TECHNOLOGY AND MARKET TRENDS

B PROCTER

Rapporteur: C Phillips

TECHNOLOGY AND MARKET TRENDS

Brian Procter
ICL Corporate Systems Division
Wenlock Way
Manchester, M12 5DR

Introduction

The benefits (and difficulties) of parallel computing have, to some degree, been recognised within the scientific and engineering communities for a decade. Within the last few years there has been an increasing interest and a growth in both the number and the range of parallel systems on the market. The benefits obtained by scientific users are beginning to attract the attention of business users. This paper focuses on the underlying factors affecting the use of parallel computing in business applications and particularly in the central "Corporate Systems" which have traditionally been the province of mainframes. In this arena, most attention falls on the multiprocessor style of parallel architecture which therefore forms the focus of the discussion herein.

The main arguments in the paper are structured under three headings: hardware technology trends, software and standards, market response. To set the scene, the overall requirements of business users are briefly outlined.

Business Requirements

Corporate systems support the core operations of an enterprise: commercial, public or government. Historically, IT was applied first to the day to day operations where the volume and complexity of business threatened to outgrow manual methods. During the eighties the role of IT has been shifting from the support of long established business practices to the enabling of new ways of doing business and of competing. Think of the transformations which have happened within the last decade:-

- to retailing with the concentration of business into a few highly efficient super-groups;
- to manufacturing with fierce world-wide competition forcing rigorous control of costs and quality;
- to banking with the expansion of services and competition;
- electronic inter-business trading;
- "world-scale" operation;
- ....

These business pressures place ever increasing demands on IT systems:-

- storage capacity
- data integrity
- transaction throughput
- management support
- service availability
- flexibility to respond quickly to the needs of the changing business environment
- life-cycle cost
- protection of existing IT investment – human skills, data, applications and hardware

Parallel computing can influence many of the above requirements (positively or negatively). Its most obvious benefits are related to throughput and operational costs where it promises improvements over conventional mainframes of at least an order of magnitude. This level of improvement would support the more demanding workload implied by the changing business. It would also positively affect some of the other requirements by removing the run-time performance and cost obstacles to the use of higher level (but less

run-time efficient) application tools e.g relational databases and fourth generation languages.

The biggest technical barrier to the acceptance of parallel computing is the lack of an evolutionary way of incorporating it into existing corporate IT structures. In particular, the lack of parallel business applications and the mysterious skills necessary to produce them are believed to be real show stoppers. As discussed later, there are actually a number of useful sources of parallelism within commercial workloads and this problem may not be as insuperable as it seems.

The following sections discuss the technology and market trends which continue to influence the acceptability of parallel computing in the corporate business environment.

Hardware Technology Trends

		Semiconductor Family		
		ECL	CMOS	DRAM
Chip Level Speed	Relative Speed Now	4	1	
	Trend % p.a.	+20%	+20%	+12%
Chip Capacity	Relative Capacity Now	1	10-25	100
	Trend % p.a.	+35%	+55%	+60%

Table 1 - Basic Semiconductor Trends

Table 1 displays the basic semiconductor level speed and density parameters. The trends shown have been maintained over many years and are not forecast to vary significantly over the next five years. The points to note from the above table are:-

- The competitive pressure on dynamic RAMs is capacity rather than speed. As a result the performance of dynamic RAMs is growing much more slowly than that of the logic families.
- CMOS logic, benefiting from the investment in RAMs and from its lower power dissipation, is increasing in capacity much more quickly than ECL logic. There is already a factor of 10 difference in semi-custom, more in custom (today semi-custom: CMOS ~200k gates v ECL ~20k gates)
- ECL retains an on-chip speed advantage over CMOS

The "dark horse" technology is GaAs. Whilst it is too early to discern a pattern from the few GaAs parts available commercially, it appears that the technology may be usable in either a high density mode with a capacity approaching that of CMOS but with higher performance, or in high performance mode with similar speed and density to ECL but giving lower dissipation.

