèmes de communication est parfois supposé "infinie", afin de simplifier. Dans ces deux cas, le contexte théorique justifie de telles hypothèses. D'autant qu'il est toujours possible d'augmenter quelque peu le nombre de messages pour en réduire la taille.

Cependant, dans certaines publications, ce sont les performances d'un algorithme distribué qui sont évaluées non seulement par sa complexité en temps et/ou sa complexité en messages, mais également du point de vue de l'information contenue par message, plus précisément, de l'ordre de grandeur de la plus grande taille de message (cf. [PA,SA]). Ceci revient, de fait, à prendre pour complexité en messages d'un algorithme A le produit [nombre maximal de messages utilisés par A] × [nombre maximal de bits par message envoyé durant l'exécution de A], autrement dit, une "complexité en nombre de bits" pour A.

Dans [PA,SA] par exemple, les divers algorithmes distribués construisant les arbres couvrants de poids minimal dans un graphe sont jugés à cette aune. L'algorithme de Gallager et alii en 5nlgn + 2e nombre de messages (n=nombre de sommets du graphe, e=nombre d'arêtes du graphe) est ainsi considéré comme meilleur que celui de Chang en O(e) messages, à cause des tailles comparées des messages dans les deux algorithmes, respectivement O(lgn) et O(n). (1)

Les résultats qui suivent peuvent permettre une évaluation de l'importance à accorder à la taille des messages dans un algorithme distribué et, partout, l'évaluation de ses performances réelles et les hypothèses à formuler sur les tailles des messages

---

(1) Dans la suite, "lgn" signifie "$\log_2(n)$".

II - DEFINITIONS

II.1. Système distribué *S*

Considérons un système distribué ***S***, c'est à dire un réseau de communication de taille d et de capacité c (d,c $\in \mathbb{N}$) : $S = \{S_1,...,S_d\}$, ensemble fini de sites dont chacun peut recevoir un maximum de c données dans sa mémoire locale non partageable et, associé à S, l'ensemble $L \subseteq S \times S$ des lignes de communication entre sites. (cf. [LA,LA], et [SA]).

Les seules hypothèses sur ***S*** sont les suivantes :

(i) L'information des d sites de S est uniquement locale : initialement, $\forall\, i \in \{1,...,d\}$, S_i ne connait que ses "voisins", c'est-à-dire les sites qui sont reliés à lui par une ligne de communication.

(ii) Le graphe G = (S,L) déterminé par ***S*** est un graphe non orienté, sans boucle ni arête multiple.

(iii) Les communications dans ***S*** se font uniquement le long des lignes de L par "messages". Un message M peut s'écrire sous la forme $M = <\mu_1\mu_2 \ldots \mu_p>$ où $\mu_i \in \{0,1\}$ $(1 \leq i \leq p)$ sans perte de généralité ; les μ_i sont les bits de M et la suite de bits $\mu_1\mu_2 \ldots \mu_p$ correspond à l'identificateur de M, sa clef, son contenu, etc. Mais, dans la suite, aucune différence n'est faite entre les μ_i.

Dans tout ce qui suit, aucune hypothèse n'est formulée sur la topologie de ***S*** ; de plus, d et c sont absolument quelconque.

II.2. Algorithme distribué A sur *S* et messages de A

Définition 1 (cf. [GA,JO])

Soit un problème quelconque Π, nous dirons qu'un algorithme distribué A, asynchrone, résoud le problème Π sur ***S*** avec une complexité en messages $C_M(A)$ si et seulement si pour toute instance I de Π, A produit une solution sur ***S***, à l'instance I de Π en utilisant un nombre de messages maximal égal à $C_M(A)$, pour toute "longueur d'entrée" n possible[1]

---

(1) La "longueur d'entrée" n pour une instance I d'un problème Π est, par définition, le nombre de symboles de la description de I par un codage de Π. n, la longueur d'entrée, n'est autre qu'une mesure de la taille de l'instance I, codée. Si on note e(I) ce codage d'une instance I d'un problème Π, n = longueur d'entrée = |e(I)|. (cf [GA,JO]).

