Base One International Corporation
44 East 12th Street
New York, NY 10003
212-673-2544
info@boic.com
www.boic.com

The Base One Grid Computing Architecture

Introduction

This white paper describes key features of Base One's distributed processing architecture, and explains how its design benefits large-scale business and scientific applications by reducing the cost of application development and operations, while increasing system performance and capacity.

Beyond the easily quantifiable metrics, but no less importantly, it is argued that this architecture forms a practical basis for building systems that are more robust, maintainable, and secure. The central conclusion is that the technology of "grid computing" has been brought out of the laboratory, with broader and more immediate applicability to commercial data processing than is generally understood.

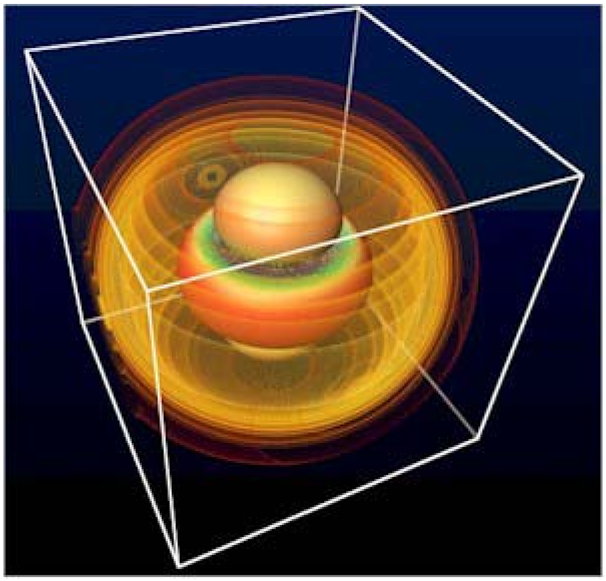
Grid computing

Like a power grid, a computational grid is based on the idea of a vast, dependable, shared resource that can be plugged into and used conveniently and economically, on demand. Its appeal lies in the economies of scale, with potential for still greater efficiency by tapping into the latent processing power of many otherwise idle computers.

Such distributed processing systems have been regarded as a somewhat theoretical ideal, more a topic of specialized scientific research than something of general commercial importance. A major deterrent to the pursuit of this ideal is the prospect of greater complexity, which adds cost, slows down development, undermines reliability, and makes systems harder to maintain and use. These considerations would certainly appear be serious impediments to the use of grid computing in routine business applications.

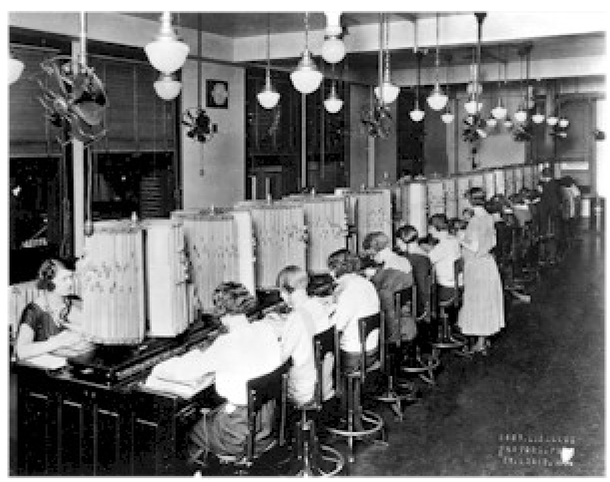
Nevertheless, the concept's feasibility has been demonstrated in at least a few conspicuous examples, SETI@home being one of the most frequently cited. Although esoteric grid applications have been the first to materialize, commercial business applications are now emerging as substantial beneficiaries of this technology. The key to broader acceptance of grid computing is low cost systems that avoid the hazards of increased complexity.

