EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN – DD /89/13

May 2, 1989

An Evaluation of the Meiko Computing Surface for HEP FORTRAN Farming

S.Booth[1)], R.W.Dobinson[2)], D.R.N.Jeffery[2)], W.Lu[2,a)], K.M.Storr[2)], and A.Thornton[1)]

Submitted to the conference on
Computing in High Energy Physics
Oxford, April 1989

1) Dept. Physics, Univ. Edinburgh, UK.
2) CERN, 1211 Geneva 23, Switzerland.
a) On leave from USTC, Hefei, China.

Results are presented of a systematic evaluation of the capabilities of a Meiko Computing Surface for High Energy Physics FORTRAN farming.

Introduction

The aim of the study was to assess whether the Meiko Computing Surface (MCS) could be used as an off – the – shelf system capable of furnishing significant computational power in the context of High Energy Physics (HEP) off – line data processing. A stepwise approach was adopted in which the following topics were addressed.

- Running FORTRAN programs on a single Transputer node
  - Porting programs
  - Single node performance
- FORTRAN farming on arrays of Transputers.
  - Event farm harnesses
  - External I/O performance
  - System cost/performance

A general assessment is made of the MCS hardware and software environment as delivered, and the conclusions of the study are presented.

System Description

The CERN MCS consists of a host T414 transputer and 8 20 MHz T800 worker transputers each equipped with 4 Mbytes of 200ns memory and 20 Mbit/s links. It is hosted by a Micro Vax II running VMS. A typical configuration is shown in figure 1.

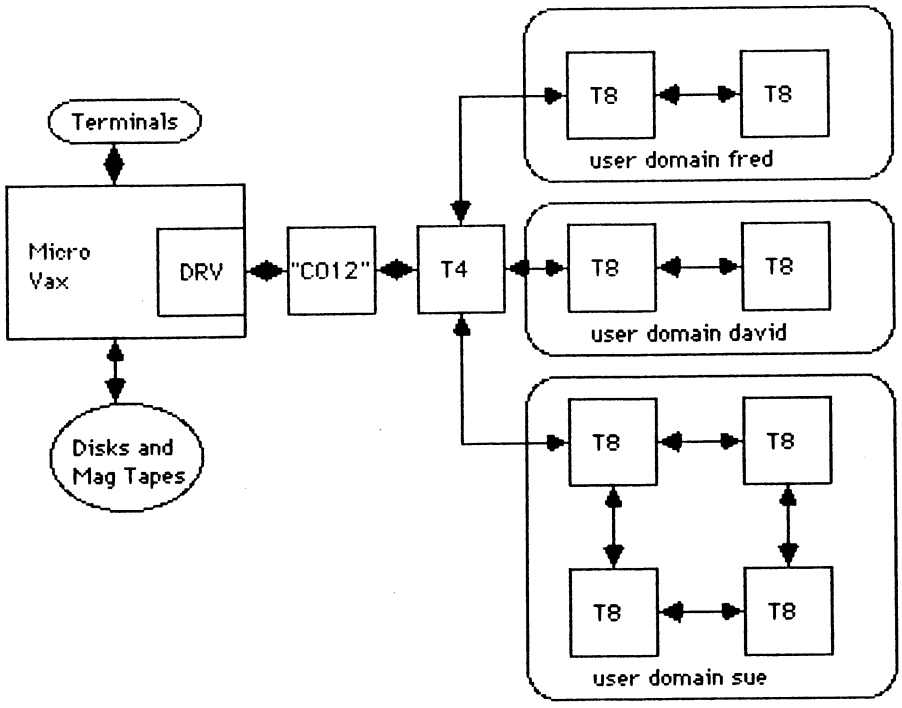

Figure 1: CERN MCS Configuration

The Micro Vax provides terminal access to the MCS and acts as a server for the RD54 disk and STC 2925 6250 bpi 100 ips tape unit.

System features are

- Transputer interconnects are programmable allowing division of the nodes into independent domains for multi−user access.
- Programs are loaded and run using the Meiko M2VCS system.
- Event farm system software is developed under OPS, the Meiko version of the INMOS Transputer development system (TDS).
- Application programs are compiled and linked on the Micro Vax with the Meiko FORTRAN 77 compiler (front end rel. 1.26, back end rel. 1.95) and linker (rel. 1.2).
- Programs can be run in interactive or batch modes.

