2006

Net-Centric Computing: The Future of Computers and Networking

Arun K. Somani
Iowa State University, arun@iastate.edu

Shubhalaxmi Kher
Iowa State University

Net-Centric Computing: The Future of Computers and Networking

Abstract

Futuristic computers will only be thought of in the context of their ubiquitous connectivity. Net-centric computing isn't communications or networking per se, although it certainly includes both. With the changes in the computing and networking environment we need a different paradigm for distributed computing. The area of net-centric computing encompasses the embedded systems but is much larger in scope. In the near future, many hardware devices will be interconnected in large and highly dynamic distributed systems, using standard communication protocols on standard physical links [1][3][5][6][7][8]. Such types of systems exist only for computers interconnected by TCP/IP networks, or for hardware devices interconnected in small areas by using specific protocols for the physical link, such as Bluetooth, Ethernet or X-10.

In this paper we review Net-Centric computing in the perspective of Hardware requirements, Embedded system design, Middleware, Control, IT and provide an insight into the issues and challenges ahead.

Keywords

Distribute Computing Environment, Common Architecture, Idle Resource, Embed System Design, Retail Electricity

Disciplines

Systems and Communications

Comments

This is a post-peer-review, pre-copyedit version of a proceeding published as Somani A.K., Kher S. (2006) Net-Centric Computing: The Future of Computers and Networking. In: Madria S.K., Claypool K.T., Kannan R., Uppuluri P., Gore M.M. (eds) *Distributed Computing and Internet Technology. ICDCIT 2006*. *Lecture Notes in Computer Science*, vol 4317. The final authenticated version is available online at DOI: 10.1007/11951957_2. Posted with permission.

Net-Centric Computing: The Future of Computers and Networking

Arun K. Somani and Shubhalaxmi Kher

Dependable Computing and Networking Lab
Iowa State University
Ames, Iowa 50011
Email: {*arun, shubha*}@iastate.edu

Abstract. Futuristic computers will only be thought of in the context of their ubiquitous connectivity. Net-centric computing isn't communications or networking per se, although it certainly includes both. With the changes in the computing and networking environment we need a different paradigm for distributed computing. The area of net- centric computing encompasses the embedded systems but is much larger in scope. In the near future, many hardware devices will be interconnected in large and highly dynamic distributed systems, using standard communication protocols on standard physical links [1] [3] [5] [6] [7] [8]. Such types of systems exist only for computers interconnected by TCP/IP networks, or for hardware devices interconnected in small areas by using specific protocols for the physical link, such as Bluetooth, Ethernet or X-10.
In this paper we review Net-Centric computing in the perspective of Hardware requirements, Embedded system design, middleware, Control, IT and provide an insight into the issues and challenges ahead.

1 Introduction

The underlying principle of Net-Centric Computing (NCC) is a distributed environment where applications and data are downloaded from servers and exchanged with peers across a network on as as-needed basis. NCC is an ongoing area of interest to a wide-variety of software engineering researchers and practitioners, in part because it is an enabling technology for modern distributed systems (e.g., Web applications) [4]. As such, knowledge of NCC is essential requirement in architecting, constructing, and evolving the complex software systems that power today's enterprizes.

The widespread interest in Ubiquitous and Pervasive Computing systems will give a new impulse to Net-Centric Computing (NCC) systems. Solutions such as OSGi have the capability to become the foundation of a new middleware for NCC systems and offer the possibility to browse the physical world in the same way as the web content is browsed.

The activities for Net-Centric Technology consists of three layers.

– The Information Service Layer pertains to the abstraction of objects. The focus here is on the quality, security, auditability and control.
– The Network Service Layer pertains to all aspects of communications, particularly configuration, fault management, security, performance and accounting.
– The Component Control Layer pertains to the development, acquisition and implementation of components that form the infrastructure for distributed computing.