Définition 2

Un message M d'un algorithme distribué A sur le système distribué $\mathcal{S}$ étant défini comme en II.1., on dira que la taille de M est le nombre total de bits constituant M et on notera t(M) = p cette taille si $M = <\mu_1\mu_2...\mu_p>$, $\mu_i \in \{0,1\}$, $(1 \leq i \leq p)$.

II.3. Notations

1. On notera $\mathcal{M}$ l'ensemble des messages nécessaires à l'exécution d'un algorithme A sur $\mathcal{S}$ pour une instance I d'un problème Π, avec n=taille du codage de I= |e(I)| ; et donc $C_M(A) = O(f(n,d)) = |\mathcal{M}|$, où f(n,d) est une fonction quelconque de n, taille du codage d'instance e(I) de Π et de la taille d de $\mathcal{S}$.

2. Dans la suite, on parlera de "complexité en nombre de bits" pour un algorithme A sur $\mathcal{S}$ dans le sens suivant : $C_b(A)$ = "complexité en nombre de bits" de A sur $\mathcal{S}$ est le nombre maximum de bits nécessaires à l'exécution de A sur $\mathcal{S}$ pour toute instance I d'un problème Π.

$$C_b(A) = \sum_{M \in \mathcal{M}} t(M),$$ où $\mathcal{M}$ est l'ensemble des messages défini en 1.

3. Rappel de la notation classique de Landau utilisée ici :

$f(n) = O(g(n))$ ssi $(\exists \lambda \in \mathbb{R})\ (\exists n_0 \in \mathbb{N})\ (\forall n > n_0$ entier$)\ |f(n)| \leq \lambda g(n)$

$f(n) = \Omega(g(n))$ ssi $(\exists \lambda \in \mathbb{R})\ (\exists n_0 \in \mathbb{N})\ (\forall n > n_0$ entier$)\ |f(n)| \geq \lambda g(n)$

III - DEUX THEOREMES

Montrons tout d'abord que, du point de vue de l'information contenue dans un message, on peut considérer des messages de taille aussi élevée que l'on veut.

Lemme :

Pour tout algorithme A sur un système distribué $\mathcal{S}$, et pour tout message M de A, il existe un algorithme A' sur $\mathcal{S}$ qui simule exactement A et nécessite deux fois plus de bits que A par message.

Démonstration

Soit A résolvant un problème quelconque Π sur $\mathcal{S}$. Considérons un message quelconque de A, $M = <\mu_1\mu_2...\mu_p>$, de taille p, envoyé d'un site S_i à un site S_j au cours de l'exécution de A.

Soit l'algorithme A' sur $\mathcal{S}$ qui, dans les mêmes conditions, envoie le message $M' = <\mu_1\mu_1\mu_2\mu_2\cdots\mu_{p-1}\mu_{p-1}\mu_p 1>$ de taille 2p, correspondant à M.

A la réception du premier bit μ_1 de M', le site S_j attendra tous les μ_i jusqu'à l'arrivée du bit de contrôle 1 qui marque la fin du message M' ; à la réception de 1, S_j sait que M' est terminé.

Ensuite, le site S_j récepteur agit exactement comme s'il avait reçu la suite de bits $<\mu_1\mu_2\cdots\mu_p>$: l'information contenue dans M et M' est exactement le même.

A cette modification près sur la taille des messages, l'algorithme A et l'algorithme A' ont rigoureusement le même comportement, et ceci pour tous les messages. Donc pour tout algorithme A et pour tout message de A, il existe bien un algorithme A' simulant A avec 2 fois plus de bits de communication par message.

q.e.d. □

Remarque

Autrement dit, tout message possèdant une quantité d'information donnée (proportionnelle à sa taille) peut être rendu aussi long que l'on veut sans gain d'information dans un système distribué.

A partir du lemme, il est possible de montrer qu'à quantité d'information égale, on peut diminuer d'un facteur k (entier donné) le nombre des messages d'un algorithme distribué A sur $\mathcal{S}$, pour toute exécution de chaque instance I d'un problème Π, en augmentant la taille des messages de A concernés.

Théorème 1.

