15th Triennial World Congress, Barcelona, Spain

A THEORETICAL MODEL OF A VIRTUAL PROCESSOR FOR PERFORMANCE EVALUATION OF A PARALLEL HETEROGENEOUS SYSTEM

J. M. Martínez-Villaseñor and M. O. Tokhi

Department of Automatic Control and Systems Engineering, The University of Sheffield, UK

Abstract: The performance demands in modern engineering applications have motivated the use of complex signal processing and control algorithms. This in turn has put constraints on computing capabilities of processors. Thus, to achieve efficient implementation of an algorithm a matching of the computing capabilities of processors with the computing requirements of the algorithm is required. This paper presents an investigation into the characteristic behaviour of algorithms for efficient real-time implementation using heterogeneous parallel computing. Accordingly, several characteristic models describing categories of algorithms are considered and a generalised mathematical model for task to processor allocation in a heterogeneous computing framework is developed and demonstrated. *Copyright © 2002 IFAC*

Keywords: Heterogeneous architectures, parallel processing, performance metrics, real-time processing.

1. INTRODUCTION

Previous research has demonstrated that high performance computing can be achieved in two ways: increasing the efficiency of the algorithm through algorithmic design and coding, and increasing the performance of the computing platform by exploiting the available resources in relation to the computing requirements of the algorithm. Daniel and Ruano (1999) affirm that for some simple matrix algorithms operating over small data sets, implementation of parallelisation over hardware as an option to improve performance is not worthwhile. Assertions such as this indicate that characteristic behaviour of algorithms described in terms of execution time on a processor over task size is of great importance in devising a suitable task to processor allocation strategy in a parallel computing environment.

In a large number of applications, a processor exhibits linear characterisation. Anomalies may exist due to run time memory management problems where the processor may need to access lower level memory. On the other hand, there are applications where a non-linear characterisation may be exhibited by a processor. These include (a) logarithmic, where the execution time varies logarithmically with task size, (b) quadratic, which may be noted with relatively small problems. Such behaviour typically arises, for example, in algorithms that process pairs of data items in double nested loops. (c) cubic, which may arise with an algorithm that processes triples of data items, possibly in a triple-nested loop, (d) exponential, that may arise with algorithms where large variations in the load size is expected, (e) polynomial characterisation. This may be used as a general representation of the linear and non-linear models, where a processor may exhibit linear

behaviour with small load sizes, and the behaviour evolves to a non-linear form with larger load sizes. Accordingly, a polynomial model may provide a close fit to such characterisations.

The concepts of speedup and efficiency have extensively been investigated for homogeneous architectures, where all the processors are of the same type. In contrast for heterogeneous systems with diverse processing elements, little work on performance evaluation has been reported. Tokhi and Ramos-Hernandez (1998) have proposed a method of evaluating performance of parallel architectures using a virtual processor model that would achieve a performance in terms of average speed equivalent to the average performance of N processors. This paper looks into extending the method to situations where the characterisation may not be linear.

2. LINEAR CHARACTERISATION

In this section linear characterisation of processors is considered, and performance metrics of a parallel architecture are accordingly developed. Consider a heterogeneous parallel architecture of N processors. To allow define speedup and efficiency of the architecture, assume a virtual processor that would achieve a performance in terms of average speed equivalent to the average performance of the N processors. Let the performance characteristics of processor $i\,(i=1,...,N)$ over task increments of ΔW be given by

$$\Delta W = V_i \Delta T_i \tag{1}$$

where ΔT_i and V_i represent the execution time increment and average speed of the processor. Thus, the execution time increment ΔT_v and the average speed V_v of the virtual processor executing the task increment ΔW can be obtained as:

$$V_v = \frac{\Delta W}{\Delta T_v} = \frac{1}{N}\sum_{i=1}^{N} V_i = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\Delta W}{\Delta T_i}\right] = \frac{\Delta W}{N}\sum_{i=1}^{N}\left[\frac{1}{\Delta T_i}\right] \tag{2}$$

and

$$\Delta T_v = N\left[\sum_{i=1}^{N}\frac{1}{\Delta T_i}\right]^{-1} \tag{3}$$

Thus, the fixed-load increment parallel speedup S_f and generalised parallel speedup S_g of the parallel architecture, over a task increment of ΔW, can be defined as:

$$S_f = \frac{\text{Execution time increment of virtual processor}}{\text{Execution time increment of parallel system}} = \frac{\Delta T_v}{\Delta T_p} \tag{4}$$

and

$$S_g = \frac{\text{Average speed of parallel system}}{\text{Average speed of virtual processor}} = \frac{V_p}{V_v} \tag{5}$$