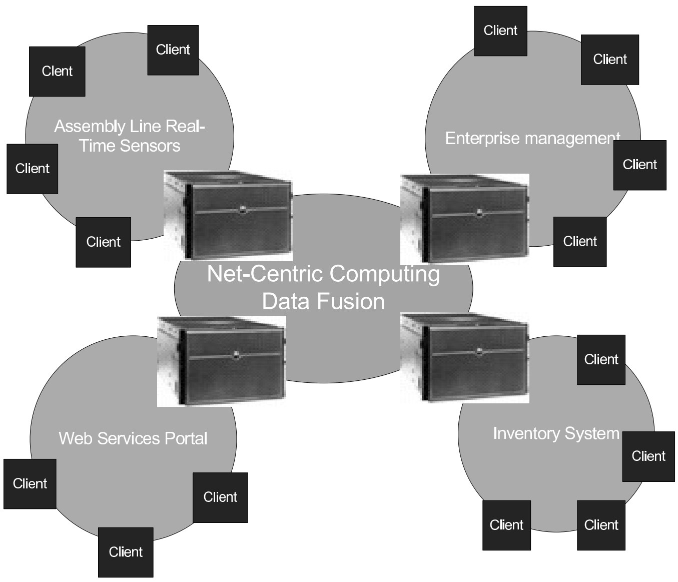

Fig. 1. Net-centric Computing and Data Fusion

Figure 1 shows the environment for Net-centric computing. For several years, business and technology observers have been talking about the major changes being brought by universal networking capabilities, such as the Internet. Today's technology solutions are what we can call "convergence" solutions: They represent the convergence of computing power, communications capability and content the information, data or knowledge that forms the "stuff" of the solution. At the heart of the solution, however, is the networkhence, network-centric, or "netcentric," solutions. Net-centric computing refers to an emerging technology architecture and an evolutionary stage of client/server computing. The common architecture of netcentric computing supports access to information through multiple electronic channels (personal and network computers, cell phones, kiosks, telephones, etc.). This information is made accessible to many more users not just an organization's workforce but also its customers, suppliers and business partners through technologies that employ open, commonly accepted standards (Internet, Java, Web browsers and so forth).

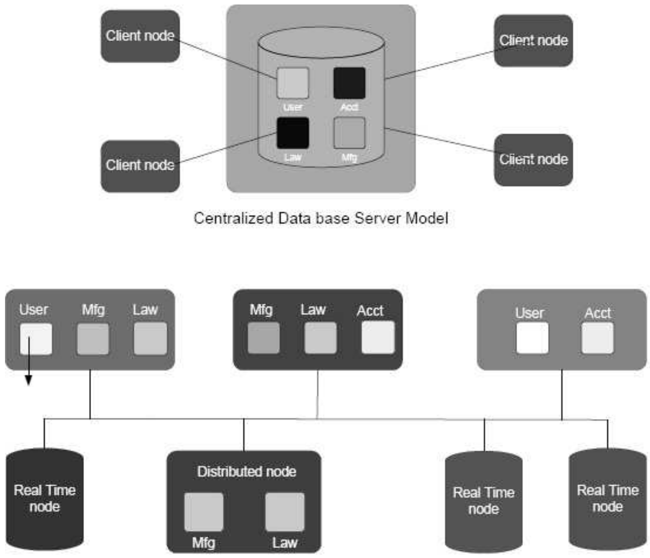

Fig. 2. Net-centric computing vs Distributed systems

Figure 2 shows how Distributed computing is different than Net-Centric computing. Netcentric computing is a common architecture built on open standards that supports many different ways for many different kinds of people to collaborate and to reach many different information sources [2]. The evolutionary yet radical nature of netcentric solutions can be seen in many of those already implemented. The robust architecture of netcentric computing has the ability to evolve as new channels and functionality become available. Netcentric computing links technological capability and strategic opportunity, helping to open today's new markets and provide the flexibility to meet tomorrow's business challenges. It can also add complexity and risk to information systems. Net-centric computing is like a box. Open the box; Inside, there is technology that seems both familiar and new. Something that will do great things for a business. It can enable true e-commerce capabilities. For any business/organization, it can link customers, suppliers, employees and other business partners to its information systems, and thus to its entire business anywhere in the world. It can maximize the flow of information inside any organization, allowing people to share data and knowledge, to collaborate more effectively. This will, in fact, redefine the industry and create new markets.

