TIME-PROCESSOR TRADE-OFFS FOR UNIVERSAL PARALLEL COMPUTERS

Friedhelm Meyer auf der Heide
University of Bielefeld
Faculty of Mathematics

4800 Bielefeld 1
West-Germany

Abstract: Various kinds of simulations among parallel computers are considered. We prove that it is not possible to simulate all parallel computers by one without an additional expense of time or of processors. Then we present a parallel computer which can simulate all parallel computers with n processors and which needs $c \cdot n^2 \log(n)$ processors and extends the time only by a factor $d \cdot \log\log(n)$.

I. Introduction

1. Parallel computers

In 1980 Paul and Galil introduced a model of parallel computation and dealt with the question: How good can one parallel computer simulate all others? (I). They found a simulation which doesn't need more processors than the computers being simulated but demands an additional expense of time of a factor being logarithmic in the number of processors.

In this paper we want to get deeper insight into the behaviour of time and number of processors necessary for these simulations.

First we give a short description of a model of parallel computation essentially due to (I).

A parallel computer (PC) M is given by

- a set $P = \{P_1, \ldots, P_n\}$ of processors which may be finite automata random access machines etc. (see II),
- a set $I \subset P$ of input processors,
- a set $O \subset P$ of output processors and
- a graph G with vertex set $V(G) = P$ and edge set $E(G)$.

A processor P_i is neighbour of P_j if the edge $\{P_i,P_j\}$ is in $E(G)$.

The maximal number of neighbours of some P_i is called the degree of G or the degree of M.

Thus M is specified by a tupel $M=(P,I,O,G)$.

Suppose, the processors are random access machines. Each of them shall have a communication-register. Then such a PC M works as follows: The processors of I contain the input $\bar{x}=(x_1,\ldots,x_n)\in\{0,1\}^p$ for some p.

In one step each processor works for a bounded amount of time dependent on its own configuration and the contents of the communication-registers of its neighbours. M stops when each processor has stopped. The contents of the communication-registers of the output processors at this time are called the result of M started with $\bar{x}$.

The number of steps M executes started with $\bar{x}$ is denoted by $t(M,\bar{x})$.

In this paper we consider PC's which can simulate other PC's. They are called simulation PC's (SPC's).

Their processors are universal (compare (II)) and their input and output processors are not specified. Therefore we can refere to them as a graph.

We say "a SPC M_0 simulates a PC M" if it is possible to initialize the (universal!) processors of M_0 and to fix input and output processors in M_0 such that the resulting PC M_1 computes for each input $\bar{x}$ the same result as M does. If for some k $t(M,\bar{x})\cdot k\leq t(M_1,\bar{x})$ for all inputs $\bar{x}$ then the simulation has time loss k.

Let $M(n,c)$ ($L(n,c)$) be the class of all PC's (SPC's respectively graphs) with n processors (vertices) and degree c.
A SPC M_0 is k-universal for $M(n,c)$ if each PC M of $M(n,c)$ can be simulated by M_0 with time loss k.

2. What can a universal parallel computer look like?

A simple method to simulate a PC $M\in M(n,c)$ by a SPC M_0 is the following:

- Choose n simulation processors $P_{i_1},\ldots,P_{i_n}$ in M_0 and let the j'th processor of M be simulated by P_{i_j}.

- Use the other processors of M_0 to transport the contents of the communication-registers of simulation processors which simulate neighbouring processors of M.

The time loss of this simulation is the time required for the transport between such simulation processors.

Such simulations are of <u>type I</u>. Consider the PC M and the SPC M_0 of figure 1. (The choice of the processors is arbitrary).