Multi-processing in scientific and business applications



Traditional differences between scientific and business applications have a bearing on the ease of adapting them to multi-processing. Scientific programs frequently correspond to an underlying mathematical model, where a simple, efficient design assumes it has plenty of main memory and many tightly-coupled, lightweight processing units. The straightforward, but costly way to speed up such an application is to buy or build a bigger supercomputer, with custom programming to optimize the algorithms for parallel processing. In those cases where operations are "vectorizable", programming for parallel execution is relatively easy, but more typically it is difficult to transform a model into parallel form, because of interdependencies in the required mathematical calculations.

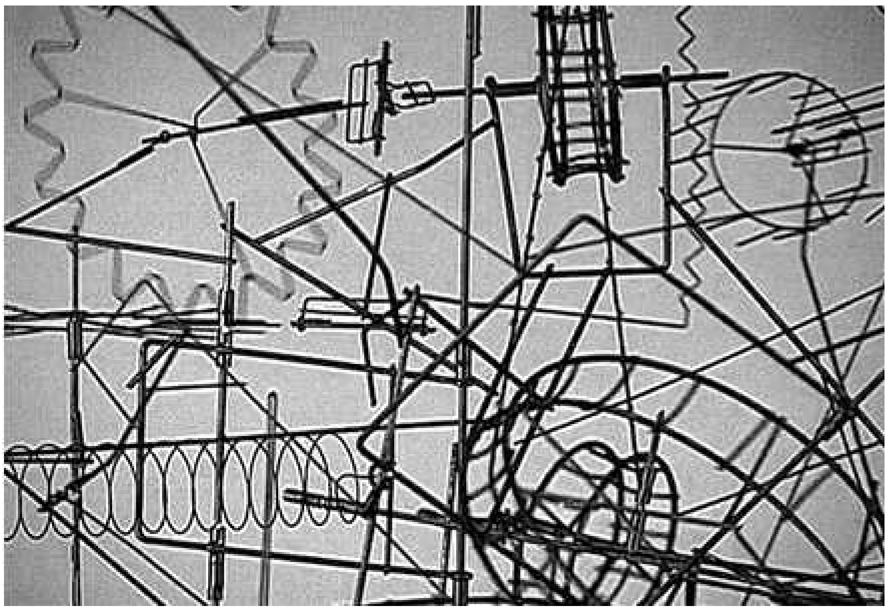
Business applications, on the other hand, generally perform a number of related database functions, pushing the limits of large amounts of data and many concurrent users. The simple way to speed up or enlarge most business applications is to get bigger, faster database servers, and faster networks. Again, this can be a costly proposition, because an existing large database and high-performance network infrastructure is a major capital investment not easily replaced. Aside from what may be imbedded in their database servers and networking components, few business applications fully exploit the potential efficiencies of multi-processing.



Historically, business applications have always had a propensity for decomposition into parallel subtasks. This follows from the natural breakdown along independent dimensions, e.g. by customer, employee, product, account, transaction, date, geography, etc. Of course there are constraints on the sequence of job steps, but within each step lies the potential for substantial speedup through parallel execution, more so as the application grows larger. Ironically, while it is common practice simply to "put more people on the job", the idea of multi-processing - putting more CPUs on the job - has been slow to catch on.

The problem of complexity in multi-processor systems

To a limited extent, advances in operating systems, compilers, and processor technology itself can provide multi-processing benefits automatically, without burdening programmers who develop applications. For example, parallel processing is relatively effortless when several jobs have already been designed to execute as independent programs, or when using library functions that take advantage of multiple processors transparently.