For the investigation of FORTRAN farming on a larger number of transputers, use was also made of a 128 node domain on an MCS at Edinburgh which is equipped with 10 Mbit/s links.

Running on a single node

An investigation of the problems that could be expected in porting programs to the MCS and estimates of the performance of a single node of the system were made using the following programs.

- CERN benchmark suite − a set of stand alone programs including Monte Carlo, event reconstruction, mathematical algorithms, etc.
- DoDuc benchmark [1]− stand alone benchmark widely used in industry.
- GEANT installation verification and benchmark programs GEXAM1 and GEXAM4. GEANT [2] is an HEP event generation and detector simulation package which makes extensive use of the CERN program library.

Porting programs

The above set of programs represents 160 K lines of FORTRAN, including the library packages which had already been in part ported to the transputer [3]. After making the standard set of changes required for installation on any new system, ie. rewrite of the system dependent library routines, and modification of the Hollerith/character conversion routines to match the order in which the compiler stores characters, few problems were encountered in getting them running [4].

- Four compiler bugs were easily circumvented,
- A few run time errors, due to non−adherence of source code to the FORTRAN 77 standard, were easily corrected.

As a result of this study, working MCS versions of GEANT and the CERN program library packages that it uses are now generally available.

Performance

Table 1 gives the execution times of the CERN benchmark suite of programs on a single node of the MCS, and the ratio of these times to those on a Vax 8600[5] which delivers one CERN unit of computing power.

Tables 2 and 3 give the execution time of the Doduc benchmark and the mean event execution time of GEXAM1 on a single node of the MCS, along with recent measurements made by CERN [6] on other systems for comparison. The Inmos B404 TRAM value for the Doduc benchmark refers to a 20

Table 1: CERN benchmark suite execution times

Program	Time	t(8600)/t(MCS)
CRN1	7.09	0.92
CRN3	1152	0.66
CRN4	116.5	0.42
CRN8	282.8	0.33
CRN9	75.4	0.26
CRN10	66.3	0.36
CRN11	245.8	0.58

MHz T800 with 128 KBytes of 150ns static RAM and 2 MBytes 200ns dynamic RAM using the 3L Parallel FORTRAN compiler. The Caplin Cybernetics system value for GEXAM1, reported at this conference [7], was obtained with the 3L compiler on a T800 with 4 MBytes of dynamic RAM and 128 KBytes of static RAM.

Table 2: DoDuc benchmark execution times

Static/dynamic refers to global saving/no saving of local variables.

System	Time
Cray XMP/48	48
IBM 3090/E	66
Apollo DN10000 (dynamic)	109
Apollo DN10000 (static)	168
DECstation 3100 (static)	189
3081/E	345
VAX 8800	390
MCS T800	**882**
Inmos B404 TRAM	971
IBM PC/RT	1360
Apollo 4000	1850
Micro Vax II	2420
VAXstation 2000	2440

Cecchet et al. [8] have reported that the T800 is equivalent to 1.4 Vax 11/780 [1]based on the Doduc benchmark and three HEP programs compiled with the 3L compiler and run on Inmos boards with 20 MHz T800 and 200 ns memory.

Recent work at CERN by Karlov [9] running a set of 18 benchmark programs on both the MCS and on the Inmos B404 TRAM board using the 3L compiler gives averaged results of 1.5 and 2.2 Vax 11/780 equivalents respectively. The largest of these programs has 2700 lines of code, but most of them are a few hundred lines long.

---

[1] The Vax 11/780 is generally accepted to be equivalent to 0.25 CERN units.

Table 3: GEXAM1 mean event execution time

The CERN and Meiko MCS results differ because Meiko forced the most often called sub-routine into on chip memory.

System	Time	t(8600)/t(System)
Cray XMP/48	3.05	8.52
IBM 3090/E	3.55	7.32
Apollo DN1000 (dynamic)	6.7	3.88
Apollo DN1000 (static)	9.9	2.63
DECstation 3100 (static)	11.6	2.24
VAX 8800	18.9	1.38
VAX 8600	26	1.0
Apollo 3500	58	0.45
Caplin T800	59	0.44
MCS T800 (CERN)	115	0.23
MCS T800 (Meiko)	98	0.27
Apollo 3000	128	0.20

Jeffery [10] has also compared Meiko and Inmos/3L systems and obtains the results given in table 4. At the same time he has showed that compared to a 16 Mhz 68020/68081 the 20 MHz T800 has equivalent integer performance and a factor 2 – 3 better floating point performance.