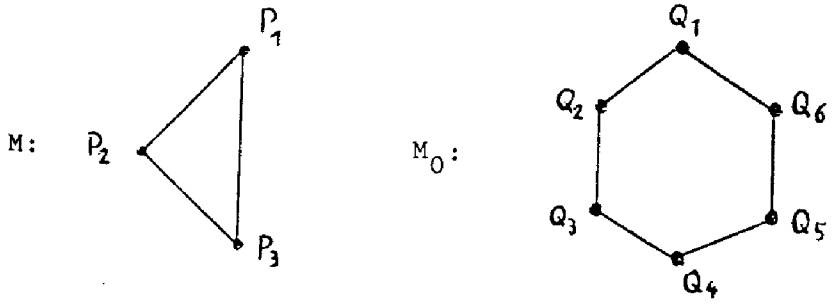

Figure 1

Let Q_{2i}, $i = 1,2,3$ simulate P_i. Then we obtain a simulation of M in M_0 of type I with time loss 2.

Now consider the following simulation. Let Q_i <u>and</u> Q_{i+3} simulate P_i, $i = 1,2,3$. Surely we obtain a simulation of M in M_0 but it has the time loss 1!

Such a simulation is called <u>of type II</u>.

In general, a simulation of M in M_0 of type II [with ℓ representants] is given by:

- n pairwise disjoint, non-empty sets $A_1,\ldots,A_n$ [with $\sum_{i=1}^{n} \#A_i \leq \ell$]*; A_i is called the <u>set of representants</u> of the i'th <u>processor</u> Q_i of M. Each processor of A_i simulates Q_i.
- Pathes from each $P' \in A_i$ to a $P'' \in A_j$, if Q_i and Q_j are neighbours in M. These pathes are called <u>transport-pathes.</u>

*For a set A, $\#A$ denotes the number of elements of A.

The time loss of such a simulation is the length (number of edges) of a longest transport path.

M_0 is called (k,ℓ) universal [k-universal] of type II for $M(n,c$ if each $M \in M(n,c)$ can be simulated of type II by M_0 with time loss k [and ℓ representants] .

We notice that for $\ell = n$ the above is a definition for universality of type I .

Let $c,d \geq 3$ be fixed integers. In this paper we prove the following:

Main Theorem: Let $M_0 \in L(m,d)$ be (k,ℓ)-universal of type II for $M(n,c)$ then there are $a_1, a_2 > 0$ such that $\ell \cdot k \geq a_1 n \log(n)$ or $m \geq n^{a_2 \cdot \frac{n^2}{\ell}}$.

The following 2 theorems are direct consequences of this Main Theorem:

Theorem 1: Let $M_0 \in L(m,d)$ be k-universal of type I for $L(n,c)$ then there are $a_3, a_4 > 0$ such that $k \geq a_3 \log(n)$ or $m \geq n^{a_4 n}$.

Theorem 2: Let $M_0 \in L(m,d)$ be k-universal of type II for $L(n,c)$ then there is $a_5 > 0$ such that $m \cdot k \geq a_5 n \log(n)$.

In the second part of this paper we present an efficient SPC .

In (I) an SPC $\in L(n,4)$ was constructed which was for some $a_6 > 0$ $(a_6 \log(n))$-universal of type I for $M(n,c)$.

We will prove the following:

Theorem 3: There is a SPC $M_0 \in L(2n^2 \log(n), 4)$ and $a_7 > 0$ such that M_0 is $(a_7 \log\log(n))$-universal for $M(n,3)$.

II. Proof of the main theorem

The following is a graph-theoretical translation of a k-simulation of type II with ℓ representants. Let $G_0 \in L(m,d)$ and $G \in L(n,c)$.

A subgraph H of G_0 is called a (k,ℓ)-embedding of G in G_0, if

- $V(H) = V_1 \cup V_2$, $\#V_1 \leq \ell$ and V_1 is the union of pairwise disjoint, non-empty sets $A_1,\ldots,A_n$,
- for neighbouring vertices i and j of G there is for each $c \in A_i$ a $b \in A_j$ such that c and b are connected by a path of length at most k .

A_i is called the H-representant of the vertex i of G in G_0 .

G_0 is called (k,ℓ)-universal for a class A of graphs if there is a (k,ℓ)-embedding for every $G \in A$. Let C_0 denote the circuit with vertices $\{1,\ldots,n\}$ and edges $\{i,i+1\}$ for $i = 1,\ldots,n-1$ and $\{n,1\}$. The set of all $G \in L(n,c)$ which contain the subgraph C_0 is denoted by $K(n,c)$.