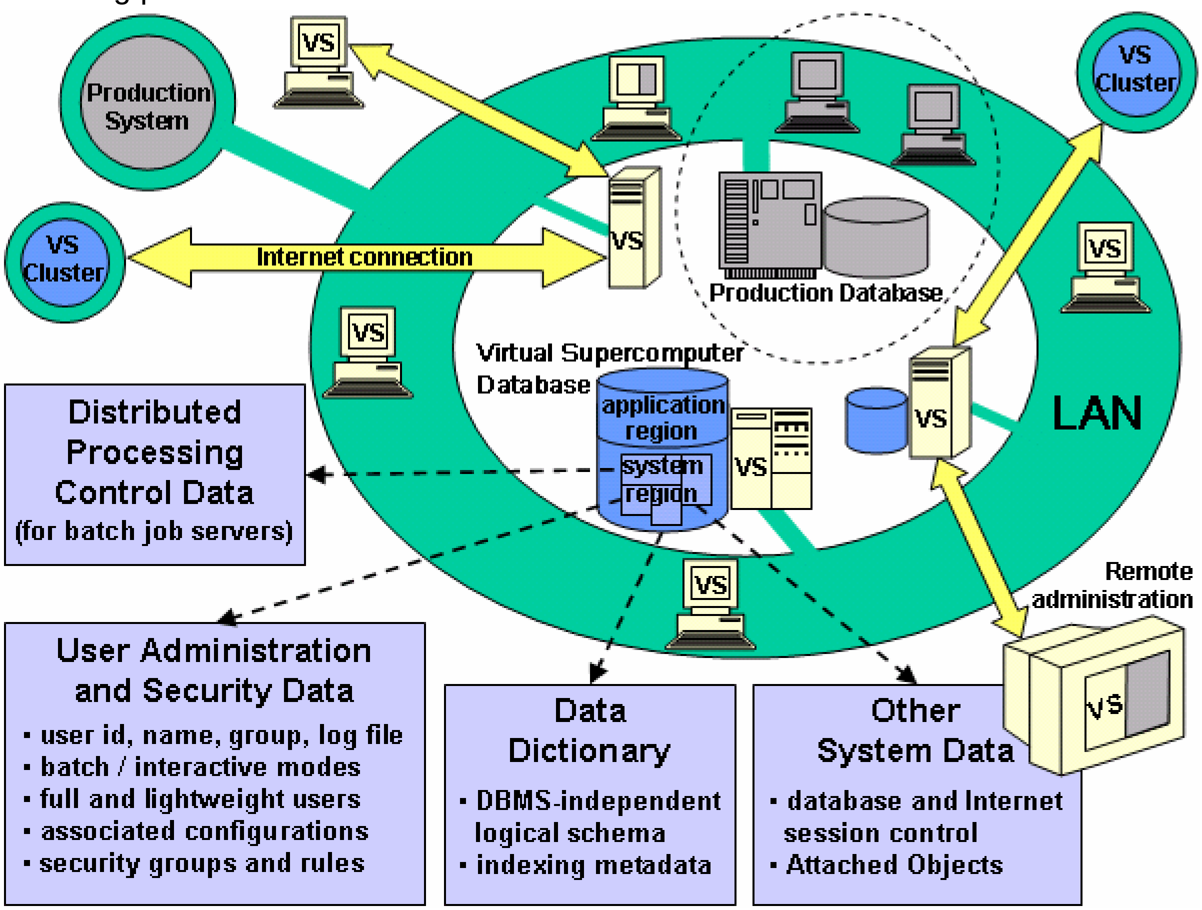


Unfortunately, large applications which stand to gain most from multi-processing tend to be the least likely to reap those gains without extra programming. These applications do a lot of processing, but the operating system sees the program as a single, indivisible job. It then falls upon application developers to design for optimal parallel execution, a potentially challenging programming task. Not only is such programming costly, it can greatly complicate the system's design, making it more fragile and harder to maintain.

Base One and the Virtual Supercomputer

Base One's work on distributed computing evolved over more than ten years of developing tools for building commercial business applications. It should therefore come as no surprise that Base One uses terminology like "batch jobs", dating back to early business systems. The concept of a batch job, however archaic it may seem, still has an important place in modern data processing: it is the abstraction of a logical unit of work, apart from its embodiment in a particular program running on a particular machine. As it turns out, this conventional representation of business applications in terms of discrete job steps is another characteristic that facilitates adaptation into a multi-processing environment.

