Characterising Computational Kernels: A Case Study

M. J. Zemerly, E. Papaefstathiou, T. J. Atherton,
D. J. Kerbyson and G. R. Nudd

Department of Computer Science
University of Warwick
Coventry CV4 7AL
email: jamal@dcs.warwick.ac.uk

Abstract

We describe the characterisation of an application kernel on a parallel system of two processors. The application kernel is a one-dimensional Fast Fourier Transform (FFT) and the processors used are two T800 transputers. Analytical expressions for the "execution time" for a single and two processors are discussed and used to obtain the performance measure.

1. Introduction

This paper is concerned with characterising a computational kernel (FFT). The aim of characterisation is to find software and hardware parameters required to predict the performance of an algorithm on the selected hardware. Kernels are used because they are the computational core of applications and predicting their performance on a machine gives reasonable estimates of how the actual applications will perform. The characterisation method used in this paper is based on the work of [Hennessy90], [Basu90] and [Nudd93]. The following sections will describe the different stages of the characterisation process. Initially, descriptions of the platform and the kernel will be provided, followed by a description of the characterisation method.

2. Description of the Hardware and the Algorithm

The platform selected is a two T800 (25 MHz with 4Mbyte memory) machine connected by a single link. The data for the operation is kept in external memory and the program in the transputers 4 Kbyte internal memory. The algorithm selected is a partial 1D FFT. The reasons for selection of the FFT are its wide usage, possibility of many forms of parallelism and simplicity of implementation and analysis. The method used to calculate the FFT is based on that described in Rabiner and Gold [Rabiner75] and has the following features: radix 2 implementation, decimation in frequency (DIF), 2048 32-bit random

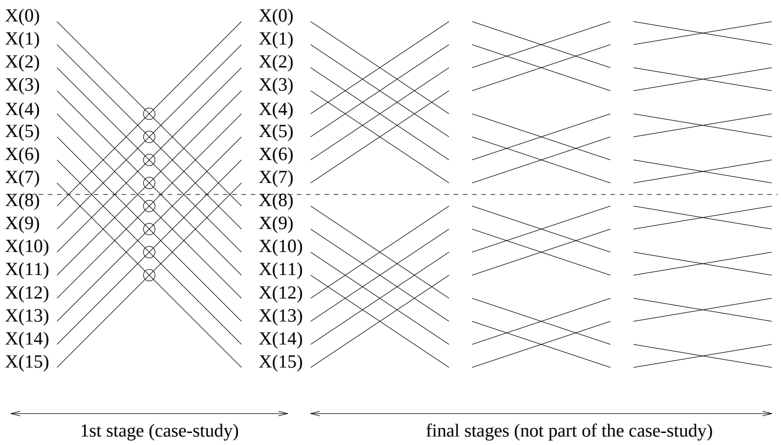

Figure 1. Flow chart of the FFT used (only 16 samples are shown here).

complex input samples and fixed coefficients to compute the same add/multiply in all butterflies. A signal flow chart of the DIF FFT is given in Figure 1.

Only the first stage of the FFT is considered in this case study; this is the only stage that requires data communication between processors. The other stages run as entirely separate serial processes without any interaction, their inclusion does not add to this study. Figure 2 shows the computation and dataflow between the two processors.

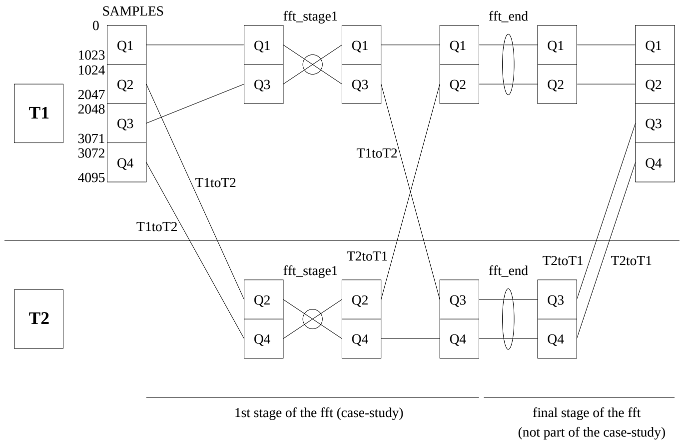

Figure 2. Computation and data flow diagram.