In this manner, the fixed-load efficiency E_f and generalised efficiency of the parallel architecture can be defined as

$$E_f = \frac{S_f}{N}\text{x}100\% \tag{6}$$

$$E_g = \frac{S_g}{N}\text{x}100\% \tag{7}$$

The concept of generalised sequential speedup can be utilised as a guide to allocation of task to processors in parallel architectures so as to achieve maximum efficiency and maximum (parallel) speedup. Let the generalised sequential speedup of processor i (in a parallel architecture) to the virtual processor be $S_{i/v}$;

$$S_{i/v} = \frac{V_i}{V_v}; \qquad (i=1,...,N) \tag{8}$$

Thus, to allow 100% utilisation of the processors in the architecture the task increments ΔW_i allocated to processors should be so that the execution time increment of the parallel architecture in implementing the task increment ΔW is given by

$$\Delta T_p = \Delta T_i = \frac{\Delta W_i}{V_i} = \frac{\Delta T_v}{N} = \frac{1}{N}\frac{\Delta W}{V_v}; \quad (i=1,...,N) \tag{9}$$

or

$$\Delta W_i = \frac{V_i}{V_v}\frac{\Delta W}{N} = S_{i/v}\frac{\Delta W}{N}; \quad (i=1,...,N) \tag{10}$$

It follows from equation (9) that, with the distribution of load among the processors according to equation (10) the parallel architecture is characterised by

$$\Delta W = V_p \Delta T_p = (NV_v)\Delta T_p \tag{11}$$

Having an average speed of

$$V_p = NV_v \tag{12}$$

Thus, with the distribution of load among the processors according to equation (11), the speedup and efficiency achieved with N processors are N and 100% respectively. These are the ideal speedup and efficiency. In practice, the speedup and efficiency of the parallel architecture will be less than these values. In developing the performance metrics for a

heterogeneous parallel architecture of N processors, the architecture is conceptually transformed into an equivalent homogeneous architecture incorporating N identical virtual processors. This is achieved by the task allocation among the processors according to their computation capabilities to achieve maximum efficiency. For a homogeneous parallel architecture the virtual processor is equivalent to a single processing element in the architecture.

3. GENERAL CHARACTERISATION

To define the speed and behaviour of a virtual processor in a non-linear context, it is necessary to assume that the amount of task executed by a processing element in the parallel system is related to the execution time through a functional relationship as

$$W_i = f_i(t_i) \tag{13}$$

Consider an *nth* order polynomial as a function fitting to this relation;

$$W_i = P_{1i}T_i^n + P_{2i}T_i^{n-1} + ... + P_{ni}T_i + P_{n+1i} \tag{14}$$

where W_i is the task size of processor i. The average rate of change of task size with respect to execution time over the time interval $[t_i, t_i + \Delta t_i]$ is:

$$\frac{f_i(t_i + \Delta t_i) - f_i(t_i)}{\Delta t_i} = \frac{\Delta W_i}{\Delta t_i} \tag{15}$$

The speed of the processor *i* at time t_i is the limit of this average rate of change as ΔT_i tends to 0. Thus the processor's speed is the rate of increment of tasks with respect to time, and is defined as:

$$V_i = \frac{\partial W_i}{\partial t} = W_i'(t) \tag{16}$$

Taking derivative of W_i in equation (14) with respect to time t_i, the speed of each processor V_i is obtained as:

$$V_i = nP_{1i}t_i^{n-1} + (n-1)P_{2i}t_i^{n-2} + ... + P_{ni} \tag{17}$$

The speed of the virtual processor according with equation (15) is now given by

$$V_v = nP_{1v}T_v^{n-1} + (n-1)P_{2v}T_v^{n-2} + ... + P_{nv} \tag{18}$$

where

$$P_{1v} = \frac{1}{N}\sum_{i=1}^{N} P_{1i}$$

$$P_{2v} = \frac{1}{N}\sum_{i=1}^{N} P_{2i}$$

.

.

$$P_{nv} = \frac{1}{N}\sum_{i=1}^{N} P_{ni}; \tag{19}$$

Integrating the function that represents the speed of the virtual processor V_v, within a time interval will yield the task sizes that can be executed by the virtual processor within the time interval;

$$W = \int_0^{t_v} V_v \partial t_v \tag{20}$$

obtaining

$$W = P_{1v}t_v^n + P_{2v}t_v^{n-1} + ... + P_{nv}t_v + W_{v0} \tag{21}$$

Where W_{v0} corresponds to task size at $t_v = 0$. In practice W_{v0} is expected to be zero.