The Base One grid architecture revolves around the model of a "virtual supercomputer", comprised of loosely-coupled "batch job servers", which asynchronously perform tasks that are specified and coordinated through *database-driven* control structures. The model is *virtual*, because it doesn't entail the actual addition of a physically separate machine. Rather, it uses the available processing power and resources of ordinary PCs and database servers, which may already exist and continue to work in their previous roles. The result is a form of supercomputer, because it presents itself as a single, unified computational resource that can be scaled to virtually unlimited capacity and processing power.



One of the key features that distinguishes Base One's architecture from other grid computing models, as well as conventional mainframes and "cluster" supercomputers, is the central role of a database in the Virtual Supercomputer. The significance of this database-driven design is that it greatly simplifies synchronizing the work of multiple processors, while providing a highly scalable solution to the problem of large-scale multi-processing.

Furthermore, it leverages the power of modern commercial DBMS technology to assure that fundamental matters of reliability and security are seriously addressed, without complicating the design. There is no critical point of failure, because the central component is a *database*, with ample provisions for backup, duplexing, and other well-established database reliability mechanisms.

Simplicity through symmetry

Symmetries in Base One's design reduce complexity, both in the grid architecture itself, and in the applications that run on it.

Each of these aspects of symmetry translates into less code, because there are fewer special cases to contend with:

- equivalence of batch job servers
- unified client/server environment
- network agnosticism
- back-end DBMS independence

equivalence of batch job servers

Base One's data-centric design eliminates any need for the usual "master/slave" distinction (an asymmetry) among processors that do the work. That is not to say that all the machines must be identical, or even similar, but from the standpoint of the system as a whole, batch servers are interchangeable to the extent that they have the required resources and permissions to perform a given job. This logical equivalence not only contributes to robustness, but also makes it possible to reduce the complex problem of scheduling and coordinating disparate processing resources to a simple table-driven implementation.

unified client/server environment

Both client and server-side programs use a common set of functions, regardless of the particular version of Windows (client or server) that runs on each computer. This allows low-end PCs to participate as "servers" for the purposes of the Virtual Supercomputer, without the expense of upgrading those machines to a true Windows Server operating system, assuming that is even possible. From a programming standpoint, the common framework makes it easy to put a piece of logic on either the client or the server side of an application, according to optimal performance criteria, not restricted by the dictates of asymmetrical client vs server programming environments.