Pour tout algorithme distribué A résolvant toute instance I d'un problème Π sur $\mathcal{S}$, et pour toute suite finie de messages $(M_i)_{1\le i\le k}$ (k entier donné) de A, il existe un algorithme A' sur $\mathcal{S}$ qui simule exactement A et nécessite k fois moins de messages d'au plus 2k.p bits, avec $p = \sup_{1\le i\le k} t(M_i)$; et ce, pour toute suite $(M_i)_{1\le i\le k}$ de messages de A.

Démonstration

Soit A résolvant un problème Π sur $\mathcal{S}$, on procède comme dans le lemme précédent.

Considérons une suite finie quelconque de messages de A:$(M_i)_{1\leq i\leq k}$ (k fixé), où $M_i = <\mu_{i_1}\ \mu_{i_2}\ ...\mu_{i_p}>$ est de taille p et $\mu_{i_j} \in \{0,1\}$ pour $1 \leq j \leq p$ et pour tous les éléments de la suite : $1 \leq i \leq k$.

D'après le lemme, il existe un algorithme A' sur $\mathcal{S}$ qui simule exactement A en envoyant, dans les mêmes conditions, un message M'_i correspondant à M_i : $M'_i = <\mu_{i_1}\ \mu_{i_1}\ \mu_{i_2}\ \mu_{i_2}\ ...\ \mu_{i_p}\ 1>$ de taille 2p ; et ce, pour tout i=1,2,3,...,k. Autrement dit, une suite quelconque de messages $(M_i)_{1\leq i\leq k}$ de A peut être "remplacée" dans A' par un seul et unique message M' qui s'écrit alors $M' = M'_1\ M'_2\ M'_3\ ...\ M'_k$, c'est à dire comme la concaténation des séquences de bits constitutifs des messages M'_i $(1\leq i\leq k)$. Ou encore : $M' = <\mu_{11}\mu_{11}\ ...\ \mu_{1p}1\mu_{21}\mu_{21}\ ...\ \mu_{2p}1\ ...\ \mu_{k1}\mu_{k1}\ ...\ \mu_{kp}1>$ si tous les messages M'_i sont de taille 2p, $\mu_{ij} \in \{0,1\}$ $(1\leq i\leq k,\ 1\leq j\leq p)$; les 1 jouant le rôle de bits de contrôle marquant la fin de chaque M_i.

La taille de M' est alors $t(M') = 2k \times t(M_i) = 2k \times p$ si on prend tous les M_i de taille p, ou bien $t(M') \leq 2k \times \sup_i t(M_i) = 2k \times p$, si on considère $p=\sup_i t(M_i)$ comme la taille maximale des messages de la suite de A considérée.

D'après le lemme, il est clair que A' simule A sur $\mathcal{S}$ et que la quantité d'information contenue dans la suite $(M_i)_{1\leq i\leq k}$ et dans l'unique message correspondant M' de A' est rigoureusement la même. D'où le théorème 1.

q.e.d. □

Inversement, le théorème 2 qui suit prouve que, sous certaines conditions, on peut de même diminuer la taille de n'importe quel message de A sans perte d'information.

Théorème 2.

Pour tout algorithme distribué A résolvant toute instance I d'un problème Π sur $\mathcal{S}$, et pour tout message M de A, il existe un algorithme distribué A' sur $\mathcal{S}$ qui simule exactement A et nécessite k fois moins de bits ($\forall$ k entier, $2\leq k\leq |M|$) si et seulement si la taille de M est inférieure ou égale à l'ordre de grandeur du nombre de messages nécessaires à A pour résoudre I de Π sur $\mathcal{S}$: $t(M) \leq C_M(A)$; pour toute taille n de codage d'instance e(I) suffisamment grande.

Démonstration

a. Condition suffisante :

L'idée est tout simplement de diviser par un entier k, $2 \leq k \leq |M|$ le (ou les) message(s) de M de taille(s) maximale(s). (Tous les messages sont de taille maximale si on suppose que les tailles sont à peu près les mêmes pour tous les messages de A). Dans tous les cas, il s'ensuit alors qu'il y a k+1 messages en plus à considérer dans M', ensemble de messages de l'algorithme A' qui simule A sur S. Mais ceci est sans importance du point de vue de la complexité en messages SI l'augmentation n'est pas trop grande et conserve le même ordre de grandeur en nombre de messages, i.e. la même complexité en messages. Formellement, on prouve ce qui suit :

Pour tout algorithme A sur S exécutant toute instance I d'un problème Π où $|e(I)| = n$ et d est le nombre de sites de S, considérons un message M de A tel que $t(M) = \sup_{1 \leq i \leq |M|} t(M_i)$.