It is easy to recognize pieces of it: your client/server system, your legacy applications, your network. Yet, taken as a whole, it represents a new kind of technology infrastructure for your organization that links, perhaps for the first time, technology capability with new business opportunity. When you read or hear today of companies opening up new markets, engaging in e-Commerce, delighting their customers with unprecedented levels of services or streamlining their internal processes by encouraging the sharing of information and data, that's netcentric computing in action.

1.1 Flow Computing Model

In this context of net-centric computing we propose a new distributed computing paradigm [8] called flow computing in which nodes comprising the internet can be dynamically contracted to perform the required computing tasks. With the increasing computing capabilities of the router nodes or by specifically deploying additional computing facilities, some of the end hosts (client or server) computation may be delegated to the intermediate nodes (INs). The instructions about how to do the processing may be provided by the end hosts. Thus internet is converted into a large distributed computing environment, we call this paradigm flow Computing because data are processed on the fly enroute from source to destination. Only a few nodes may be equipped to support such Flow computing. Figure 3 shows a flow computing model.

Individual nodes may be designed to provide specific or general computing services. To facilitate such a facility, we propose to develop a reliable transport layer protocol, called Intermediate Processing Protocol (IPP) for processing within the internet. The protocol design makes provisions for connection set up handshake, processing capability reservation, intermediate processing, data acknowledgement, buffering and retransmission, flow and congestion control, ordered delivery and security issues.

Flow computing is different than Peer to Peer computing and grid computing environment models where the computing nodes are known in advance with the help of certain services. In our model, the end nodes need not be aware of location of such services and no centralized or distributed directory services are maintained. Figure 4 shows components of

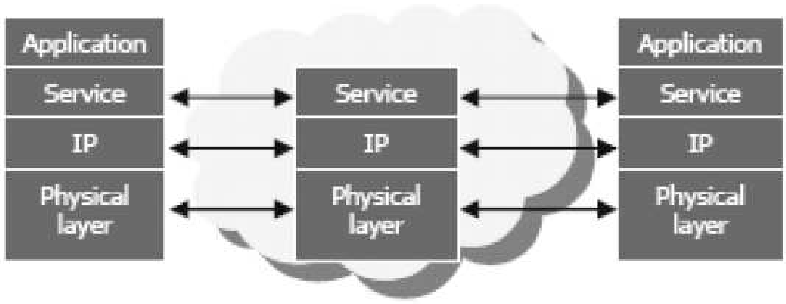

Fig. 3. Flow computing model

a flow computing model. The broader impact of flow computing approach is to allow a large number of different kinds of hosts to be integrated into the internet computing environment and provide ubiquitous services to a large class of small devices and reduce need for large and or centralized servers. Some of the major advantages of flow computing are: full CPU/memory utilization; new business model for ISPs; simplified client and server implementation; support for new multimedia applications for wireless clients; and new distributed computing paradigm by way of merging/processing data as it flows from various sources to clients.

In a flow computing model, application flows are meaningfully processed to satisfy end host requirements, and efficiently utilized for computing and network resources. Due to the support for intermediate processing this model would be a generalization of other distributed object models like grid computing, peer to peer computing and client server computing. Our flow computing model has the advantages of both P2P and grid computing as it enables data sharing and processing by treating the entire network as a reliable computing engine, including the intermediate routers or enhanced routers with little additional overhead.

Flow computing proposes to use a new transport layer protocol that seamlessly integrates with the widely used TCP/IP protocol suite. The idea in this work stems from the observation that there is an increasing number of devices with varying computing powers and energy sources being connected to the internet. End hosts may or may not be able to