We prove the following

Proposition: Let $c \geq 6, c$ even and $G_0 \in L(m,d)$ be (k,ℓ)-universal for $K(n,c)$, then there are $a_1,a_2 > 0$ such that $\ell \cdot k \geq a_1 n \log(n)$ or $m \geq n^{a_2 n^2/\ell}$.

By a simple graph theoretical construction it is possible to extend this proposition such that it holds for all $c \geq 3$. Therefore, this proposition obviously implies the main theorem.

Let from now on $c \geq 6$ be even, d arbitrary.

The proof of the proposition follows this pattern:

- Bound $\#K(n,c)$ from below,
- bound the number $b(n,c)$ of graphs $G \in K(n,c)$ for which there is a (k,ℓ)-embedding in G_0 from above.

As for every $G \in K(n,c)$ there must be a (k,ℓ)-embedding in G_0 (G_0 is universal for $K(n,c)$!) . We obtain that $b(n,c) \geq \# k(n,c)$.

This unequality will prove the proposition.

Lemma 1: $\#k(n,c) \geq n^{\frac{c-2}{2}n} \cdot e^{-cn} \cdot c^{-n}$.

Lemma 2: $b(n,c) \leq (\frac{men}{\ell})^{\frac{\ell}{n}} e^{3\ell} d^{(k+1)\ell} (\frac{d^{k+1}e}{c})^{cn} cn$.

From these two lemmas we can conclude the proposition:

As $b(n,c) \geq \#K(n,c)$, we obtain that

$$\left(\frac{men}{\ell}\right)^{\ell} e^{3\ell}\, d^{(k+1)\ell} \left(\frac{d^{k+1}e}{c}\right)^{cn} cn \geq n^{\frac{c-2}{2}n} \cdot c^{-cn} \cdot e^{-cn} .$$

A ruff estimation guarantees that

$$m \geq 2^{\left(\frac{c-2}{2}\, n^2 \log(n) - 2\log(d)\, k\, n^2 - 2\log d\; k\, \ell\, n\right)^{1/\ell}} .$$

Choose $a_1 > 0$ such that

$$a_2 := \frac{c-2}{2} - 4\, a_1 \log(d) > 0 .$$

Suppose that $k\ell < a_1\, n \log(n)$. Then as $\ell \geq n$ we may conclude that $k < a_1 \log n$.

Therefore we obtain:

$$m \geq 2^{a_2 \frac{n^2}{\ell} \cdot \log n} = n^{a_2 \frac{n^2}{\ell}} .$$

We don't prove lemma 1.

Sketch of the proof of lemma 2: Let $A_1, \ldots, A_n$ be pairwise disjoint, non-empty subsets of $\{1, \ldots, n\}$ with $\sum_{i=1}^{n} \#A_i \leq \ell$ and G_0 as in the proposition.

We need the following sublemmas.

Sublemma 1: Let R be the number of graphs of $K(n,c)$ for which there is a (k,ℓ)-embedding H in G_0 such that for all $i \in \{1, \ldots, n\}$, A_i is the H-representant of the vertex i of G in G_0. Then $R \leq \left(\frac{d^{k+1}e}{c}\right)^{cn} c^n$.

Sublemma 2: Let S be the number of choices of $A_1, \ldots, A_n$ such that there is a (k,ℓ)-embedding H of C_0 in G_0 for which A_i is the H-representant of i in G_0 for all $i \in \{1, \ldots, n\}$.
Then $S \leq \left(\frac{men}{\ell}\right)^{\ell} e^{3\ell} d^{(k+1)\ell}$.

We don't prove these two sublemmas here but show how they yield lemma 2.

As each $G \in K(n,c)$ contains C_0 we only may choose a tupel $A_1, \ldots, A_n$

with the property of sublemma 2 in order to obtain a (k,ℓ)-embedding of G in G_0.

As $A_1,\ldots,A_n$ cannot deliver a (k,ℓ)-embedding in G_0 for more than R graphs of $K(n,c)$, it follows that $b(n,c) \leq R \cdot S$ which implies lemma 2.