3. Description of the Characterisation Method

The "execution time" is selected as a performance measure. For a program running on a single processor the execution time can be given by:

$$\mathrm{T} = (n_{\mathrm{ex}} + n_{\mathrm{mem}}) \times t \tag{1}$$

where t is the processor clock cycle time, n_{ex} is the number of clock cycles needed for program execution and n_{mem} is the number of extra clock cycles required for accessing external memory. This can be defined in terms of the number of memory accesses per program, $\mathrm{M}_{\mathrm{acc}}^{\mathrm{prog}}$, miss penalty, $\mathrm{T}_{\mathrm{penalty}}$, and miss ratio, $\mathrm{M}_{\mathrm{ratio}}$:

$$n_{\text{mem}} = \mathrm{M}_{\text{acc}}^{\text{prog}} \times \mathrm{M}_{\text{ratio}} \times \mathrm{T}_{\text{penalty}} \quad (2)$$

For parallel systems, algorithms can be considered two ways according to the computation-communication relationship [Basu90]: either overlapped or non-overlapped models. In both models the execution of an algorithm is modelled as repetitive steps where a step consists of a computation followed by a communication. In the overlapped model, computation and communication can be partially or completely overlapped. In the non-overlapped model all the processors do some computation and then synchronise and exchange data. The non-overlapped model resembles the theoretical BSP (Bulk Synchronous Parallel) model [Valiant90]. The non-overlapped model was selected for the implementation of the FFT. Figure 3 shows the characterisation model selected for the FFT. As can be seen from the model shown in figure 3 there is one processing and four communication modules.

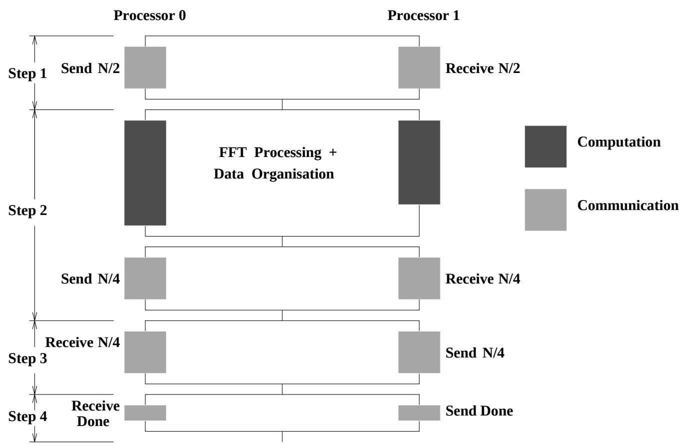

Figure 3. Non-overlap computation-communication model for the FFT.

Assuming perfect load balancing, the execution time T_k for a program running on a k processor system is given by:

$$\mathrm{T}_k = \frac{\mathrm{T}_{\mathrm{p}}}{k} + \mathrm{T}_{\mathrm{s}} + \mathrm{T}_{\mathrm{po}} + \sum t_i \quad (3)$$

where T_{p} is the execution time of the algorithm part that can be parallelised, T_{s} is the execution time of the serial part of the algorithm and initialisation time, T_{po} is the parallel overhead processing required when parallelising an algorithm and t_i is the communication time for step i. The communication time can be estimated from the number of bytes transferred in each step using:

$$t_i = t_{\text{start-up}} + t_{\text{send}} \times \mathrm{B}_i \quad (4)$$

where $t_{\text{start-up}}$ represents the message start-up overhead or latency, t_{send} the pure communication time, which is inversely proportional to the link bandwidth, and B is the number of bytes transmitted between adjacent processors.

Some of the parameters used in these equations are readily available such as the clock cycle time, the number of processors and the memory miss rate which is assumed 1 since the data is stored in external memory. Other parameters have to be measured from the system or have to be assumed. The characterisation parameters required for the program, external memory, parallel overhead and communication are described in the following sections.

4. Measuring Characterisation Parameters

4.1. Measuring Program Parameters

First the number of cycles of the FFT program is obtained using the following procedure.