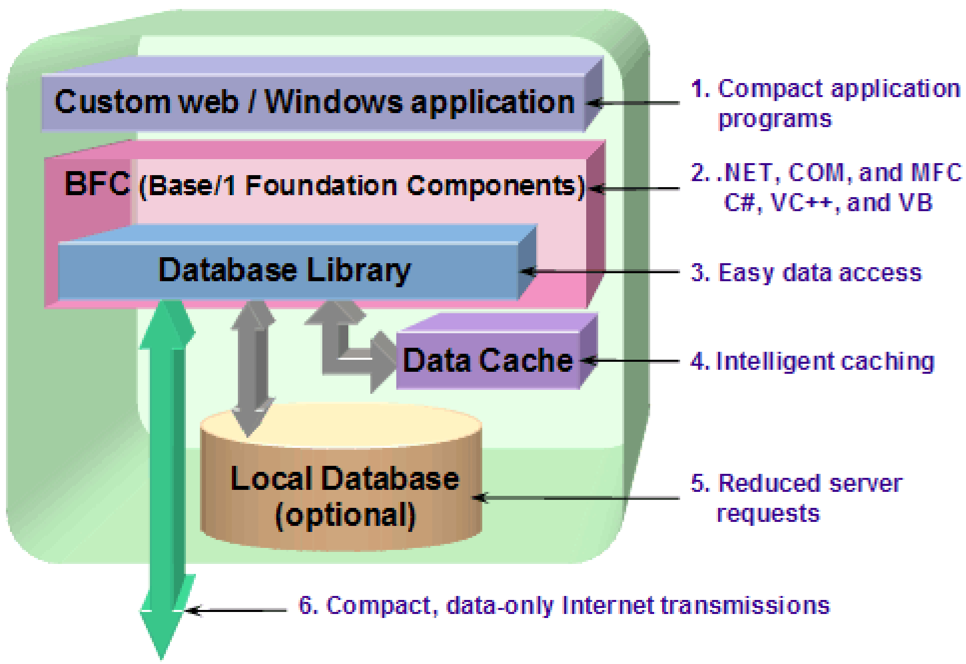
network agnosticism

The core database access layer automatically manages issues of database locality, including local caching, access across a LAN, WAN, or the Internet. Thus applications using the Base One Foundation Classes (BFC) have no need to know what sort of network access they are using, if any at all. A Base One application may be developed and tested on an isolated notebook computer, and then deployed in an entirely different network configuration without any reprogramming whatsoever.

back-end DBMS independence

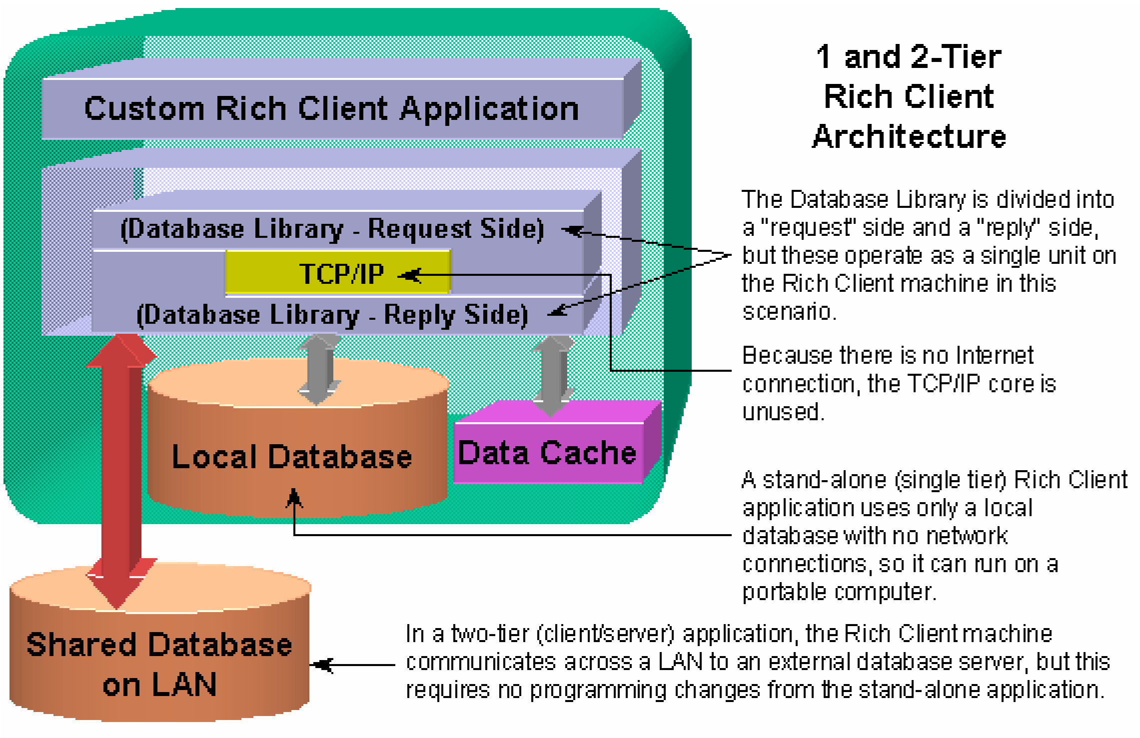
Last, but not least, the database access layer also shields applications from the complexity of differences between commercial database management systems. Base One's architecture makes it easy to construct a system that works, without reprogramming, against any of the major database systems (Oracle, MS-SQL, DB2, and others), or with any mix of those products.