Par hypothèse (condition suffisante), on a $t(M) \leq C_M(A) = O(f(n,d))$ (où f est une fonction quelconque de $n = |e(I)|$ et de $d = |S|$).

L'exécution totale de A nécessitant $|M|$ messages de longueur totale $C_b(A) = \sum_{M \in M} t(M)$, on a la double inégalité qui suit sur t(M) :

$(\exists \lambda \in \mathbb{R})$ tel que $t(M) \leq \lambda f(n,d)$ et
(pour tout n et d assez grand)

$$\left\lfloor \frac{C_b(A)}{\lambda f(n,d)} \right\rfloor \leq t(M) \leq C_b(A) - (\lambda f(n,d) - 1)$$

Pour tout k entier tel que $2 \leq k \leq \lambda f(n,d)$, on aura alors

$$(1) \quad \left\lfloor \frac{C_b(A)}{\lambda f^2(n,d)} \right\rfloor \leq \left\lceil \frac{t(M)}{k} \right\rceil \leq \left\lceil \frac{C_b(A) - \lambda f(n,d) + 1}{2} \right\rceil$$

et ce, pour tout message M de A de taille maximale.

Un algorithme A' sur S qui exécute toute instance I de Π en simulant exactement A se définit comme suit. Si on note M' l'ensemble maximum des messages nécessaires à A'

sur S pour simuler A et $t(M') = \sup_{M' \in M'} t(M') = \left\lceil \frac{t(M)}{k} \right\rceil$ la taille maximale de message de A' sur S. ($C_b(A') = \sum_{i=1}^{|M'|} t(M'_i)$). A' simule A pour tout message de M car ($\forall$ M$\in M$) ($\exists$ M'$\in M'$) tel que, si $M = \langle \mu_1 \mu_2 \dots \mu_p \rangle$, le message M' de A' correspondant est partitionné en k "sous-messages", $M'_1 = \langle \mu_1 \mu_2 \dots \mu_q \rangle$, $M'_2 = \langle \mu_{q+1} \dots \mu_r \rangle$, ..., et $M'_k = \langle \mu_{s+1} \dots \mu_p \rangle$ avec $t(M'_i) = \left\lceil \frac{t(M)}{k} \right\rceil$ ($1 \le i \le k$).

Par conséquent, $t(M'_i)$ vérifie les inégalités de (1) (pour i=1,...k) et $|M'|$ = nombre maximum de messages nécessaires à l'exécution de A' s'écrit en fonction de $|M|$, pour tout message M$\in M$:

$|M'| = |M| + k + 1$; donc, puisque $k \le \lambda f(n,d)$ et $|M| = \lambda f(n,d)$

$|M'| \le \lambda f(n,d) + \lambda f(n,d) + 1$

et $|M'| = O(f(n,d))$

Par conséquent, $\boxed{C_M(A) = C_M(A') = O(f(n,d))}$

Il s'agit bien sûr ici d'<u>un</u> majorant pour k, car si on souhaite seulement obtenir, pour plus court message, un message de taille t(M') = 1 bit, si $t(M) = p \le \lambda f(n,d)$ il suffit de prendre $k = p \le \lambda f(n,d)$.

<u>b. Réciproquement</u>

Nous allons considérer un algorithme quelconque A sur S nécessitant comme en a. $|M|$ message pour exécuter toute instance I de taille de codage n d'un problème Π. Tous les messages de A sont à peu près du même ordre de taille, strictement plus grand que l'ordre de grandeur de f(n,d) (où $C_M(A) = |M| = O(f(n,d))$).