Table 4

Program	MCS	Inmos/3L	Vax 11/780
Lund	71	109	131
Doduc	900	1020	2112

All the above benchmark results exhibit a wide range of performance and their interpretation should take account of

- the use of the local on chip memory – small programs will benefit more in comparison to large ones, and different performances will be obtained with different compilers. Some improvement can be expected by judicious placement of routines at link time, but this requires run time tracing/analysis and user guidance.

- The superior performance for floating point compared to integer operations, eg. DoDuc is largely floating point.

The discrepancy between the CAPLIN and MCS results on GEXAM1 is being investigated.

It should be noted that at present neither the compiler nor the run time libraries are optimized, and some improvement can be expected.

It has already been shown that the FORTRAN farm can play an important role in both off – line and on – line processing of HEP data by exploiting the natural parallelism arising from the independence of HEP events [11]. In such schemes copies of a program running on multiple processing nodes each handle a complete event under the control of a master. Until now such farms have been custom built and one of the aims of this study was to investigate if an off – the – shelf system could do the job.

Event farm harnesses

A harness is the software that facilitates the distribution of the processing task on an event basis over a number of worker processing nodes. The harness provided by Meiko was found to be inadequate for the needs of HEP data processing, and therefore in order to continue the study it was necessary to develop one. Two approaches were tried [12].

- The classical approach which follows the master/worker philosophy of the CERN/SLAC 3081/E and FNAL ACP farms, in which the application program is split into a master task which performs all external I/O and distributes events to and receives results from copies of the cpu bound part of the program running on workers. In contrast to normal practice, however, in this implementation the master task was split into sender and collector tasks in order to make efficient use of the multiple transputer links. The harness
  - loads the user code into the master and worker transputers,
  - keeps track of free worker processors and schedules events to be analysed,
  - transfers events from the FORTRAN user sender program to a specified worker, splitting them into packets and taking care of intermediate buffering and routing,
  - collects results from workers and transfers them to the user collector program.
- The second approach allows the worker program to read events and write results itself using standard FORTRAN I/O. It has the following features.
  - The master task is replaced by a specially written file server which responds to Meiko's file system protocol commands.
  - Events and results reside on files accessible from all workers.
  - Communication beween the workers and the file server takes place using Meiko's file system protocol via the FORTRAN I/O library.

The implementations of these Occam harnesses are illustrated in figures 2 and 3. It is clear that from a user point of view, the second is more interesting because it requires only minor changes to the original program.

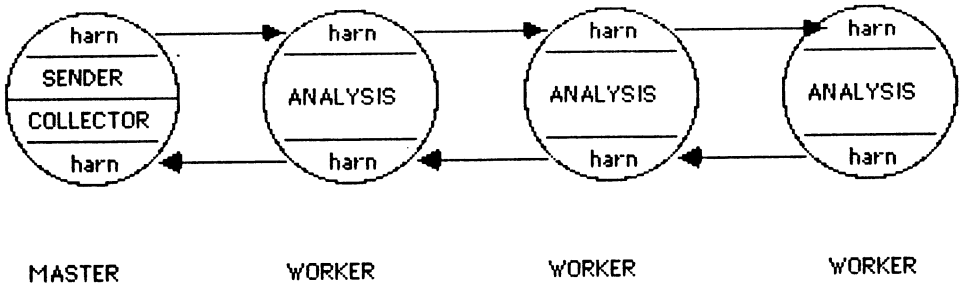

Figure 2: Master/worker harness

Harness performance was measured by simulating an application using programs in which the execution times and event/result sizes varied randomly between predefined limits. No external I/O was performed, ie. the results relate only to the internal data transfer capability of the system. Measurements were made both on the CERN 8 node, 20 Mbit/s link MCS, and a 128 node domain on the 10 Mbit/s link MCS at Edinburgh.