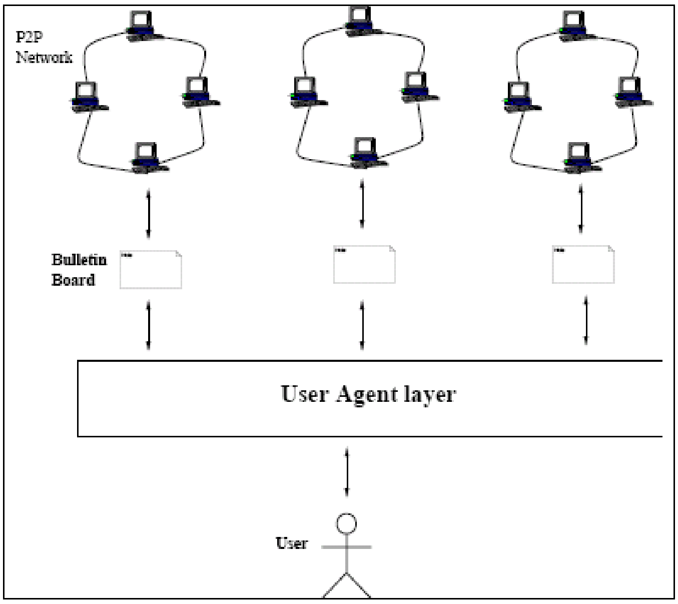

Fig. 4. Components of Flow model

perform all the computations whose results are of interest to them due to unavailability of computation power or need for energy conservation. The data source may not be able to provide such computing services, but may get overloaded due to a large number of such clients. Moreover in certain environment all raw data may be stored on large data repositories such as disk farms that may be optimized towards streaming data from disks to networks to be processed by servers located elsewhere. Related is a scenario where data needed for processing may be distributed such as sensor networks. In such cases it would be more prudent to assign computing task to some intermediate node rather than to a single source/server.

One such environment is Enterprize computing, with a secured networking environment. Consider a scenario an executive of a company would like to view total sales data for a specific product line on his cell phone. The computation needs to obtain raw sales data from the data server and after processing display the result. If the data server does not have the needed processing power and cell phone would not like to manage this data, who should provide the service? Our goal is that either the cell phone host on the forward path or the data server on the backward path during establishment of connection should have contracted this service out to some intermediate node who is a willing participant and has idle capacity available at the time the request is being processed.

2 Aspects of Net-centric Computing

2.1 Architectures

ComP2P CompuP2P given in [7] is an architecture for sharing of computing resources in Internet-scale computational grids. It provides idle resources, such as processing power, memory storage etc., of computing engines that have Internet connectivity and are under-utilized most of the time to user applications that might require them. It achieves this objective by allowing owners of idle resources to sell to those who need them. Thus applications, like scientific simulations and data mining, requiring large processing requirements, can tremendously benefit from potentially unlimited availability of compute power provided by CompuP2P. Likewise, database applications, requiring huge storage, can harness the disk capacity of virtually millions of machines connected to the Internet.

Fig. 5. ComP2P architecture

CompuP2P shown in Figure5 uses light-weight protocols for building and operating dynamic computing resource markets, where sellers and buyers can come together to negotiate transfer (usage) of resources from seller to buyer nodes. To meet the resource sharing requirements, the lookup of such markets and the availability of resources are robust even in the face of several nodes entering or leaving the network at the same time. CompuP2P uses ideas from game theory and microeconomics to devise incentive-based schemes for motivating users to share their idle computing resources with each other. The trading and pricing of resources is done in a completely distributed manner without requiring any trusted centralized authority to oversee the transactions.

Pricing Structure and Incentive Based Allocation Pricing in CompuP2P markets, where buyers (clients) pay to sellers (computing nodes) and intermediaries (MOs) that facilitate the transactions, can be a non-trivial issue. Unlike in real-world, there likely will be no well established protocols (government rules and policies), and institutions and infrastructure (such as stock exchanges) in an Internet-scale computational grid setting that can govern the parameters (such as the price charged, the place of occurrence etc.) of the transactions. Due to such constraints several non-trivial issues need to be addressed - setting resource prices, determining payoffs to intermediate, preventing cheating etc. We again borrow ideas from game theory and microeconomics for developing appropriate pricing strategies, which addresses the above issues to some extent. Utilizing the model that a transaction involving the trading of resources can be modeled as a one-shot game and using the results from game theory (the classical Prisoner's dilemma problem), we observe that long-term collusion among resource sellers (and MO) is unlikely to occur. In one-shot Prisoner's dilemma game, non-cooperation is the only unique Nash equilibrium strategy for the players. In fact, the model of Bertrand oligopoly suggests that sellers (irrespective of their number) would not be able to charge more than their marginal costs (MCs) for selling their resources. In Bertrand oligopoly sellers strategy is to set "prices" (as opposed to "outputs" in Cournot oligopoly) and is thus more reasonable to assume in the context of CompuP2P. This is because in CompuP2P all the sellers in a market sell the same kind (volume) of a resource. As a consequence, sellers (irrespective of how many there are in a market) in CompuP2P set prices equal to their marginal costs only.