- Create a software execution structure for the function. For the main (sine and butterfly processing) function the execution structure is:

```
finit (function init)
vinit (Variable init)
L0 (loop 0)
   sine (sine array processing)
   L1 (loop 1)
      L2 (loop 2)
         L3 (loop 3)
            proc (butterfly processing)
```

- Identify the number of cycles for each node of the structure. In order to do that the source code for each part is isolated and the assembly code for it is obtained. The number of times each instruction is executed is counted. The number of cycles for each of the instructions is then extracted using technical data [Inmos89] and the total number of cycles required to execute the node is then calculated.

The following values were obtained from these measurments. C(L0) = C(L1) = C(L2) = C(L3) = 17 cycles, finit = 70 cycles, vinit = 153 cycles and proc = 539 cycles. A separate routine was written to measure the sine time. A thousand measurements were taken and the average time was 195 μs with variance of 0.25. The number of sines required for N data samples is equal to 3N/4.

4.2. Measuring External Memory Parameters

The Transputer external memory is characterised by extra processor cycles per external memory cycle, e [Inmos89]. The time penalty to access the external memory depends on the value of e which is typically equal to (and in this case assumed to be) 5. The external memory is accessed 22 times per butterfly in the FFT.

4.3. Measuring Parallel Overhead Parameters

The parallel overhead within the FFT kernel consists of: organising the data, timing measurements and initialising the links between the two processors. Initialising the links will be discussed in the communication section. The number of cycles for the function used to organise the data for the processors, was obtained from its assembly code. The number of cycles required to copy 4 Kbyte was found to be 2064 cycles. The external memory was accessed 1024 times for a read and 1024 for a write (with 5 cycles penalty for every read/write). The total is 12304 cycles.

For timing, the transputer instruction *ldtimer* was used with high priority to give an accuracy of 1 μs. The number of cycles required for the timer function was obtained from its assembly code and is 57 cycles. The timer function was called 16 times on processor 1. Some other miscellaneous instructions such as difference of times and assigning timing variables were estimated at 200 cycles.

4.4. Measuring the Communication Parameters

The number of cycles required to initialise the links between the two processors is equal to 29 cycles plus the cost of the function *ConnectLink*. A program was written to measure the time for *ConnectLink*. The average *ConnectLink* time was 1130 μs with variance of 14.29.

A further routine was used to measure the communication bandwidth and the communication start-up time between two processors. A block of data was sent from one processor to the other, returned and the total time ($\pm 15 \mu s$) recorded. The block size was increased from 4 to 262144 byte. The communication start-up time was obtained by fitting a straight line to half the time vs block size data and finding the time at block size of 0. Figure 4 shows the relationship between block size and time.

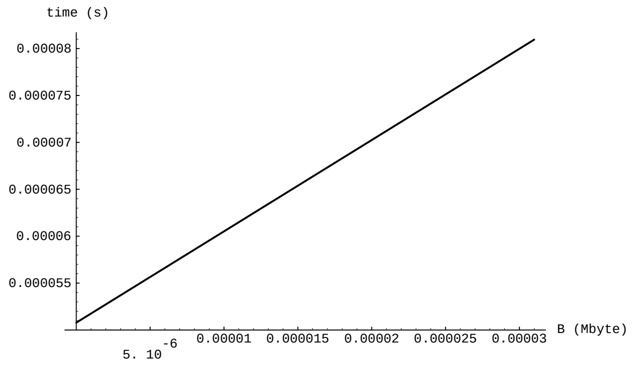

Figure 4. Block size vs time for communication between two processors.

The equation obtained from the fit was (units in seconds and Mbyte):

$$t_i(\mathrm{B}) = t_{\mathrm{start-up}} + t_{\mathrm{send}} \times \mathrm{B} = (51 \times 10^{-6} + 0.97 \times \mathrm{B})$$

Note that the bandwidth is the inverse of t_{send} giving a figure of 1.03 Mbyte/s.