The chip capacity available from CMOS already allows the packaging of complete functional units on a single chip, thereby minimising interconnect delays. By contrast, the limited capacity of ECL implies multi-chip functional units. The net effect today is to

reduce the logic speed advantage of ECL from a factor of 4 within a chip to a factor of 2-3 at the logic unit level. Higher speeds incur larger reduction factors.

The large and rapidly growing capacity of CMOS chips has enabled the emergence of complex microprocessor architectures. In terms of instruction processing, the 1991 microprocessor approaches the architectural sophistication of mainframes. In 1992, superscalar RISC chips are expected which, *for some classes of work,* will outperform all but the largest mainframe uniprocessors.

The microprocessor market is highly competitive, with world-wide investment by the semiconductor industry. As well as driving the performance, the competition holds prices down. The combination of high performance today, low price and the promise of future performance growth makes the commodity microprocessor a very attractive choice for parallel computing.

Corporate business application profiles typically exhibit a high ratio of input/output to processing and frequent context switches as the processing resources are moved between the many concurrent users. The result is a high net store traffic rate/instruction. The combination of the very slow improvements in the speed of dynamic RAMs coupled with the rapid speed-up in microprocessor instruction rates is clearly heading for a problem. Mainframes and, increasingly micros, resort to "heavy engineering" at this point with multiple levels of caching. Mainframes have also been forced into the use of expensive static RAM main memory.

The store bottleneck impacts directly on the multiprocessor (MIMD) styles of parallel computing. Compared with uniprocessors, shared store multiprocessors must solve two extra problems:-

- the introduction of a sharing mechanism allowing every processor to access every store without, at the same time, increasing the store access delay
- preventing a further overloading of the store bottleneck through multiple processors accessing the same store

Clearly these problems will become more acute as processor speeds increase still further relative to store. The most common sharing mechanism – the electrical bus – offers relatively little scope for speed improvement due the difficulties of electrical matching. Caches, introduced to reduce loading on the store, create further problems for the bus because the need to maintain consistency amongst them places extra load on the sharing mechanism. The FutureBus standard from IEEE allows networks of buses with overall cache coherency protocols to extend the scaleability of the basic bus. The Scaleable Coherent Interface (SCI) from the same committee seeks to extend the principles further. Performance considerations suggest that it will be necessary to localise processes in these architectures to reduce remote store traffic, but this begins to impact the simple homogeneous process model with which the shared store multiprocessor has won friends.

The distributed store multiprocessor avoids the need for a processor sharing mechanism at the store interface and has a smooth scaling characteristic in respect of store bandwidth. In doing so it trades the latency of message passing and the cost of the associated process switches for the bus delays and synchronisation waits of the shared store model.

Software and Standards

One of the greatest barriers to the use of parallel computing for any application is the difficulty of extracting parallelism. In business IT, the use of powerful processors (eg the commodity microprocessors discussed in the previous section), combined with modest levels of parallelism (say tens to a few hundreds), will provide substantial gains in performance over existing solutions. The difficulty of finding parallelism are likely to be markedly reduced where only modest amounts are needed. There a number of potential sources of parallelism in commercial applications:-

a) natural concurrency arising from many users operating independent tasks on the same system
b) natural concurrency arising from repeating similar operations over a large population (eg payroll)
c) the splitting of applications into smaller functional "chunks", primarily for software engineering purpose
d) specifically architecting an application to exploit parallelism
e) extracting parallelism behind a language interface
f) employing parallelism behind a service interface

a) is what multi-user machines have been doing for twenty years and places no new requirements. Where the tasks are completely independent they are nowadays likely to be distributed to PCs or departmental machines. A more likely case for co-residency is many tasks accessing a common body of data. The usual example here is the traditional transaction processing system. In this case the main difficulties in parallelisation arise in the data access rather than in the application execution. This point is further discussed below. Typical concurrencies for applications will range from a few up to the high tens. TP concurrencies range up to a few thousand.

b) typifies many of the activities of corporate systems. Because of the replicated nature of the concurrency, a mapping to a distributed system model of computation seems natural (eg OSF DCE or UI Atlas). A side advantage of this mapping is that it allows the same application to be run either physically distributed or as a centrally run application on a parallel computer. Typical available concurrencies will range from a few to a few hundred.