2.2 Embedded Systems Design

Embedded designers were among the first to pick up on the significance of the Internet and the World Wide Web and start using them. This is one of the largest markets in the embedded arena is, of course, networking and data communications. The nature of this market is changing from the use of a computer as a data processing or control engine to a data flow engine, moving data from "in here" to "out there" and vice versa. Combined with the higher bandwidths and the large numbers of users, the network processors embedded in the router-and-switch fabric are essentially multiprocessor applications, not loosely coupled, but tightly coupled. This situation is causing some developers to worry about parallel programming issues - a can of worms with no satisfactory methodologies that are widely accepted.

2.3 Middleware Technology in Net-centric Computing

One of the key components of NCC technology is middleware. It is the "glue" that connects disparate components in a heterogeneous environment across a network. Middleware is a well-established research topic in software engineering. In an NCC context, middleware functionality informs the decisions made by all stakeholders, since applications must be engineered within the constraints of the available technology [6].

As NCC applications become more pervasive, the need for new developments in middleware technology becomes apparent. The unique requirements of today's NCC operational environment, such as the need to incorporate security policies across all aspects of the system, expose gaps in current offerings. The identification of such shortcomings in turn provides opportunities for novel developments in the area in the coming years.

2.4 Control Objective for Net-centric Technology

As distributed computing systems grow to hundreds or even thousands of devices and similar or greater magnitude of software components it will become increasingly difficult to manage them without appropriate support tools and frameworks. For example: In distributed applications, single transactions may span multiple hosts and multiple processes. However, applications must still guarantee the atomic integrity of transactions (that is, a unit of work). In distributed environments, both users and their applications can move. Users can access applications from just about anywhere and system administrators enjoy the luxury of moving applications/components among various machines based on such factors as load, hardware failure, performance and others. Applications no longer deal with only simple data types. Current technologies allow system designers to incorporate enhanced objects such as video, audio and multimedia into even the most basic applications [3].

3 Role of Net-Centric Computing in Enterprize Integration Architectures

Enterprize integration has the goal of providing timely and accurate exchange of consistent information between business functions to support both strategic and tactical business goals in a seemingly seamless manner [9]. Although there have been some success, in general there is no clear roadmap for how to achieve effective integration of information systems [5]. Full scale integration efforts tend to focus on integration across an organization's information systems, or in B2B applications between organizations. However, smaller scale efforts can focus on integration at different levels of granularity.

3.1 Net-centric Scenarios

Example 1. NBTel, the telecommunications provider for the Canadian province of New Brunswick, had an existing customer self-service application, installed in the early 1990s, that enabled customers to conduct transactions and make inquiries by phone. As NBTel moved to enhance this application, netcentric computing allowed the organization to evolve this approach to a richer multimedia environment through interactive channels, including the Internet. Today, NBTel's Interactive Phone Store gives customers access, through their personal computers, to fully interactive service over the organization's broadband and Internet network. Interactive television access is also planned. Rockwell Collins has extensive domain knowledge in net-centric computing due to its participation, along with General Dynamics, in developing the Integrated Computer System (ICS) for the U.S. Army's Future Combat System (FCS).

Today's netcentric solutions would not be possible if we were still in the era of proprietary architectures and incompatible technologies, software and standards. The common architecture of the netcentric approach solves a number of information technology issues, such as systems interoperability, and provides the benefits of moving to an open computing environment and common standards, linking everything between the desktop and the Internet. The benefits of this openness can be seen in most netcentric solutions today.

Example 2. Commerzbank, the fourth largest bank in Germany. Commerzbank wanted to push the technology envelope in order to reach a younger, more profitable customer base without incurring the costs of a branch delivery system. Its solution: COMLINE, a "virtual" bank subsidiary designed to attract new, high-income customers between the ages of 25 and 50 with a range of direct-banking services never before offered. The new bank needed to provide a unique combination of flexibility, capability and convenience. This is possible with Net-centric computing.