III. Proof of theorem 3

We consider PC's whose processors are random access machines.

Let $q = 2^p$ for some integer p.

We need a graph G with the following properties:

1. $V(G) = V_1 \dot{\cup} V_2$, $V_1 = \{a_1,\ldots,a_q\}$.
2. For each a,b, $a \cdot b \leq p$ and for each set of permutations $\{\pi_1,\ldots,\pi_b\}$ on $\{1,\ldots,a\}$ there exist pairwise disjoint pathes from $i \cdot a+j$ to $i \cdot a+\pi_i(j)$ for all $0 \leq i < a$, $0 \leq j < b$. Each of these pathes has length at most $a_8 \log(a)$ for some suitable $a_8 > 0$.

Such a graph is called q-permutation network (q-PN). V_1 is called the base of the q-PN.

Lemma 3: There is a q-PN G_q with $2q \log(n)$ vertices and degree 6.

Such a G_q can be constructed similar to a Waksman permutation network (compare III).

Let $n = 2^{2^{r-1}}$. The following lemma implies theorem 3.

Lemma 4: Let M_n be a SPC whose graph is a n^2-PN. Then M_n is $(a_7 \log\log(n))$-universal for $M(n,c)$ and the processors of the graph to be simulated are simulated by processors from the base of M_n.

Proof of Lemma 4:

- If $r = 1$, the simulation is trivial.
- Suppose the lemma holds for $n = 2^{2^{r-1}}$. We show how the simulation for $m = 2^2 = 2^{2^r}$ works. Let M_m be specified by a m^2-PN G_0. Divide

the base of G_0 into m parts $C_1,\ldots,C_m$ of length m. Each C_i can be identified with a SPC M_i which is specified by a m-PN.

We want to simulate $N \in M(m,3)$. Let G be the graph of N. Consider for each $i \in V(G)$ the set $U_s(i)$ with $s = 2^{r-3} = \frac{1}{4}\log(n)$. Then $\#U_s(i) \leq m^{\frac{1}{2}} = n$. Initialize G_i such that G_i simulates L_i - the induced subgraph of G with vertex set $U_s(i)$ - for all i.

Claim 4: Consider the PC N_i which is the restriction of M on the processor set $U_s(i)$. If the processors of N_i have the same configuration as the corresponding processors of N, then after 2^{r-3} steps the configuration of the processor i of N_i resp. N are still the same.

The proof is obvious.

The information about a processor i is the contents of the communication registers of its neighbours during the last $\frac{1}{4}\log(n)$ steps.

The information is a string of length $a_9 \log(n)$ for some suitable a_9.

The simulation for $2^{r-2} = \frac{1}{4}\log(m)$ steps:

Do two times the following:

1. Simulate every N_i in M_i for $\frac{1}{4}\log(n)$ steps.
2. For all i transport from each M_i the information about the processor R_i which simulates i, to all other processors which simulate i.
3. For all i, all these processors use this information to compute up to the moment when they have the same configuration as R_i has.

Let $T(m)$ be the time necessary for $\frac{1}{4}\log(m)$ steps of such a simulation. As 2) and 3) can be executed in logarithmic time we obtain the recursion:

$$T(2^{2^r}) \leq 2T(2^{2^{r-1}}) + a \cdot 2^{r-1}$$

and

$$T(4) \leq a$$

for a suitable a. This recursion in solved by

$$T(2^{2^r}) \leq ar\ 2^{r-1}$$

respectively $T(n) \leq \frac{a}{2} \log(n) \log\log(n)$.

This yields a time loss of $a_7 \log\log(n)$ for a suitable $a_7 > 0$. a_7 is larger then $2a$, because the time t which M works need not be a multiple of $\frac{1}{4} \log(n)$.

References

1. Z. Galil, W.J. Paul: *A Theory of Complexity of Parallel Computation.* To appear in STOC 81

2. W.J. Paul: *Komplexitätstheorie.* Teubner-Verlag, Stuttgart, 1978

3. A. Waksman: *A Permutation Network.* Journal of the ACM, 15(1) (1968), pp. 159-163