($\forall$ M $\in M$) $t(M) = O(\sup_{1 \le i \le |M|} t(M_i))$ et $t(M) > |M|$ donc

($\forall$ M $\in M$) ($\exists$ $\lambda \in \mathbb{R}$) $t(M) > \lambda f(n,d)$

Posons donc ($\forall$ M$\in M$) $t(M) = \lceil \phi(f(n,d)) \times f(n,d) \rceil$ où ϕ est une fonction réelle croissante avec f et telle que

($\forall$ f) $\phi(f) \ge \lambda$ pour tout paramètre n et d assez grands.

Pour tout entier k, $2 \le k \le |M|$ on aura alors, (avec ϵ réel, $0 < \epsilon < 1$), $\epsilon \le 2 \le k < \lceil \phi(f) \times f \rceil$ donc

(2) $$\left\lfloor \frac{f.\phi(f)}{f.\phi(f)} \right\rfloor \leq \left\lceil \frac{t(M)}{k} \right\rceil < \left\lceil \frac{f.\phi(f)}{f^{1+\epsilon}} \right\rceil \qquad (0 < \epsilon < 1)$$

Donc, si on considère, comme en a. un algorithme A' simulant A sur $\mathcal{S}$ dans l'exécution de toute instance I de Π, $(\forall\ M' \in \mathcal{M}')$ $t(M') = \frac{t(M)}{k}$ et donc vérifiera l'inégalité (2). C'est à dire : $t(M') \geq 1$ (en fait, on arrive facilement à $t(M') = 1$ bit) et, du point de vue du nombre maximum de messages de A', $|M'| = |M| + k + 1$ pour tout messages $M \in \mathcal{M}$.

Donc, comme $k > \left\lfloor \frac{t(M)}{f.\phi(f)} \times f^{1+\epsilon} \right\rfloor$ $(0 < \epsilon < 1)$ d'après (2), pour tout algorithme A et pour tout message M de A, il existe un algorithme A' dont le nombre maximum de messages $|M'|$ sur $\mathcal{S}$ pour exécuter toute instance I d'un problème Π est de la forme

$$|M'| > |M| + \left[f(n,d)\right]^{1+\epsilon} + 1 \qquad (0 < \epsilon < 1)$$

sachant que $|M| = O(f(n,d))$

$$|M'| > O(f(n,d)) + f(n,d) \,.\, \left[f(n,d)\right]^{\epsilon} + 1 \qquad (0 < \epsilon < 1)$$

Donc $|M'| = \Omega(f^{\epsilon}(n,d) \times f(n,d))$ $(0 < \epsilon < 1)$

et

$$\boxed{C_M(A) = O(f(n,d)) \text{ mais } C_M(A') = \Omega(f^{\epsilon}(n,d).f(n,d)) \quad (0 < \epsilon < 1)}$$

(avec $f^{\epsilon}.f > f$) q.e.d. □

Exemple :

$f(n,d) = \lg n$ et $\phi : f \rightarrow \phi(f) = f^2 + \lambda$ par exemple

on a, en fixant $\epsilon = 1/2$:

$$\left\lfloor \frac{1}{(\lg n)^{3/2}} \right\rfloor \leq 2 \leq k < (\lg n)(\lg n)^2 + \lambda \quad \text{pour } n \geq 2$$

alors, $|M'| = |M| + k + 1 > O(\lg n) + (\lg n)^{3/2} + 1$ si $|M| = O(\lg n)$.

Dans ce cas, on voit bien que $C_M(A) = O(\lg n) < C_M(A')$

et $C_M(A') > (\lg n)^{3/2} = \Omega((\lg n)^{3/2})$

(avec $(\lg n)^{3/2} > \lg n$ pour $n \geq 3$)

Quoiqu'il en soit, $C_M(A')$ est d'un ordre de grandeur strictement supérieur à celui de $C_M(A)$. Par conséquent, lorsque la taille des messages de A est strictement plus grande que l'ordre de grandeur du nombre de messages total nécessaires à l'exécution de A sur S pour toute instance I d'un problème Π, il s'ensuit qu'il existe un algorithme A' sur S qui simule A avec des messages dont la taille maximale est inférieure à celles des messages de M, mais faisant croître la complexité en messages à un ordre supérieur $C_M(A') > C_M(A)$.