Outside the enterprise, the reach of netcentric computing allows an organization to link its employees, customers, suppliers, partners and others, irrespective of time, location or device. But reach applies inside the enterprise as well among its departments and business units, at home and around the world. It enables the sharing of information and allows everyone to tap into the organization's brain power.

An important netcentric initiative that extends internal reach has come from Motorola. Motorola designed a self-service network, or Enet, that gives its employees 24-hour access to basic information. This helps them put HR personnel to address higher value added work. , This network is accessible through the organization's intranet, as well as through stand-alone corporate kiosks. It not only provides information on demand but is integrated with the organization's human resources system as well.

Netcentric computing is today's enabler of major business capabilities. Perhaps for the first time in the history of computing, people are defining an evolutionary stage in the industry, not just by pointing to a new set of hardware components, but by telling stories about what their business is now capable of doing, and about the value being derived from netcentric solutions.

Example 3. A US-based utility that manages a pool of retail electricity suppliers for all or parts of six states. The utility today handles almost 8 percent of the electric power in the United States. As a result of deregulation, electricity customers were given a choice of electricity provider. This meant that the industry needed to think in new ways about how to schedule and provide service. The utility's business strategy was to operate the first bid-based regional energy market in the United States. The netcentric technology enabler for that business strategy was the first Internet-based retail contract and scheduling application in the United States electric power industry.

The system enables participants to buy and sell energy, schedule bilateral transactions and reserve transmission service; it also provides accounting and billing services for these transactions. The system allows more than 30 retail electricity suppliers to manage their energy services to residents in their area. The utility has since become one of the most liquid and active energy markets in the United States.

Example 4. Figure 6 gives an example scenario to show how a long distance customer can get connected to a manufacturer/vendor in a Netcentric computing to order custom made cars. Accenture has been working on a systems development project for a large automotive manufacturer. They recently reviewed the technical infrastructure for a centralized system that was to be integrated with local dealer systems and asked the organization executives, What if, at some point in the future one or two years from nowthe way in which the business is transacted changes? What if, for example, somebody walked into a dealership on a Thursday or Friday, interacted with the system (that is, created a customized car, selected the color, looked at the finance or lease package) but then said, "I want to go away and think about this for the weekend." Then suppose the organization had a 24-hour call center. That same customer could call and, by speaking to a customer-service representative about the details of the package designed at the dealership, talk through the deal on a Sunday afternoon. Right there, the customer could make some adjustments and close the transaction over the phone. Of course, being a 24-hour operation, the call center could be based anywhere in the world.

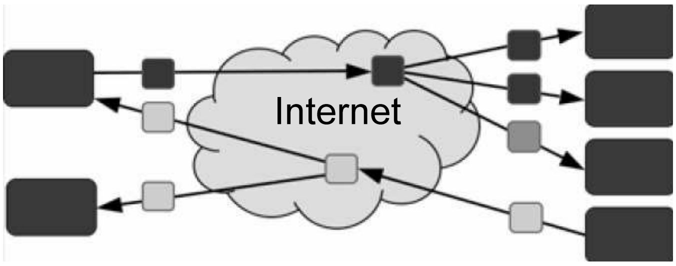

Fig. 6. Scenario for Automotive Manufacturers

This would be a new way to buy cars not dramatically different, perhaps, but it would have a big impact on the technology infrastructure of the business. All of a sudden, the entire context of the transaction from the showroom conversation on Thursday or Friday would have to be made available to a call center three days later, potentially on the other side of the world.

We asked our client if this scenario was within the realm of possibility. It was. Would there be a significant impact on the infrastructure if we did not design and build in that potential capability today? Yes. Netcentric architectures provide the flexibility and resiliency required in such scenarios.

Features of Netcentric computing:

- It creates a more resilient architecture that evolves and extends over time.
- It enables companies to transition their legacy systems into a new environment.

Flexibility also becomes the key to how an organization plans for developing future netcentric capabilities. A critical point for companies today is that even if their current business solutions do not involve netcentric computing, it is vital to maintain the flexibility in technical architecture that will permit a move to netcentric computing in the future.