c) this approach is attempting to find parallelism as a by-product of methodologies put in place for other purposes. Current development methodologies do not necessarily provide either the containment or synchronisation necessary for parallel operation. A rigorous object oriented development methodology would be a useful step in solving the containment issue but synchronisation (or ordering) would remain an extra burden specific to parallel computing. Regrettably the business world has still a long way to go before it espouses object oriented programming (COBOL++ perhaps!!).

d) is the 'full frontal' approach to parallelism where applications are explicitly designed and implemented for parallel execution. The approach has been used to apply parallel computing to scientific and engineering problems. Implementation can be in a language with explicit parallel constructs (eg OCCAM) or can use operating system interfaces to parallelism mechanisms (eg UNIX processes or POSIX threads) or employ the distributed computing model mentioned in b) above. Careful design will allow this approach to extract the maximum degree of parallelism possible with the computational model used (e.g. threads are lighter mechanisms than UNIX processes and hence can usefully capture smaller grains of parallelism). The problem with this approach is the cost and skills necessary to parallelise an application. In the author's opinion it will not find much use for general business applications although it will be used to a limited degree by software vendors of high value, performance critical packages (eg database managers).

e) here the business programmer is little burdened by the need to make parallelism visible. That job is (nearly) all taken care of by the language compiler and run-time system. This happy state can only be reached through languages which abstract from the basic sequential computational paradigm (*not* COBOL!!). Much work has been done on the automatic extraction of parallelism behind Single Assignment, Functional, Logic, Object and Relational Database languages. Functional, logic and object languages have all seen use in sophisticated decision support applications. Unfortunately, the very nature of these applications makes them relatively rare (albeit high in value). Fourth generation languages are in much more common use and it is a matter of some surprise to the author that they have not received more attention as potential sources of implicit parallelism. (As an aside, a possibly more fruitful generalisation of the last remark is the linking of automatic parallelism to the whole methodology for database application design e.g. Jackson,

Yourdon). SQL, the relational database language, is in common use and is therefore the subject of much attention in the parallel world (see also paragraph (f) below). SQL is, however, computationally incomplete and is usually embedded in either another language or a tool.

f) this is the ideal way for commercial users to employ parallelism – the total black-box approach. Inside the box, the supplier has done all the hard work to extract parallelism using the above methods. Provided that the service interfaces do not reflect parallelism concerns, the commercial user is unaffected. If these interfaces are also existing standards, then there is also a natural evolution route from the use of sequential technology to parallel computing. The trick, of course, is to identify service functions which are sufficiently computationally expensive to benefit from parallel technology and whose interfaces are standard. Fortunately the trend towards client/server business IT architectures has added impetus to interface standardisation. Examples of server functions where large gains can be had from the use of parallel technology are to be found in the companion paper "Parallel Computing for Commercial Applications" by N.P.Holt.

Business users are concerned with protecting their long term investment. Software language and interface standards are important as a secure platform for applications whose life expectancy will often extend over 10-20 years. Open standards are welcomed by the business user because they bring a guarantee of security, interworking, choice of vendors, a wide range of products and lower prices through competition. Parallel computing will need to conform to Open standards to be widely accepted.

Reviewing the means of exploiting parallelism listed above against the standards criteria shows that the interfaces to operating system and distributed system forms of parallelism will be well covered by standards already extant or expected during the next 2-3 years. SQL is the standard relational database language. With the exception of Common Lisp (where extensions are required for parallelism) there are no standards for the newer languages. Parallel extensions to Fortran exist, but its use for business applications is limited.

Market Response

This section looks at the reasons for increased interest in parallel computing from the business community and considers some further factors which will affect its uptake.

Whilst parallel computing has been known and used in the scientific circles for many years as a result of the efforts of suppliers such as AMT, Meiko, Intel etc., it has remained largely invisible to business users. Within the last two years that position has slowly begun to change.