Rich Client architecture

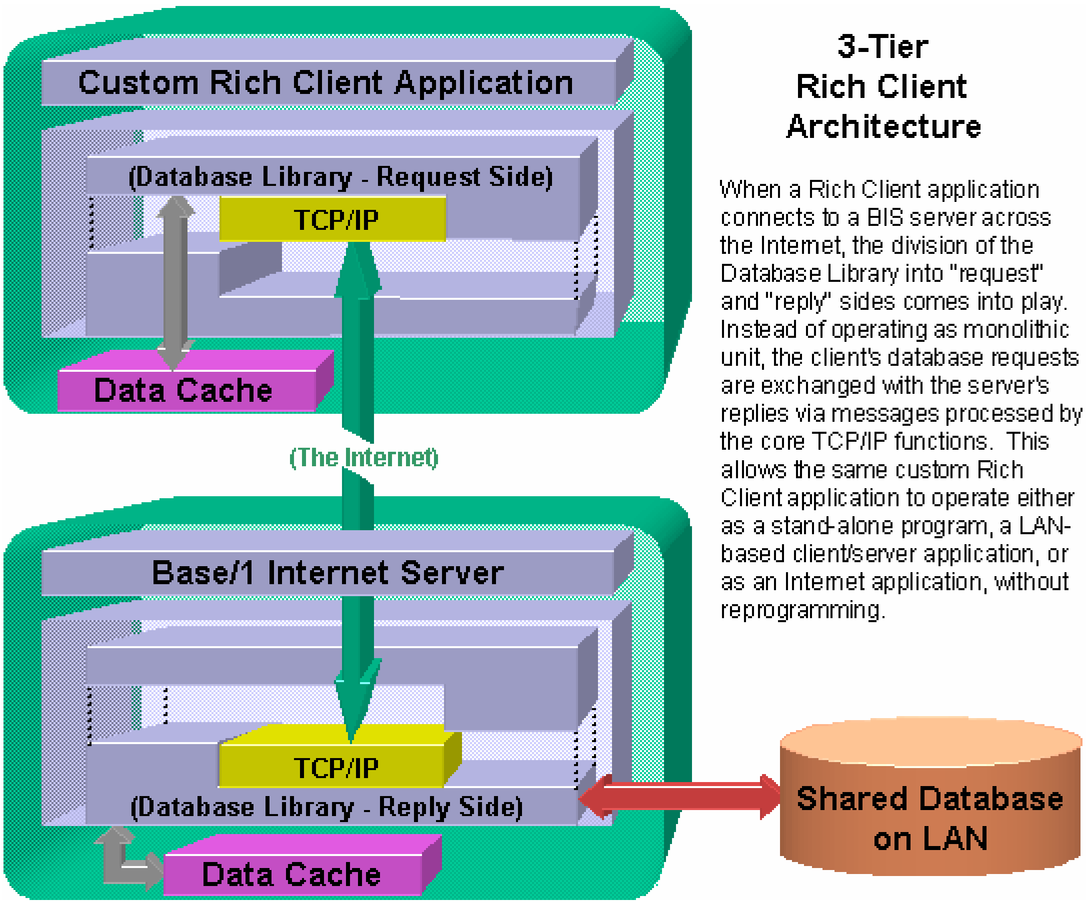


Base One's underlying "rich client" architecture originated in the course of developing client/server systems. From the outset, the design objectives were to:

- localize DBMS dependencies, so that a single application design could operate efficiently against databases from various vendors, without reprogramming
- provide the performance boost of intelligent caching, without placing the burden of this optimization on application programmers
- allow a larger share of processing to be performed on client machines, to minimize the load on the network and central database servers



In the data-centric Rich Client model, network communication is handled entirely within the core database classes. This frees the rest of the architecture from having to be concerned with networking details, i.e. it achieves the simplification of "network transparency".