IV - PROPOSITION CONCERNANT LA DIMINUTION DE TOUTES LES TAILLES DE MESSAGES

Le précédent résultat est valable pour tout message M de A, algorithme distribué qui résoud toute instance I d'un problème Π sur M, mais ne concerne pas tous les messages de A "à la fois".

De fait, si toutes les tailles des messages de A sont divisées par des entiers k ($2 \leq k \leq |M|$), l'augmentation du nombre de messages nécessaire à l'algorithme A' pour simuler l'algorithme A peut être assez importante pour que $C_M(A') > C_M(A)$, sous l'hypothèse du théorème où $(\forall M \in M)$ $t(M) \leq C_M(A) = O(f(n,d))$.

La proposition suivante précise ce point.

Proposition :

Soit un algorithme distribué A résolvant toute instance I d'un problème Π sur S tel que tout message de A dans son exécution de I soit de taille inférieure ou égale à $C_M(A)$ pour toute taille de codage d'instance suffisamment grande. Alors, tout algorithme A' qui simule exactement A sur S avec des messages de tailles k fois inférieures à celles des messages de A est tel que $C_M(A) = C_M(A')$ SI les entiers k divisant les tailles de tous les messages de A vérifient la condition $2 \leq k \leq \rho$ avec $\rho < |M|$ et ρ indépendant du nombre de message $|M| = O(f(n,d))$.

Démonstration

Identique à la précédente (condition suffisante).

Comme précédemment, $(\forall M \in M)$ $(\exists \lambda \in \mathbb{R})$ tel que $t(M) \leq \lambda f(n,d)$ pour tout n et d suffisamment grands (n=taille de e(I), d=taille de S = nombre de sites de S).

A' simule A sur $\mathcal{S}$ pour résoudre toute instance I d'un problème Π avec un nombre de message total $|M'|$ de tailles telles que

$$(\forall\, M' \in M') \quad t(M') = \left\lceil \frac{t(M)}{k} \right\rceil \quad , \quad k \text{ entier quelconque } 2 \le k \le \rho$$

avec $\rho < |M|$ et ρ indépendant de $|M|$.

Posons $K = \max\{k \in \mathbb{N} \;/\; 2 \le k \le \rho$ et k divise t(M) pour tout $M \in M\}$. A' est tel que $|M'| \le |M| + (K+1)\,|M|$ puisque tous les messages de A sont au plus divisés par K.

Donc $|M'| \le 2\,|M| + \rho|M|$, et comme $K \le \rho < |M|$, ρ indépendant de $|M|$, on a $|M'| < (\rho+2)\,|M|$ et donc $|M'| = O(f(n,d)) = C_M(A)$ ou encore

$$\boxed{C_M(A) = C_M(A')}$$

q.e.d. □

Remarque

Ces résultats sont valables dans un cadre asynchrone. Mais, si on considère sur $\mathcal{S}$ un algorithme A synchrone, le fait d'augmenter la complexité en messages de A peut obérer la complexité en temps de A.

En effet, dans ce type d'algorithme, on fait souvent l'hypothèse que les activités de communication dépassent celle du "calcul" local des processus sur les sites. La complexité en temps de ces algorithmes distribués ne peut être augmentée que par les délais de communication de site à site. Or,

1. Dans le cas du théorème, diviser la taille d'un message M par k oblige à envoyer k+1 messages en plus (de taille $\lceil t(M)/k \rceil$), donc augmente inévitablement les délais d'acheminement de l'information initialement contenu dans $M = \langle \mu_1 \mu_2 \ldots \mu_p \rangle$ de site à site. dans ce cas, il faut que les t(M) = p bits de M soient envoyés dans les k+1 messages "suffisamment vite" si l'on veut sauvegarder l'ordre de grandeur des délais de communication de l'algorithme.

2. Par ailleurs, toute augmentation substantielle du nombre des messages en transit dans le système distribué $\mathcal{S}$ peut entraîner une saturation de ce système $\mathcal{S}$ avec, éventuellement, des interblocages au sein même de $\mathcal{S}$.