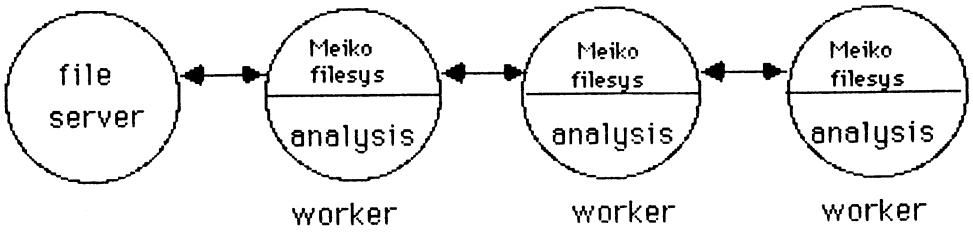

Figure 3: Worker only harness

The master/slave harness significantly outperforms the worker only harness. Tests with both Occam and FORTRAN worker programs indicate that this is mainly due to the performance of the FORTRAN I/O library, discussed in the next section. The use of the worker only harness is effectively precluded at this time. The master/slave results are summarised in figures 4 – 6.

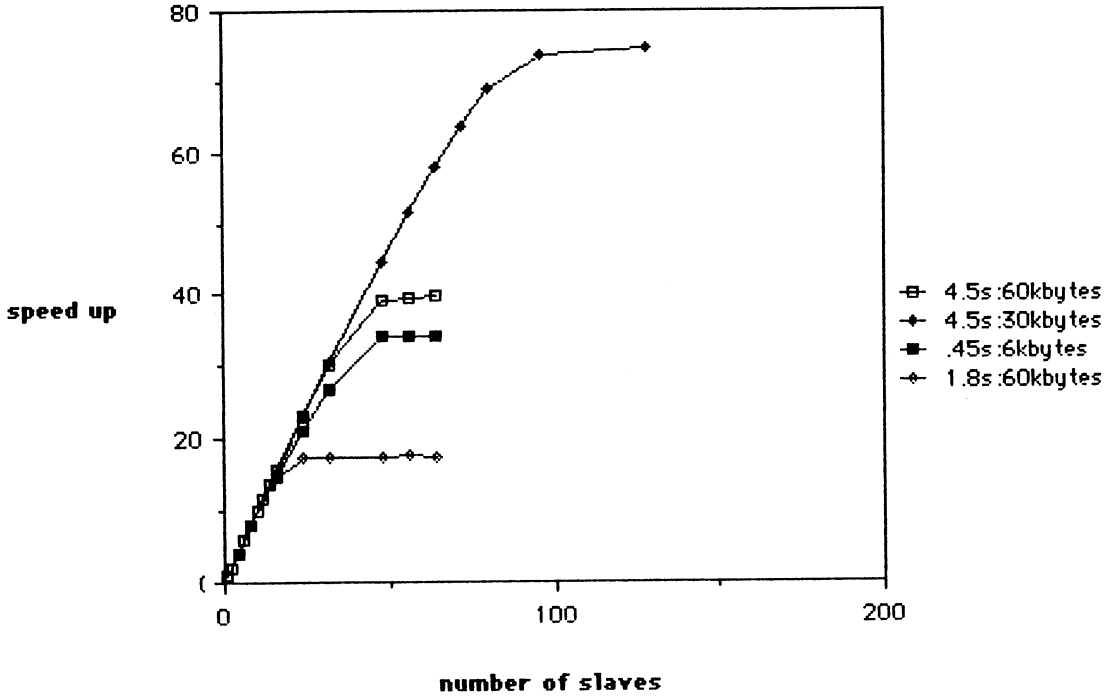

Figure 4. Speedup for different execution times and I/O loads (event = result) using 10 Mbit/s links.

Defining speedup by

Speedup = Single node elapsed time / elapsed time on n nodes

the following observations can be made.

- A linear increase in speedup as a function of number of nodes is achieved when the system I/O rate is unsaturated.
- The saturation limit depends primarily on link speed, although the limit falls for smaller event sizes.
- Significant deviations from linear speedup occur after 80% of the master I/O capability is reached.