The motivation for the new interest is the pressures of business demand on IT facilities and the concomitant cost of providing those facilities using a conventional solutions. The introduction to this paper outlined some of the dramatic changes which have taken place in the business world over the last few years. Paradoxically, whilst the business community has undoubtedly taken to distributed computing, the effect of many of the changes in the business world have been to reinforce the need for centralised control systems. The problem (as ever) is the cost of building, running and maintaining these mega centralised business IT systems.

Open standards have caused a profound change in the computer market, particularly in the midrange. The business community have discovered a wider choice of systems and have benefited from lower prices. The same line of thinking is prompting them to consider the inclusion of non-proprietary systems as part of their corporate facilities. A door is thus opened for parallel computing, *providing* it fits into the Open standard world.

A third reason for interest is the increased publicity which parallel processing has begun to get. Supplier announcements targeted at the business user and press comment linking parallel machines to the business market place are becoming commonplace. The announcement of a "massively parallel" version of the popular Oracle relational database was a significant step, as has been the recent launch of the NCR3600, both announcements specifically targeted at the commercial user.

A fourth factor attracting the attention of the business user is the appearance of parallel machines in the Transaction Processing Council's benchmarks. These benchmarks detail both performance and price/performance. Parallel computers are beginning to appear in the league tables and are substantiating their claims to be at the top of the performance league, at the same time matching much smaller systems in their price/performance.

Having said that interest is growing, it must be pointed out that:-

- it is growing very patchily: the majority of business IT departments still know nothing about the business application of parallel processing
- there is still a long way to go before more than a handful of brave spirits even start experimenting, let alone using it in anger.

Business IT decision makers are very conservative – of necessity since their investment decisions can directly affect the viability of the main business. They are acutely (and often painfully) conscious of the risk involved in bringing any new project on stream. New technology is seen as an additional risk which they find difficult to evaluate. The risks are made more acceptable where the technology is presented in a form which allows a step-by step approach to its use.

It must also be said that many of the parallel systems available today betray their pedigree in scientific processing or in the price conscious midrange market. They are often limited in features considered important for a corporate role:-

- input/output connectivity
- reliability and resilience
- system management

Further, some of the manufacturers are small and unknown to business users.

The underlying performance and cost benefits of the technology guarantee that these inhibitors will be overcome, but history suggests that it will be a gradual process throughout the remainder of the decade.

Conclusions

Corporate systems have a business need for higher performance and lower cost than are achievable with the historic 25% per annum improvements delivered by the IT industry to date.

Hardware trends indicate that parallelism will be a more effective generator of performance growth than straight technology improvement.

The performance and cost benefits of parallel computing guarantee it a role in business systems.

The use of parallelism for normal multi-user computing and within standard server functions provides a straightforward evolutionary route for business users to exploit parallelism. They are likely to be the first business applications of parallel computing.

The use of parallelism within general applications remains a problem. It seems unlikely that commercial users will bother with the complexities associated with explicitly maximising parallelism. Of more use would be a straightforward methodology for exploiting the natural concurrency and modularity present. More work is required on

methods/languages which give rise to implicit parallelism, especially those directed to operational tasks.

There is poor understanding of the significance, applications, limitations, methodology or practises of parallel computing within the business IT community.

Whilst interest in parallel computing within the business IT community is undoubtedly growing, there are currently too many uncertainties and omissions for early, large scale acceptance. Rather, parallel computing will be introduced gradually throughout the decade, working alongside existing architectures. It could, nevertheless, become the supplier of the major part of the corporate IT horsepower during the latter half of the decade by acting as the engine for a relatively small set of high usage operations.

DISCUSSION

Rapporteur: Chris Phillips

Professor Hall commented that it was right to emphasise the evolutionary approach to parallelism, but that the revolutionary approach was also of interest. Mr. Procter agreed that the topic was worth discussing, if only from a theoretical point of view. The problem is that it can be difficult to think in such terms.

Professor Tanenbaum returned to Mr. Procter's earlier comments on the way commercial computing was going. The `brontosaurus' approach of a large central system was in decline, to be replaced by desktop workstations. The need for parallelism in such an environment was thus in question. Mr. Procter contended that there are coherency and management problems with distributed systems. Some local processing can be done on a desktop machine, but there is the need to maintain data centrally, which is then available to all users.