To the equation (9) to allow 100% utilisation of the processors in the architecture, the relation between task increments allocated to processors and the execution time increment of the parallel architecture is given by

$$\partial t_i = \partial t_p = \frac{\partial W_i}{V_i} = \frac{\partial T_v}{N} = \frac{1}{N}\frac{\partial W}{V_v} \tag{22}$$

$$\partial W_i = \frac{V_i}{V_v}\frac{\partial W}{N} = \frac{1}{N}S_{i/v}\partial W \tag{23}$$

and the parallel architecture is characterised by

$$\frac{1}{NV_v}\partial W = \partial t_p \tag{24}$$

Thus, the speed of the parallel architecture represented in terms of the increment of the amount of task in relation with the time required by the architecture is given by

$$V_p = \frac{\partial W}{\partial t_p} = NV_v \tag{25}$$

which gives the definition of the speed of the parallel processor according with V_v as:

$$V_p = N\left(nP_{1v}T_v^{n-1} + (n-1)P_{2v}T_v^{n-2} + ... + P_{nv}\right) \tag{26}$$

Accordingly, using the polynomial characterisation the parallel architecture is represented as

$$W = N(P_{1v}t_p^n + P_{2v}t_p^{n-1} + ... + P_{nv}t_p) + W_{p0} \qquad (27)$$

where W_{p0} represents the task size for $t_p = 0$

4. IMPLEMENTATION

A flexible manipulator system is considered in this paper to test the results given by the theoretical model. The dynamic behaviour of the manipulator has been modelled using FE methods (Mohamed, 1995).

A heterogeneous architecture comprising a T805 (T8) transputer and a TMS320C40 (C40) DSP device has been utilised to implement this algorithm (Shaheed, 2000).

The T8 transputer is a general-purpose 32 bits Inmos RISC processor with 25 MHz clock speed., yielding up to 20 MIPS performance, 40 KB on chip RAM and is capable of 4.3 MFLOPS. It contains 4 serial bi-directional communication links, for interprocessor communication, operating at speeds of 20Mbits/sec, achieving uni-directional data rates of up to 1.7 MB/sec or bi-directional data rates of up to 2.3 MB/sec (Transtech Parallel Systems, 1993).

The TMS320C40 is a 32-bit DSP processor with 40 MHz clock speed, 8KB on-chip RAM and 512 bytes on-chip instruction cache. It is capable of 275 MOPS and 40 MFLOPS. The device possesses 6 parallel high-speed communication links for interprocessor communication with 20 MB/sec asynchronous transfer rate at each port and 11 operations/cycle throughput.

The time required by each processor to execute the amount of task has been obtained from previous implementation. The order of the polynomial functions to characterise the behaviour of the processors in this case is 2. This order has been selected because it is the lowest polynomial function order, which fits the relations.

Two second order polynomial functions are utilised in the general characterisation in order to obtain the virtual model for the parallel implementation.

Figure 1 shows the results in implementing the algorithm on the C40+T8 architecture. The characteristics of the uni-processors, virtual processor and of the corresponding theoretical parallel architecture are also shown.

It is noted that the characteristics of the parallel model are better than each of the single processors. The virtual processor provides an average characterisation of the two processors. It is important to mention that communication aspects are not considered in this model. This parallel implementation assumes that there is no data shared between processors.

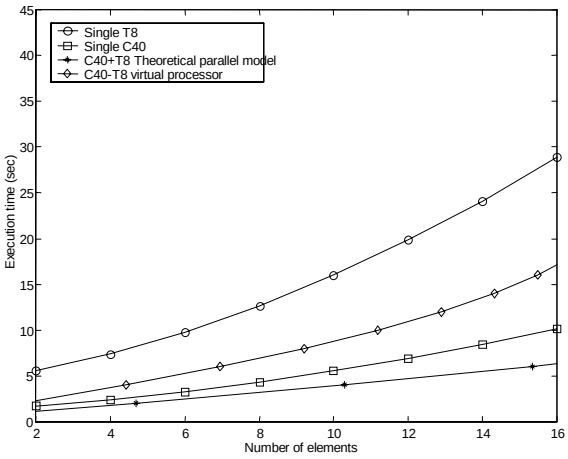

Fig. 1. Execution time to task size relation with various processors.

5. CONCLUSION

An investigation into the development of a generic mathematical model, characterising the behaviour of parallel heterogeneous architectures has been presented. Individual characterisations of processors have been approximated by polynomial functions. Such a strategy allows assessing performance of different processor architectures, and also allocating tasks to processors in a parallel architecture for maximum speedup and efficiency. Future work will look at practical implementation and realisation of the approach on different architectures.

REFERENCES

Mohamed, Z. (1995). A finite element approach to modelling a single-link flexible manipulator system. MSc Dissertation: Department of Automatic Control and Systems Engineering, University of Sheffield, UK.

Daniel, H.A. and A.E.B. Ruano (1999). Performance comparison of parallel architectures for real-time control. *Microprocessors and Microsystems*, **23**, pp. 325-336.

Tokhi, M.O. and D.N. Ramos-Hernandez (1998). *Performance metrics and load-balanced task to processor allocation in parallel architectures*, Research Report 735, Department of Automatic Control and Systems Engineering, University of Sheffield, UK.

Shaheed, M.H. (2000). *Neural and genetic modelling, control and real-time finite element simulation of flexible manipulators*, PhD Thesis: Department of Automatic Control and Systems Engineering, University of Sheffield, UK.

Transtech Parallel Systems. (1993). *Transtech parallel technology,* Transtech Parallel Systems Ltd, UK.