Simultaneous bi–directional transfer rates of 1.05 MBytes/s and 0.54 MBytes/s for 20 Mbit/s and 10 Mbit/s links respectively were measured using a simple Occam program. The classical harness investigations showed that data can be transferred along a chain of transputers from sender to receiver FORTRAN programs at 90% of the Occam link speed for a degradation of only 4% in the workers

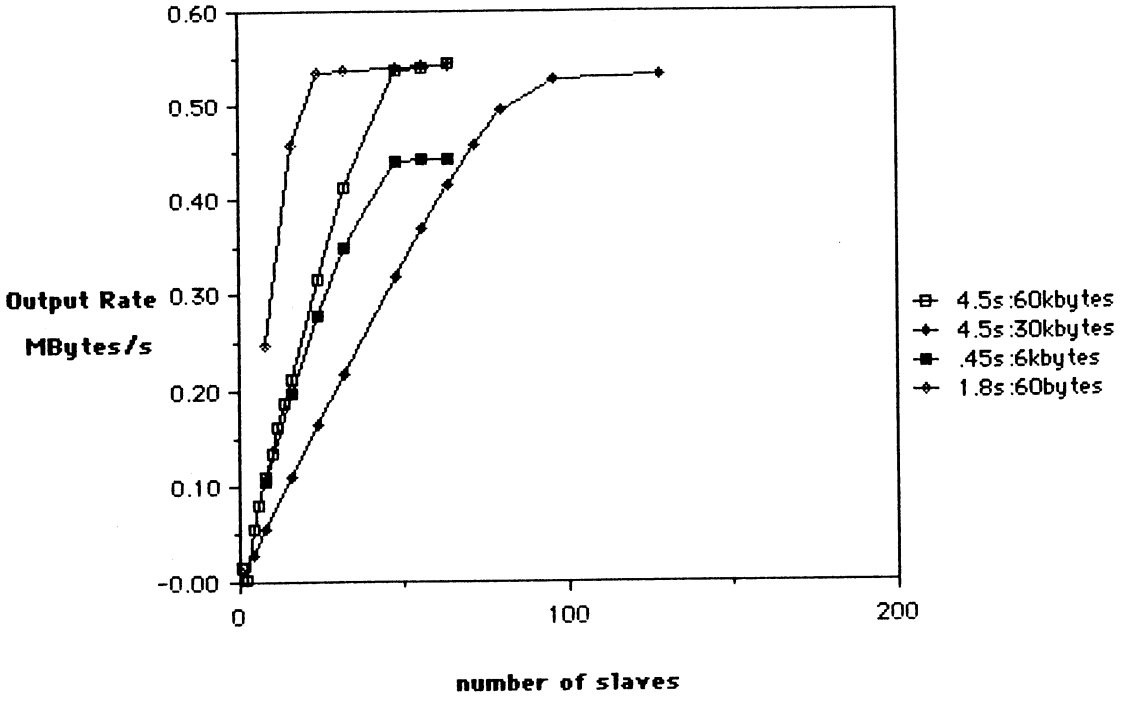

Figure 5. Master output rate for different execution times and I/O loads (event = result) using 10 Mbit/s links.