Another benefit of this design is the way it neatly extends to a 3-tier Internet architecture. By inserting a kernel of peer-to-peer TCP/IP functions and splitting the database functions into two complementary sides, a Rich Client can be physically divided across the Internet, without modifying the original application.

This idea, a central feature in Base One's pending U.S. Patent, was pivotal to the development of an elegant grid computing architecture based on the Rich Client model.

The Rich Client model, by virtue of its symmetry, enhances flexibility and simplifies programming of database applications in general, even in the absence of a full-blown grid environment. The model's natural evolution from client/server to 3-tier Internet applications, and then to becoming the basis for a solid grid architecture has been a testament to its soundness.

Putting the model to work

It would be nice if one could just take an existing application, plop it into a souped-up grid computer, and effortlessly attain a heightened level of efficiency. In some cases it really is that easy, but for the vast majority this is fantasy. More realistically, the problem is to have the software tools that make is as easy as possible to adapt existing applications into a form that is amenable to multi-processing, and to create new, full-featured grid applications with those same tools. That is the objective of Base One's distributed computing software and application development tools.

Base One employs a highly symmetrical, data-centric design to achieve a clean, scalable grid computing architecture. Having evolved from the perspective of business data processing, this architecture is particularly appropriate to the database orientation of business programs, and the need to preserve business logic. Further contributing to the ease of building grid-enabled business applications, Base One provides a feature-rich database application development framework, BFC, which includes security and administration, reporting, data dictionary, utilities, and other general purpose components. These tools have proven their worth in a number of large-scale commercial applications, such as a $50 billion securities custody system that handles millions of financial transactions for Deutsche Bank, and other examples.

Although Base One's grid computing architecture has a distinctly business-oriented slant, it is well-suited to scientific data processing. One reason is a growing recognition of the need for very large databases and transaction processing techniques, not just raw computational power, in connection with scientific applications.

Another notable point is that as one constructs more and more massively parallel programs, the problem of complexity, the nemesis of reliability, becomes increasingly significant. Thus the distinction between business and scientific applications blurs, much to the benefit of scientific computing, because efficient, robust, secure, database technology has been highly refined under the auspices of commercial business enterprise.

Summary and conclusion

Grid computing promises to become a source of cheap, abundant computing power, but that promise seems distant for most applications. As hardware costs have declined, e.g. with clustered "blade" servers and commodity PCs, multi-processing has been brought within reach of a larger audience.

Hardware advances, however, are not enough to fully realize the potential benefits of parallel processing. The challenge is to overcome the complexity of building large applications that run efficiently, reliably, and securely in a grid environment.

The simplicity and power of Base One's "shrink-wrapped" solution holds the promise of bringing grid computing to a much wider audience, for both business and scientific applications.

References

Grid Computing

- Internet Computing and the Emerging Grid, Foster, I., *Nature,* (12/7/2000)
- Harnessing the Power of Grid Computing, *NPACI*, **5**(10), (5/16/2001)
- The Grid: A New Infrastructure for 21st Century Science, Foster, I., *Physics Today*, **55**(2) 42, (2002)
- Survey: Interest in grid computing grows, Thibodeau, P., *Computerworld*, (1/6/04)
- Getting down to grid computing, Scannell, E., *Info World,* **4**(1) 16, (1/16/2004)
- Grid Computing's Promises And Perils, Conry-Murray, A., *Network Magazine*, (02/05/2004)
- Grid Computing in the Enterprise, Coffee, P., *eWeek*, (2/9/04)

Additional Links, References and Acknowledgements

- Arthur Ganson - Machines at http://www.arthurganson.com/
- Bell System - Historical Photos at http://www.bellsystemmemorial.com/
- M. C. Escher - Picture Gallery "Symmetry" at http://www.mcescher.com/
- Scott Camazine - Self-Organized Patterns in Nature at http://www.scottcamazine.com/
- Wilson Bentley - Snow Crystal Display at http://www.jericho-underhill.com/