4 Risks and Challenges

Hardware Complexity

Netcentric computing presents new complexities and risks involved in building netcentric infrastructures. If we view it in the context of mainframes where we may have had three or four major infrastructure components: the database, the CPU, a network and some terminals. Two or three suppliers provided all those major components, so choices were limited. Also that there were only six to eight combinations of all components and vendors, the complexity was manageable. During the client/server computing, infrastructure components increased from five six to some more : workstation, workgroup server, enterprise server, a database and a couple of networks. Each had five to seven possible suppliers. So instead of six to eight combinations, we were looking at up to 40. The size of the problem was bigger, and the nature of the computing solution got more complex.

Component	Net-Centric Computing	Traditional Client-Server computing
Thin Client Devices	10	-
Middleware(metal Frame)	1(with 10 user license)(1 server)	-
Number of Processors	A dual processor(server)	10(client)1(server)
Hard disks	A @12 GB server	@2GB(10 clients)+1 Server
Memory	1(144Mb RAM)(server)	@32MbRAM(10 clients)+1 server
Disk Drives	1(server)	10 Client+1 server
Monitor, keyboard, mouse	10 clients + 1server	10 Client+1 server

Fig. 7. Hardware requirements in Net-Centric Computing

With netcentric or flow computing, if we add only 20 percent more components and, let us say, 30 percent more providers, we may have between 40 combinations to up to 100. It is more exciting to see that the components and providers change practically every day. The vendors are numerous, and the products are younger the "average version number" is much lower.

This complexity can be managed through the use of a proven architectural approach. Experience from client/server development clearly demonstrated that technology architectures can isolate and manage risk. They permit the development of consistent, reliable and high-quality applications that are also better integrated within and between an organization's business units. The netcentric architecture framework again, an evolution from client/server architectures helps neutralize technology complexity by managing the explosion of new technologies. The framework has been successfully used by hundreds of companies to deliver new business capabilities across a wide range of industries and environments.

In a net centric environment the challenges include performance, security, reliability, and usability. The vendor claims that it is faster, cheeper and easier. Our framework accomplishes two major things. First, it helps manage the development of the netcentric solution, serving as a guide and a completeness check, allowing one to assess more easily the types of services and products needed for any specific situation. Second, the framework is a logical representation of the environment in which the business problem will be addressed by the computing solution. It is a tool that allows development teams to break the problem down into component parts and reassemble a solution.

5 Conclusions

From a technical point of view, netcentric computing is breaking down the hierarchy of command and control that was implicit in earlier technologies. It is moving intelligence to the edge of the network and flattening the hierarchy, while it creates new strategic opportunities. Today's opportunity, however, is tomorrow's necessity. Innovators in netcentric computing today can redefine their markets and seize the advantage. Tomorrow's imitators will be forced toward these new technologies simply to stay in business.

In all, the capabilities of net-centric computing provide greater flexibility, enabling companies to rapidly adopt new technology to support innovative marketplace solutions.

References

1. Bernard Conrad Cole,The Emergence of Net-Centric Computing: Network Computers, Internet Appliances, and Connected PCs,Prentice Hall (January 1999)(Hard-

cover).
2. Hugh W. Ryan et.al, Netcentric Computing: Computing, Communications, and Knowledge, Auerbach Publications (1997).
3. ISACA(Society that leads in IT security, assurance and control); Control objective for Net-Centric Technology, IEEE Press.
4. STEP 2005 workshop on Net-Centric Computing (ncc2005).
5. Dennis Smith, and Scott Tilley; On the role of Net-Centric Computing in enterprise Integration Architectures, ASERC Software Architecture workshop 2001.
6. Thierry Bodhuin, Italy Gerardo Canfora, Italy Rosa Preziosi,an Italy Maria Tortorella, "Open Challenges in Ubiquitous and Net-Centric Computing Middleware," 13th international workshop on engineering and practice (STEP O5).
7. R. Gupta, and A. K. Somani, "CompuP2P: An Architecture for Sharing of Computing Resources In Peer-to-Peer Networks With Selfish Nodes," In Online Proceedings of Second Workshop on the Economics of Peer-to-Peer Systems, Harvard University, June 2004.
8. Arun K. Somani, "Proposal on Internet as Distributed Computing Environment," personal.
9. Dennis Smith, and Scott Tilley, "On the role of net centric computing in enterprise integration architecture"', ASERC Softaware Workshop 2001.