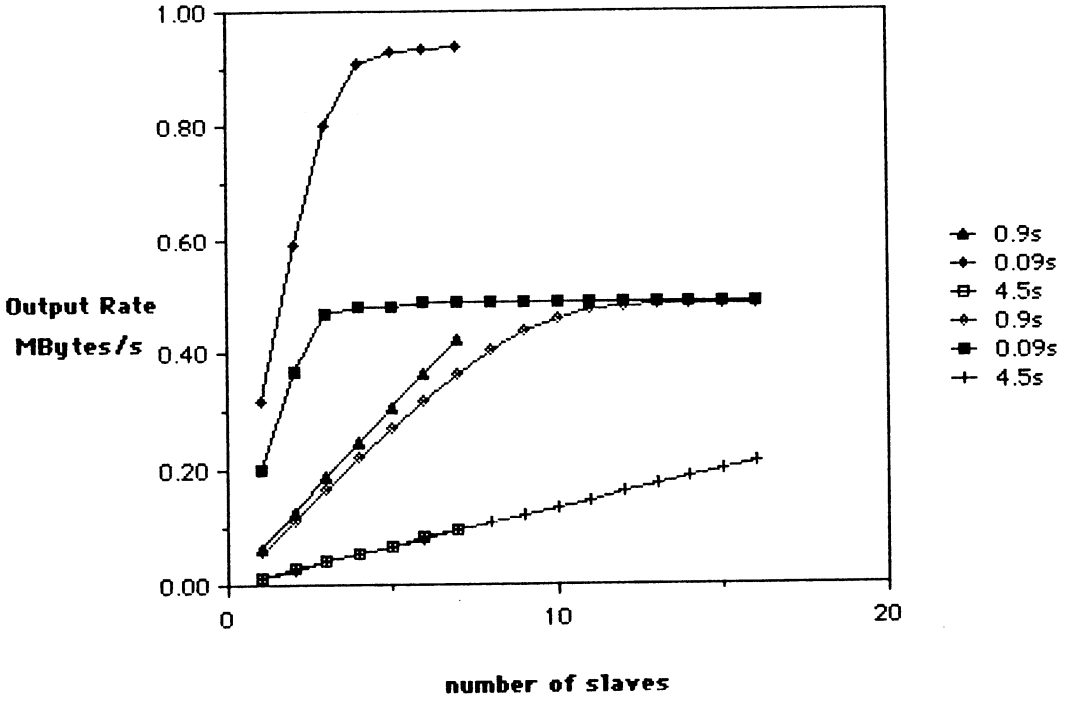

Figure 6. Master output rate for different execution times and I/O loads (event = result) using 10 and 20 Mbit/s links.

computational ability.

Meiko external I/O

Transfer rates from a Meiko node to Micro Vax disks(tapes) were measured to be

- 30(60) KBytes/s from an Occam program, approximately matching FORTRAN read/write performance on the Micro Vax.
- 2(5) KBytes/s from a FORTRAN program. This low rate is due to the poor performance of the I/O library.

One problem encountered is that the Meiko and Vax unformatted file formats are incompatible.

Much higher rates can be expected by attaching the external peripherals directly to the MCS using SCSI interfaces which are available from Meiko.

System cost/performance

Assuming a performance range of 0.25 − 0.5 CERN units for the T800 and that a 90% speedup efficiency could be achieved for a given workload, a 64 node MCS would have a total system power of 14 − 28 CERN units. Table 5 shows the list price of such a system equipped with 1.2 GBytes disk, 2 3480 cartridge drives, and essential software, as a function of the amount of memory/node [13]. It is assumed a host would not be required and that the I/O devices would be attached directly.

Table 5

Memory/node (MBytes)	Cost (KSFr)
1	530
2	610
4	770
8	1080

It should be noted that some HEP off−line production programs now require large amounts of memory, eg. currently up to 12 MBytes for LEP experiments.

It can be seen that if large application programs can be conveniently split over several nodes such that events can be processed without either significant loss in speedup due to mismatch in subevent processing times or an excessive increase in I/O load due to inter−processor communication, then MCS node memory can be reduced and the cost performance becomes more attractive. For example, coarse grain splitting might allow 4 MBytes/node. Finer grain splitting could produce significant cost benefits, but would require much more work from the user.

It should also be noted that the component costs of a 20 MHz T800 and 1 MByte of 200 ns DRAM are the same, ie. 500 SFr. In terms of real estate, 1 MByte of DRAM occupies the same space as a T800, and they can therefore be equated in terms of cost and printed circuit board space. The results of this study show that as long as the application I/O requirements can be supported by the system, additional nodes can easily be added, and therefore, within the constraints of the particular application, there is scope for balancing processing power and memory in a given budget.

The cost per CERN unit of a 64 node, 8 Mbyte/node (256 MBytes total memory) MCS is in the range 40 − 80 KSFr, which can be compared to 50 KSFr quoted for a similarly configured 4 processor 64 MByte Apollo DN10000 system which has a total power of 10 units (neglecting utilisation effi-

ciency effects). For those applications that require less memory, the cost of the MCS becomes significantly more attractive, eg. 20 – 40 KSFr/unit for 1 MByte/node.

General Comments

During the course of this study, an overall impression of the MCS has been built up. The following comments should be taken into account by prospective users of the system.

Hardware characteristics

- Well engineered and reliable – no failures in four months of operation.
- The flexibility of having programmable transputer interconnects is very valuable.
- The supervisor bus allows a useful low speed diagnostic path independent of transputer links.
- The boards used in this study are designed for large amounts of memory, whereas for certain applications, it would be more attractive to trade memory for processing power.
- External I/O rates to the Micro Vax are not adequate for HEP event farming.
- The possibility of attaching external I/O devices directly to the MCS not only enhances the I/O capability of the system, but removes the necessity of having a host and therefore brings cost benefits.

Software environment

From the system programmer's viewpoint

- The MCS was delivered with first generation software adequate for single node applications but not suitable for multi – transputer applications in FORTRAN.
- The Vax – MCS integration is weak and inadequate.
- The OPS system resembles an early version of TDS but with annoying differences. The latest TDS revision D, which is readily available on other hosts, is superior.

Significant improvements in all three areas have been announced by Meiko and the situation will be re – assessed as part of the planned continuation of this study.

From the FORTRAN farmer's viewpoint

- Multi – user access via separate domains is very useful.
- The batch execution facility is a powerful tool.
- As can be expected with early product releases, the compiler and linker are very slow running on the Micro Vax.
- There are no library manipulation tools, eg. add/delete/replace member.
- There is no run time debugger.
- There is only a crude run time tracing/analysis tool which makes it difficult to make optimal use of the transputer on chip memory.

Most of the problems are currently being addressed by Meiko.

Conclusions

As delivered to CERN, the MCS did not meet the requirements for an off−the−shelf FORTRAN farm facility due to the inadequacy of the available harness software. However, it has been demonstrated that efficient event farm harness software can be written and good results have been obtained relating to task processing speedup as a function of number of nodes.

The node processing unit, the 20 MHz T800, with current compilers, appears to suffer a factor of 5−10 speed disadvantage compared to state of the art FORTRAN oriented RISC cpus. This can be off−set against the demonstrated ease of interconnection and the configuration flexibility and potential scalability of the MCS. In terms of cost/performance, a 64 node, 8 MByte/node MCS is comparable to alternative solutions.

The shortcomings of the system software described in this paper, including the lack of a suitable event farm harness, are expected to be overcome in the next release which will be evaluated as part of a continuing program of collaboration with Meiko.

Acknowledgements

We are extremely grateful to Meiko Limited for making the Computing Surface available at CERN.

Julius Zoll provided modified ZEBRA routines and has created standard MCS versions of the KERNLIB and ZEBRA packages.

Thanks are due to Rene Brun for his consultations on GEANT, to Federico Carminati for helping with general library manipulation problems, and to Eric McIntosh for providing help with the suite of CERN benchmark programs.

References

(1) N.DoDuc, *FORTRAN Central Processor Time Benchmark, Framentec, June 1986, Version 13*

(2) R.Brun, F.Bruyant, M.Maire, A.C.McPherson, P.Zanarini, *GEANT 3, CERN/Data Handling Division DD/EE/84−1*

(3) Jeremy M.Carter and Ian Glendinning, *Recent Experience with Transputer based Processor Farms, International Conference on The Impact of Digital Microelectronics and Microprocessors on Particle Physics, Trieste, 28−30 March 1988*

(4) K.M.Storr, *Porting GEANT to the Meiko Computing Surface, CERN/Data Handling Division AC Group Note, 21/2/89*

(5) E.McIntosh, *Private communication*

(6) R. Brun, *Private communication*

(7) J.M.Carter et al., *Transparent Use of Transputers for Off−line Computation, Poster presented at this conference*

(8) Cecchet et al., *Transputer T800 Performance in a FORTRAN environment, International Conference on The Impact of Digital Microelectronics and Microprocessors on Particle Physics, Trieste, 28−30 March 1988*

(9) A.Karlov, *Private communication*

(10) D.R.N.Jeffery, *Private communication*

(11) P.M.Ferran et al., *The 3081/E Emulator, a Processor for Use in On−line and Off−line arrays, Proceedings of the Conference on Computing in High Energy Physics, Amsterdam, 25−28 June 1985*

T.Nash et al., *The Fermilab Advanced Computer Program Multi−Microprocessor Project, Proceedings of the Conference on Computing in High Energy Physics, Amsterdam, 25−28 June*

1985

(12) Stephen Booth, Bob Dobinson, Mick Storr, William Lu and Andrew Thornton, *Harnesses for Running HEP FORTRAN Programs on the MEIKO Computing Surface, CERN/Data Handling Division AC Group Note*

(13) Meiko Limited, Bristol, UK, *Informal UK price information, April 1989*