5. Predicted Times

5.1. Predicted sequential Time

The predicted time for the FFT running on one processor is given in equation 1. Here n_{ex} is given by:

$$n_{\mathrm{ex}} = \mathrm{A} + \mathrm{b_c} \times \mathrm{Butterflies} + \mathrm{C(sine)} \tag{5}$$

where A is a constant and equal to clock cycles outside the main processing loops, $\mathrm{b_c}$ is the number of butterfly cycles, Butterflies is the number of butterflies and is equal to N/2 (N is the number of data samples) and C(sine) is the number of cycles it takes to compute the sine array. From equation 2, n_{mem} is given by:

$$n_{\mathrm{mem}} = (\mathrm{b_{macc}} \times \mathrm{Butterflies}) \times 1 \times e \tag{6}$$

where e is the time penalty required to access the external memory and $\mathrm{b_{macc}}$ is the number of memory accesses per butterfly and is equal to 22. The time can be expressed in terms of the FFT size (N) as follows:

$$\mathrm{T} = \left\{ \frac{\mathrm{N}}{2}(\mathrm{b_c} + e \times \mathrm{b_{macc}} + \frac{3\mathrm{C_s}}{2}) + \mathrm{A} \right\} \times t \tag{7}$$

where $\mathrm{C_s}$ is the number of cycles for one sine (195 μs), $\mathrm{b_c}$ is 556 cycles and A is 343 cycles. Converting all units to μs and substituting in equation 7 gives T = 327857 μs.

5.2. Predicted Parallel Time

The predicted time equation for the FFT running on two processors is given in equation 3. For simplicity and because it is small compared with the total time, the effect of $\mathrm{T_s}$ is neglected here. The values of $\mathrm{T_{po}}$ and $\sum t_i$ were calculated to be 3142 and 15379 μs respectively. Substituting these in equation 3 gives $\mathrm{T_2} = 182449 \mu s$.

6. Program Time Measurement

The program time measurement were taken between different steps of the program. The sequential time was measured at 335592 μs and the times on two T800s, in μs, are given in Table 1. The total times in Table 1 were fluctuating in the order of $\pm$ 30 μs.

	Total	Initlink	S/R N/2	Proc	S N/4	R N/4	S/R 1	memcpy
Proc 1	180330	1125	7819	161929	3930	3883	67	1520
Proc 2	181635	2146	8902	161925	3870	4314	52	383

Table 1. Timings for the FFT program on two T800 (25 MHz).

7. Summary

The predicted times obtained from the characterisation model gave similar answer to the measured times. The differences between the sequential and parallel predicted and measured times were less than 4%. The results obtained are summarised in Table 2..

	Sequential	Parallel	Speedup
Measured	335592	180330	1.86
Predicted	327857	182449	1.80
% Difference	-2.3	1.18	-3.23

Table 2. Summary of performance results (measurement and characterisation)

The characterisation method described in this paper will be extended to deal with other computational kernels with larger problem size and on larger systems.

Acknowledgments

The authors would like to thank Francois Manchon of Simulog (France) and Michel Foquet of Thompson Sintra ASM (France) for their comments on this work. This work is supported by the ESPRIT project 6942- PEPS, Performance Evaluation of Parallel Systems.

References

[Basu90] A. Basu, S. Srinivas, K. G. Kumar, and A. Paulraj. A Model for Performance Prediction of Message Passing Multiprocessors Achieving Concurrency by Domain Decomposition. In H. Burkhart, editor, *Proc. of the Joint Int. Conf. on Vector and Parallel Processing*, pages 75–85, Zurich, Switzerland, 10-13 September 1990. Springer-Verlag, Berlin.

[Hennessy90] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann Publishers, 1990.

[Inmos89] Inmos. *The Transputer Databook.* Redwood Burn Ltd, 2nd edition, 1989.

[Nudd93] G. R. Nudd, E. Papaefstathiou, T. Papay, T. J. Atherton, C. T. Clarke, D. J. Kerbyson, A. F. Stratton, M. J. Zemerly, and R. Ziani. A Layered Approach to the Characterisation of Parallel Systems for Performance Prediction. In *Proc. Workshop on Performance Evaluation of Parallel Systems*, University of Warwick, 29-30 November 1993.

[Rabiner75] L. R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing.* Prentice Hall, 1975.

[Valiant90] L. G. Valiant. A Bridging Model for Parallel Computation. *Communications of the ACM, Vol. 33*, pages 103–111, 1990.