Par conséquent, même si du strict point de vue de la complexité en messages d'un algorithme distribué A sur $\mathcal{S}$, la diminution de la taille des messages peut n'avoir aucune conséquence, il peut se faire qu'il en aille tout autrement du point de vue de la complexité en temps de A (synchrone) ou du point de vue du fonctionnement même du système distribué $\mathcal{S}$.

V - CONCLUSION

Dans un système distribué $\mathcal{S}$ quelconque, on a vu que pour tout message M de tout algorithme A sur $\mathcal{S}$ qui résoud toute instance I d'un problème Π, l'ordre de grandeur de la taille de M (en nombre de bits) est sans influence sur la complexité en message de A si et seulement si la taille de M est inférieure ou égale à la complexité en message de A pour résoudre les instances de Π sur $\mathcal{S}$.

Par contre, il n'est pas possible de diminuer arbitrairement les tailles de tous les mesages nécessaires à l'exécution d'un algorithme A quelconque résolvant toute I de Π sur $\mathcal{S}$. La complexité en messages de A, $C_M(A)$, dans ces conditions, reste du même ordre de grandeur si on diminue toutes les tailles des messages de A d'un facteur indépendant de $C_M(A)$ et inférieur à $C_M(A)$.

Par conséquent, la "complexité en nombre de bits" d'un algorithme quelconque A vérifiant les hypothèses ci-dessus n'est pas une mesure réellement significative de la performance des algorithmes distribués : le nombre maximum de bits par message de A n'a pas, en général, à intervenir dans l'évaluation de sa complexité en message pour résoudre toute instance I d'un problème Π sur un système distribué.

On peut interpréter concrètement ces résultats grâce aux exemples donnés en Introduction.

Dans la mesure où l'on ne connaît pas à l'avance la complexité en messages d'un algorithme distribué, on peut faire de fortes hypothèses sur la taille des messages (cf. [SA] dans un cadre purement théorique) ... quitte à les affaiblir une fois cette complexité connue.

Par contre, en ce qui concerne les performances comparées des deux algorithmes cités en Introduction : Gallager et alii d'une part, Chang d'autre part, force est de constater que ces algorithmes sont de complexité en messages tout à fait comparable (sauf si le graphe est un arbre et donc e=n-1), savoir : O(nlgn) + O(e) pour Gallager et alii et

O(e) pour Chang. Les tailles des messages des deux algorithmes étant inférieurs à ces quantités (O(lgn) et O(n) respectivement), le nombre maximal de bits par message n'a pas à intervenir dans la comparaison de la complexité en messages des deux algorithmes - sauf, encore une fois si le graphe dont on cherche l'arbre couvrant minimal est déjà un arbre !.

Par contre, il peut se faire qu'un nombre de bits en O(n) par message induise des difficultés d'implémentation de l'algorithme et, dans le cas où celui-ci est synchrone, augmente la complexité en temps par rapport à un algorithme en O(lgn) nombre de bits par message.

BIBLIOGRAPHIE

[CH] G.J. CHAITING, A Theory of Program Size Formally Identical to Information Theory, JACM, Vol. 22, n° 3, July 1975, pp. 329-340.

[GA,JO] M.R. GAREY, D.R. JOHNSON, Computers and intractability - Freeman (ed), (1979).

[HA] J. HARTMANIS, Generalized Kolmogorov complexity and the structure of feasible computations, Proc. 24th IEEE Symp. Foundations of Computer Science, 1983, pp. 439-445.

[LA,LA] I. LAVALLEE, C. LAVAULT, Algorithmique parallèle et distribuée - Rapport de Recherche INRIA n° 471, (Dec. 1985).

[PA,SA] D.S. PARKER, B. SAMADI, Distributed Minimal Spanning Tree Algorithms - Performance of Data Communication Systems and their Applications, G. Pujolle (ed), North-Holland, (1981).

[SA] N. SANTORO, On the message complexity of Distributed Problems - School of Computer Science (Carleton Un., Ottawa), T.R. N° 13 (Dec. 1982), and Journal of Computing Information Science (to be published).

[SH] C.E. SHANNON, A mathematical theory of communication, The Bell System Technical Journal, Vol. 27, n° 3, July 1948, pp. 379-423 and pp. 623-656.

[RE] A. RENYI, Calcul des probabilités, Dunod 1966.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique