EXHIBIT D

KEKER & VAN NEST, LLP
ROBERT A. VAN NEST - #84065
BRIAN L. FERRALL - #160847
LEO L. LAM - #181861
RYAN M. KENT - #220441
710 Sansome Street
San Francisco, CA 94111-1704
Telephone: (415) 391-5400
Facsimile: (415) 397-7188

Attorneys for Defendant
BLUEARC CORPORATION

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

NETWORK APPLIANCE, INC., Plaintiff, v. BLUEARC CORPORATION, Defendant.	Case No. C 03-05665 MHP **EXPERT REPORT OF MARSHALL KIRK MCKUSICK, Ph.D.**

EXPERT REPORT OF MARSHALL KIRK MCKUSICK

I. Introduction and Overview

I have been retained by Keker & Van Nest, litigation counsel for Defendant BlueArc Corporation in this action, as a testifying expert witness. I submit this expert report as a statement of the testimony and opinions regarding validity that I may offer at trial, and the bases for my opinions. I reserve the right to supplement this expert report in light of any additional relevant information that may be discovered during the course of this continuing litigation.

II. Qualifications and Compensation

Attached as Exhibit A to my expert report is a copy of my curriculum vitae, presenting my background and qualifications, including all publications for the last 10 years. Within the last four years, I have given expert testimony at deposition in only one case, *Veritas v. Integratus* filed in United States District Court, Northern District of California, San Jose.

I am compensated at the rate of $375 per hour for my time spent on this case. No part of my compensation depends upon the outcome of this matter.

III. Information Considered in Forming Opinions

In connection with forming the opinions stated in my expert report, I have reviewed and considered the following information:

1. United States Patents 5,802,366 ('366 patent), 5,931,918 ('918 patent) and 6,065,037 ('037 patents) and their file histories.

2. Documents produced in litigation discovery labeled: BARC 426-514, 3391-3407, 3408-3409, 3410-3419, 3618-3646, 4141-4153, 4166-4178, 4179-4191, 4256-4267, 4268-4296, 4602-4609, 5069, 5127-5131, 5302-5320, 5512-5523, 5615-5636, 56831-56834, 56862, 56923-56935, 57256-57264, 57389, 57390-57425, 57426-57439, 57448-57552, 112-187, 523-543, 602-619, 693-706, 733-744, 3684-3739, 3742-3805, 3855-3864, 3884-3934, 4299-4307, 4388-4396, 4514-4533, 5070, 56837, 57107-57129, 5713-5716, 57145, 5717-5718, 57193-57215, 5719-5722, 57231-57255, 57285-57364, 57835-57845, 57879-57999, 58491-58510, 58511-58522.

3. References cited in the '037 prosecution history: U.S. Patents No. 4,626,634; 4,649,473; 4,709,325; 4,903,258; 4,993,017; 5,133,053; 5,218,697; 5,218,697c; 5,557,798; 5,506,988; 5,355,453; and papers by Barrera, Draves, Kupfer, Osterhout, Rogado, Roy and Welch.

4. Documents labeled BERG 1-249.

5. Network Appliance, Inc.'s Updated Disclosure of Asserted Claims and Infringement Contentions Pursuant to the Court's Means-Plus-Function Claim Construction Order of

1/7/05 (P.R. 3-1), (dated January 24, 2005).

6. Claim Construction Memorandum and Order for United States Patent Nos. 5,802,366, 5,931,918 and 6,065,037, (dated November 30, 2004).

7. Claim Construction Memorandum and Order for United States patent Nos. 5,802,366, 5,931,918 and 6,065,037: Construction of Means-Plus-Function Claim Elements, (dated January 5, 2005).

8. BlueArc's Preliminary Invalidity Contentions Under Patent L.R. 3-3

9. BlueArc's Final Invalidity Contentions (Including Means-Plus-Function Claims).

10. BlueArc's Supplemental Invalidity Contentions.

11. "Evolutionary Path to Network Storage Management" by Israel

12. Each of the following references cited in BlueArc's invalidity contentions:

Author	Doc Title
Almes Zwaenepoel	Understanding and Exploiting Distribution
Berger	US 4590556 Co-processor combination
Berglund	Introduction to the V-System IEEE Micro
Bowman	Multitasking keys SCSI control of large SMD-compatible disks
Cheriton Malcolm Melen Sager	Thoth, a Portable Real-Time Operating System
Cheriton	Design of a Distributed Kernel
Cheriton Williamson	Network Measurement of the VMTP Request-Response Protocol in the V Distributed System Stanford University Department of Computer Science report
Cheriton	V Kernel: A Software Base for Distributed Systems IEEE Software
Cheriton Dewing	Host Groups: A Multicast Extension for Datagram Internetworks Stanford University Department of Computer Science Report
Cheriton	V Distributed Operating System: Principles and Principal Experiences
Cheriton	Experiment Using Registers for Fast Message-Based Interprocess Communication
Cheriton	Local Networking and Internetworking in the V-System
Cheriton	V Distributed System Communications of the ACM
Cheriton Zwaenepoel	Distributed V Kernel and its Performance for Diskless Workstations ACM and Stanford University Department of Computer Science Report

Author	Doc Title
Cheriton Roy	Performance of the V Storage Server: A Preliminary Report Proceedings of the 1985 ACM Computer Science Conference
Cheriton Zwaenepoel	Distributed Process Groups in the V Kernel ACM Transactions of Computer Systems
Cheriton Mann	Decentralized Naming Facility Stanford University department of Computer Science Report
Cheriton	UIO: A Uniform I/O System Interface for Distributed Systems Stanford University Department of Computer Science
Cheriton Zwaenepoel	One-to-Many Interprocess Communication in the V-System Stanford University Department of Computer Science Report
Cheriton	Unified Management of Memory in the V Distributed System Stanford University Department of Computer Science
Cheriton	Exploiting Recursion to Simplify RPC Communication Architectures Stanford University Department of Computer Science Report
Cheriton Mann	Uniform Access to Distributed Name Interpretation in the V-System
Computer Design	Ethernet compatible frontend processor
Computer Design	Network controllers grow smarter/smaller
Coulouris Dollimore	Distributed Systems
DiDio	Sun eases networking of diskless workstations
Ennis	Netbios: IBM's PC interface for distributed applications
Excelan	NX 200 Network Executive Reference Manual Revision A
Howard Kazar Menees Nichols Satyanarayanan Sidebotham West	Scale and Performance in a Distributed File System ACM
Israel Jett Pownell Ericson	Eliminating Data Copies in UNIX-based NFS Servers
Kanakia Cheriton	VMP Network Adapter Board (NAB): High-Performance Network Communication for Multiprocessors
Kent	TCP/IP Print Server: Server Archtecture and Implementation
Kleiman	Vnodes: An Archtecture for Multiple File System Types in Sun UNIX USENIX 1986 Summer Conference Proceedings
Levy Silberschatz	Distributed File Systems: Concepts and Examples

Author	Doc Title
Marzullo Schmuck	Supplying High Availability with a Standard Network File System IEEE
McLeod	Sacrifices to Ra or Learning to Administer a Sun Network EUUG
Millard	US 4096567 Information Storage Facility With Multiple Level Processors
Moran Sandberg Coleman	US 5359713 Method and Apparatus for Enchancing Synchronus I/O in a Computer System with a Non-volatile Memory and Using an Acceleration Device Driver in a Computer Operating System
Mullender	Amoeba distributed operating system: Selected papers 1984-1987
Panjawani	NFS on ISDN
Pawlowski Tumminaro Hixon Stein	UniForum Conference Proceedings Network Computing in the UNIX and IBM Mainframe Environment
Powers	UniForum Conference A Front-End TELNET/Rlogin Server Implementation
Ramakrishnan Emer	Model of File Server Performance for a Heterogenous System
Rosen Wilde	NFS Portability
Sandberg	Sun Network Filesystem: Design, Implementation and Experience
Schroder	PEACE: The distributed SUPRENUM operating system Parallel Computing 7
Siegel	US 5073852 Network Protocol Translator
Stokes	US 4101960
Sunshine Ennis	Broad-Band Personal Computer LAN's
Tanenbaum Van Renesse	Distributed Operating Systems

Author	Doc Title
Cheriton Lanz (principal investigators) contributing authors include: Berc Berglund Bothner Brooks Cheriton Deering Dunwoody Edighoffer Finlayson Gray Hitson Kaelbling Lantz Mann Maslen Nagler Nowicki Pallas Roy Schuster Stumm Theimer Zuleeg Zwaenepoel	V-System 6.0 Reference Manual
Way	Front-end processors smooth local network-computer integration Electronics
Williamson Cheriton	Overview of the VMTP Transport Protocol IEEE
Zwaenepoel	Message Passing on a Local Network
Fong Polster	Build a High-Performance SCSI Bridge Controller [Not included in Exh A to the Inv. Contentions, but mentioned in later exhibits]
Leiberman	Hierarchical File Server Puts Archives On-line
Haskin	How to Satisfy a Voracious NFS Appetite

IV. Summary of Opinions

As explained in detail below, it is my opinion that, based on Network Appliance's infringement contentions, the asserted claims of the '366, '918, and '037 patents-in-suit are anticipated and/or rendered obvious by the prior art. It is also my opinion that the subject matter of certain asserted claims lack adequate support based on the disclosure provided in their corresponding patent specification.

I may testify further as to subject matters within my area of expertise that will be useful to inform the finder of fact as to the bases for my opinions. I may also testify concerning additional subject matters properly raised at trial within my area of expertise, and regarding other subject matters responsive or in rebuttal to contentions advanced before or during trial.

V. Statement of Expected Testimony, Opinions, and Bases Therefor

In my view, one of ordinary skill in the art to which the patents-in-suit pertain would have had at least a bachelor's degree in computer science or computer engineering as well as five years' experience, prior to the application date of the patent, in network communications, computer architecture, and computer software engineering as well as knowledge of computer hardware engineering.

I am advised that a patent claim is "anticipated" if all the limitations of the claim are disclosed in prior-art patent or other printed publication.

I am also advised that even if all the claim limitations are not disclosed in a single prior–art reference, the patent claim may be rendered "obvious" if the difference(s) between the claimed invention and the prior art would have been obvious to one of ordinary skill in the art at the time the invention was made. I understand that a combination of prior art can render a patent claim obvious if the combination would meet all the limitations of the claim when there is motivation to combine such prior art.

A. The Asserted '366 Patent Claims Are Anticipated And/Or Rendered Obvious by the Prior Art

Anticipation

1. Coulouris Textbook

The Coulouris textbook "serve[s] as a text for an advanced undergraduate or graduate level course on Distributed Systems[.]" Thus the material covered in the book would be expected to be known to those that would be considered of ordinary skill in the art. The Coulouris textbook discloses various distributed-processing systems, including the V-system.

The V-system is a multiple facility operating system architecture designed to place as many of the usual operating system functions outside the V-kernel as efficiency permits, including file service. The Coulouris textbook also discloses various network file service software to be run on such distributed systems, including NFS. In light of Network Appliance's infringement contentions, the Coulouris reference discloses all limitations of—and thus anticipates—the '366 claims-in-suit, as shown in the claim charts provided below.

	'366 Claim 1	**Coulouris Reference**
1.1	Apparatus for use with a data network and a mass storage device,	Coulouris discloses an apparatus made up of a federation of machines each doing a specific task for the rest of the machines. "Loosely-coupled distributed systems are particularly effective in exploiting the power and flexibility of single-user computers or workstations, enabling them to access shared data and resources located in other server computers via high-speed local networks (Figure 1.1)." p. 2. One of the services shown in Figure 1.1 is a file server. That machine is connected to a mass storage device to hold the files that it served.
1.2	comprising the combination of first and second processing units,	Each of the machines disclosed by Coulouris have a processing unit in them: "The key features of current workstations include ... a 32-bit processor with at least 2Mb of RAM, ... an interface to a local network, the dominant network technology being the Ethernet." pp. 3-4. Thus each machine is connected to a common network and is able to communicate with the other machines in the network.
1.3	said first processing unit being coupled to said network and	Each of the workstations shown in Figure 1.1 on page 3 can constitute the first processing unit.
1.4	performing procedures for satisfying requests from said network which are within a predefined non-NFS class of requests, and	On page 4, Coulouris states: "In distributed systems that include file servers, *diskless* workstations can be used." And further "The UNIX operating system is still widely used in workstations[.]" Such a diskless workstation would be able to perform procedures for satisfying requests from the network which are within a predefined non-NFS class of requests.
1.5	said second processing unit being coupled to said network and to said mass storage device and	The file server shown in Figure 1.1 on page 3 can constitute the second processing unit. Like all the machines, it is connected to the network. The file server has a mass storage device attached.
1.6	decoding NFS requests from said network, performing procedures for satisfying said NFS requests, and encoding	The file server in Figure 1.1 on page 3 receives requests from the network, performs procedures for satisfying the NFS requests and encodes an NFS reply message that is returned on the network. Coulouris discloses the details

	NFS reply messages for return transmission on said network,	of an implementation of an NFS server in section 10.3 on pages 264 through 268. In particular, "The NFS service is implemented in terms of remote procedure calls between kernels. …If the file is in a remote directory the local kernel makes an [NFS] service call to the appropriate remote kernel." The method of performing a procedure in response to a request and sending a reply is disclosed on page 84 and shown in Figure 4.3. The figure shows a remote procedure call as the receipt of a request from the network that is handled as local function (in this case as a local filesystem access). The file server then encodes an NFS reply message with the result which it returns on the network.
1.7	said second processing unit not satisfying any requests from said network which are within said predefined non-NFS class of requests.	On page 27 Coulouris states: "These [servers] provide simple sharing of hardware devices such as printers, access to shared data through file servers, and several other services that are designed to replace the functions of a centralized operating system." The role of the file server is to service NFS requests and not the non-NFS class of requests.

	'366 Claim 3	**Coulouris Reference**
3	Apparatus according to claim 1, wherein said first processing unit includes a general purpose operating system and wherein said second processing unit does not include a general purpose operating system.	Coulouris discloses the construction of a system built from a lightweight kernel on pages 39 and 40: "A simpler system kernel providing lightweight processes and interprocess communication is a better basis for building workstation software. Server software may also be built on such a basis." Thus while a client workstation runs a general purpose operating system, the file server runs just the lightweight kernel and the additional software needed to handle NFS requests.

	'366 Claim 5	**Coulouris Reference**
5	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	See analysis of prior art for limitations 1.5, 1.6.
5.1	means for decoding NFS requests from said network;	Notwithstanding the Court's definition of the "means for decoding" as "microprocessor 210 programmed to convert data from the NFS message format to LNFS message format," NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. Coulouris discloses a "transport software module" that

		decomposes the request into a form that can be communicated via another, lower level protocol, such as the OSI transport layer disclosed on pages 59 through 65: "The transport layer's task is to provide a network-independent message transport service between pairs of network **ports** (also known as **sockets**). Ports are software-definable destination points for communication within a host computer." Coulouris also discloses communications at the LNFS or presentation layer: "The presentation layer software in each computer transforms data from the formats used in the computer it is running on into a set of standard network representations called **external data representation (EDR)** before transmission. Received data is transformed from the network representations into the local data formats." Therefore, Coulouris fully discloses handling NFS requests at either the transport or presentation layer.
5.2	means for performing procedures for satisfying said NFS requests, including accessing said mass storage device if required; and	The Court has defined the means for "performing procedures for processing NFS [or file system] requests" to include the "network controller 110, file controller 112, storage processor 114, system memory 116, and VME bus 120" programmed to "perform such procedures based on the algorithms disclosed" in sections 17:20-18:7, 19:24-21:12, 26:26-28:67 of the specification. Again, notwithstanding the Court's definition, NetApp contends that the network interface card in BlueArc's products meet this limitation because it "performs NFS request dispatching and IP routing by programming the microprocessor . . . to dispatch and route information using standard dispatching and routing procedures." Coulouris discloses a file server that would perform such standard dispatching and routing procedures of an NFS server in section 10.3 on pages 264 through 268: "The NFS service is implemented in terms of remote procedure calls between kernels. ...If the file is in a remote directory the local kernel makes an [NFS] service call to the appropriate remote kernel." Coulouris further discloses the method of performing a procedure in response to a request on page 84 and in Figure 4.3. Figure 4.3 shows a remote procedure call as the receipt of a request from the network that is handled as local function (in this case as a local filesystem access), and the return of the result to the network. Coulouris also discloses the mechanism for accessing mass storage to satisfy NFS file requests in chapter 6 on pages 135 to 153. Specifically, on page 146, Coulouris states: "The block service supports block read and write operations, implementing them as more primitive disk operations[.]"

		Coulouris fully discloses this limitation based on NetApp's contention.
5.3	means for encoding any NFS reply messages for return transmission on said network,	Notwithstanding the Court's definition of the "means for encoding" as "microprocessor 210 programmed to convert data from the LNFS message format to the NFS message format." NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. Coulouris discloses a "transport software module" that decomposes the request into a form that can be communicated via another, lower level protocol, such as the OSI transport layer disclosed on pages 59 through 65, stating: "The transport layer's task is to provide a network-independent message transport service between pairs of network **ports** (also known as **sockets**). Ports are software-definable destination points for communication within a host computer.'' Coulouris also discloses communications at the LNFS or presentation layer, stating: ``The presentation layer software in each computer transforms data from the formats used in the computer it is running on into a set of standard network representations called **external data representation (EDR)** before transmission. Received data is transformed from the network representations into the local data formats." Therefore, Coulouris fully discloses handling NFS requests at either the transport or presentation layer.
5.4	said first unit lacking means in said first unit for executing any programs which make calls to any general purpose operating system.	As discussed for claim 3, Coulouris discloses a file server that runs just the lightweight kernel and the additional software needed to handle NFS requests. Such a file server would not run a general purpose operating system.

	'366 Claim 6	**Coulouris Reference**
6.1	A network file server according to claim 5, further including a second unit comprising means for executing programs which make calls to a general purpose operating system.	As shown in Figure 1.1 on page 3, Coulouris discloses a second unit – a client workstation – that runs a general purpose operating system and that can access a file server.

	'366 Claim 7	**Coulouris Reference**

7.1	A network file server according to claim 6, wherein said first unit lacks means in said first unit for executing any programs which make calls to a UNIX operating system, and wherein said second unit comprises means for executing programs which make calls to a UNIX operating system.	As discussed for limitation 6.1, Coulouris discloses a second unit – a client workstation – that can run a general purpose operating system and that can access a file server. Coulouris also discloses an example on page 16 wherein the general-purpose operating system is UNIX: "An example of NFS in use – Figure 1.6 shows the distributed system environment in the Computer Science Department at Queen Mary College London[.] BSD 4.2 UNIX and Sun NFS run in almost all of the computers and the role of the file server is played by the Sequent, the Vax and some of the Suns." As discussed for claim 3 and limitation 5.4, while a client workstation runs a general purpose operating system, the first unit – the file server – runs just the lightweight kernel and the additional software needed to handle NFS requests.

	'366 Claim 8	**Coulouris Reference**
8	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	See analysis of prior art for the preamble of claim 5.
8.1	means for decoding NFS requests from said network;	See analysis of prior art for limitation 5.1.
8.2	means for performing procedures for satisfying said NFS requests, including accessing said mass storage device if required; and	See analysis of prior art for limitation 5.2.
8.3	means for encoding any NFS reply messages for return transmission on said network,	See analysis of prior art for limitation 5.3.
8.4	said first unit lacking means to execute any user-provided application programs on said first unit.	See analysis of prior art for limitation 5.4.

	'366 Claim 9	Coulouris Reference
9	A network file server according to claim 8, further including a second unit running a user-provided application program.	See analysis of prior art for claim 6.

	'366 Claim 18	Coulouris Reference
18.1	A network file server for use with a network and at least one mass storage device, said network file server including:	Coulouris discloses a file server in Figure 1.1 on page 3. Like all the machines, it has a network interface connected to the network. The file server has at least one and possibly more mass storage devices attached.
18.2	a network interface, coupleable to said network, for receiving NFS requests from said network;	The network interface of the file server is able to receive NFS requests from the network.
18.3	a file server processor coupled to the network interface and coupleable to said at least one mass storage device for executing essentially only NFS requests from said network interface including accessing said at least one mass storage device if required.	The file server receives NFS requests from its network interface and performs the procedure disclosed above in limitation 1.6 in response to such requests. As discussed above for limitation 1.7, the role of the file server is to service NFS requests, not the non-NFS class of requests. When necessary, the file server will access a mass storage device to satisfy NFS file requests. Coulouris discloses this procedure in chapter 6 on pages 135 to 153, specifically stating on page 146: "The block service supports block read and write operations, implementing them as more primitive disk operations[.]"

	'366 Claim 19	Coulouris Reference
19.1	A network file server for use with an Ethernet network and at least one mass storage device, said network file server including:	Coulouris discloses a file server in Figure 1.1 on page 3. Like all the machines, it has a network interface connected to the network: "an interface to a local network, the dominant network technology being the Ethernet." As shown in Figure 1.1, the file server has at least one and possibly more mass storage devices attached.
19.2	a network interface, coupleable to said Ethernet network, for receiving from said Ethernet network packets containing NFS requests to read data from or write data to said at least	As discussed for limitation 19.1, Coulouris discloses a network interface connected to an Ethernet network. As Coulouris discloses on page 59, the network

	one mass storage device;	interface – the transport software module – receives the packets containing the NFS request. As shown in Figure 10.3, the NFS requests can access multiple mass storage devices to either read or write data. The details of handling NFS requests are further disclosed in section 10.3 on pages 264 through 268.
19.3	a parallel bus;	Coulouris discloses using a Sun NFS file servers on page 16: "An example of NFS in use – Figure 1.6 shows the distributed system environment in the Computer Science Department at Queen Mary College London[.] BSD 4.2 UNIX and Sun NFS run in almost all of the computers and the role of the file server is played by the Sequent, the Vax and some of the Suns." The Sun hardware's backplane is a VME bus, which is a high-speed parallel bus.
19.4	a dedicated file server processor, coupled to said network interface by means of said parallel bus, and coupleable to said at least one mass storage device, for executing essentially only NFS requests from said network interface, including accessing said at least one mass storage device if required.	As discussed for claim 3, the file server runs just the lightweight kernel and the additional software needed to handle NFS requests, and thus has a processor dedicated to running only the NFS file service. That file server can run on a Sun hardware that uses a high-speed parallel bus to connect to the network interface, as discussed for limitation 19.3. As shown in Figure 1.1 and 10.3, the file server can be attached to at least one and possibly more mass storage devices.

2. Epoch Reference

The Epoch reference titled "How to Satisfy a Voracious NFS Appetite" discloses a multiple-processor architecture that breaks down the functional units in the same way as the '366 patent. The novelty of the Epoch-1 is made clear in the section of the reference describing its performance: "Because this is the first NFS server that DR Labs has tested, we did not have any other servers to compare it with." In light of Network Appliance's infringement contentions, the Epoch reference discloses each and every limitation of—and thus anticipates—the '366 claims-in-suit, as shown in the claim charts provided below.

	'366 Claim 1	**Epoch Reference**
1.1	Apparatus for use with a data network and a mass storage device,	According to the Epoch reference: "The Epoch-1 Infinite Storage Server from Epoch Systems can truly provide … a virtually inexhaustible supply of data storage for Unix users[.]" Even minimally configured, the Epoch-1 is disclosed as having

		"three or more magnetic disk drives" and "two Ethernet interfaces."
1.2	comprising the combination of first and second processing units,	"The Epoch-1 sports two processors, both Motorola 68020 CPU's running at 25Mhz."
1.3	said first processing unit being coupled to said network and	The first processor "supports the full complement of TCP/IP networking utilities[.]"
1.4	performing procedures for satisfying requests from said network which are within a predefined non-NFS class of requests, and	The first processor, called the Applications Processor (AP), is responsible for performing "Unix housekeeping chores." These housekeeping chores are within a predefined non-NFS class of requests.
1.5	said second processing unit being coupled to said network and to said mass storage device and	The second processing unit, the Front-End Processor (FEP), "handles all I/O operations for the SCSI controllers, [and] two Ethernet interfaces[.]" Thus, the FEP is connected to the network through its two Ethernet interfaces and to the mass storage devices through its SCSI controllers.
1.6	decoding NFS requests from said network, performing procedures for satisfying said NFS requests, and encoding NFS reply messages for return transmission on said network,	The FEP "handles all I/O operations for the SCSI controllers, [and] two Ethernet interfaces[.]" Thus, the FEP is connected to the network through its two Ethernet interfaces and to the mass storage devices through its SCSI controllers. Therefore, the FEP receives and decodes the incoming Ethernet packets containing NFS requests. The FEP then performs procedures to satisfy NFS requests by, for example, transferring decoded NFS requests to the Applications Processor, as well as by transferring and processing requests to access the mass storage devices. It then encodes the NFS reply message into the appropriate packets for return transmission.
1.7	said second processing unit not satisfying any requests from said network which are within said predefined non-NFS class of requests.	The FEP runs "Ready Systems' VRTX real-time multitasking kernel", not a general purpose operating system like 4.3BSD Unix. Therefore, the FEP does not handle non-NFS class of requests such as Unix commands.

	'366 Claim 3	**Epoch Reference**
3	Apparatus according to claim 1, wherein said first processing unit includes a general purpose operating system and wherein said second processing unit does not include a general purpose	"The first processor, called the Applications Processor (AP), runs a somewhat modified version of 4.3BSD Unix[.] … The other processor – known as the Front-End Processor (FEP) – runs Ready Systems' VRTX real-time multitasking kernel[.]'" The AP is running Unix; the FEP runs

	operating system.	VRTX, not a general purpose operating system.

	'366 Claim 5	Epoch Reference
5	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	The Epoch-1 reference discloses an NFS file server with a data network and several mass storage devices as discussed for limitation 1.1
5.1	means for decoding NFS requests from said network;	Notwithstanding the Court's definition of the "means for decoding" as "microprocessor 210 programmed to convert data from the NFS message format to LNFS message format," NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. The Front-End Processor in Epoch is a Motorola 68020 CPU's running at 25Mhz. The FEP "handles all I/O operations for the SCSI controllers, [and] two Ethernet interfaces[.]" Thus, the FEP is connected to the network through its two Ethernet interfaces and to the mass storage devices through its SCSI controllers. Therefore, the FEP decodes the incoming Ethernet packets containing NFS requests.
5.2	means for performing procedures for satisfying said NFS requests, including accessing said mass storage device if required; and	The Court has defined the means for "performing procedures for processing NFS [or file system] requests" to include the "network controller 110, file controller 112, storage processor 114, system memory 116, and VME bus 120" programmed to "perform such procedures based on the algorithms disclosed" in sections 17:20-18:7, 19:24-21:12, 26:26-28:67 of the specification. Again, notwithstanding the Court's definition, NetApp contends that the network interface card in BlueArc's products meet this limitation because the it "performs NFS request dispatching and IP routing by programming the microprocessor . . . to dispatch and route information using standard dispatching and routing procedures." The Front-End Processor in Epoch is a Motorola 68020 CPU's running at 25Mhz. The FEP "handles all I/O operations for the SCSI controllers, [and] two Ethernet interfaces[.]" Thus, the FEP is connected to the network through its two Ethernet interfaces and to the mass storage devices through its SCSI controllers. Therefore, the FEP receives and decodes the incoming Ethernet packets containing NFS requests. The

		FEP then performs procedures to satisfy NFS requests by, for example, transferring decoded NFS requests to the Applications Processor, as well as by transferring and processing requests to access the mass storage devices. It then encodes the NFS reply message into the appropriate packets for return transmission.
5.3	means for encoding any NFS reply messages for return transmission on said network,	Notwithstanding the Court's definition of the "means for encoding" as "microprocessor 210 programmed to convert data from the LNFS message format to the NFS message format." NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. The Front-End Processor in Epoch is a Motorola 68020 CPU's running at 25Mhz. The FEP "handles all I/O operations for the SCSI controllers, [and] two Ethernet interfaces[.]" Thus, the FEP is connected to the network through its two Ethernet interfaces and to the mass storage devices through its SCSI controllers. Therefore, the FEP encodes the reply messages to be sent across the network into Ethernet packets.
5.4	said first unit lacking means in said first unit for executing any programs which make calls to any general purpose operating system.	"The other processor – known as the Front-End Processor (FEP) -- runs Ready Systems' VRTX real-time multitasking kernel and handles all I/O operations for the SCSI controllers, two Ethernet interfaces, modem port and system console port." The processor managing the mass storage and network interfaces is running a real-time operating system that is not able to execute any programs that make calls to a general purpose operating system.

	'366 Claim 6	**Epoch Reference**
6.1	A network file server according to claim 5, further including a second unit comprising means for executing programs which make calls to a general purpose operating system.	"The first processor, called the Applications Processor (AP), runs a somewhat modified version of 4.3BSD Unix[.] ... Epoch users can have use of the *telnet, ftp, rlogin, rsh,* and *rcp* utilities for transferring files between the Epoch and other hosts that support TCP/IP or for logging in to a remote host. In addition, a few new commands and utilities have been added for the management of the Epoch-1's files and disks." The AP processor is able to execute programs that make calls to a general purpose operating system (4.3BSD Unix).

	'366 Claim 7	Epoch Reference
7.1	A network file server according to claim 6, wherein said first unit lacks means in said first unit for executing any programs which make calls to a UNIX operating system, and wherein said second unit comprises means for executing programs which make calls to a UNIX operating system.	"The Front-End Processor (FEP) – runs Ready Systems' VRTX real-time multitasking kernel[.]" VRTX is not Unix and cannot emulate Unix. "The Applications Processor (AP), runs a somewhat modified version of 4.3BSD Unix[.]" 4.3BSD Unix is Unix and can run Unix programs.

	'366 Claim 8	Epoch Reference
8	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	See analysis of prior art for limitation 5.
8.1	means for decoding NFS requests from said network;	See analysis of prior art for limitation 5.1.
8.2	means for performing procedures for satisfying said NFS requests, including accessing said mass storage device if required; and	See analysis of prior art for limitation 5.2.
8.3	means for encoding any NFS reply messages for return transmission on said network,	See analysis of prior art for limitation 5.3.
8.4	said first unit lacking means to execute any user-provided application programs on said first unit.	See analysis of prior art for limitation 5.4.

	'366 Claim 9	Epoch Reference
9	A network file server according to claim 8, further including a second unit running a user-provided application program.	See analysis of prior art for limitation 6.1.

	'366 Claim 18	Epoch Reference
18.1	A network file server for use with a network and at least one mass storage device, said network file server including:	According to the Epoch article: "The Epoch-1 Infinite Storage Server from Epoch Systems can truly provide … a virtually inexhaustible supply of data storage for Unix users[.]" The minimally configured Epoch-1 is disclosed as having "three or more magnetic disk drives'' and "two Ethernet interfaces."
18.2	a network interface, coupleable to said network, for receiving NFS requests from said network;	The network interfaces on the Epoch-1 file server include at least two Ethernet interfaces and are able to receive NFS requests from the network.
18.3	a file server processor coupled to the network interface and coupleable to said at least one mass storage device for executing essentially only NFS requests from said network interface including accessing said at least one mass storage device if required.	"The other processor – known as the Front-End Processor (FEP) – runs Ready Systems' VRTX real-time multitasking kernel and handles all I/O operations for the SCSI controllers, two Ethernet interfaces, modem port and system console port." The FEP manages the mass storage and network interfaces and processes NFS requests from its network interfaces. The FEP will access its connected mass storage devices in response to an NFS request. The FEP does not run a general purpose operating system and does not handle the "Unix housekeeping chores."

	'366 Claim 19	Epoch Reference
19.1	A network file server for use with an Ethernet network and at least one mass storage device, said network file server including:	According to the Epoch article: ``The Epoch-1 Infinite Storage Server from Epoch Systems can truly provide … a virtually inexhaustible supply of data storage for Unix users[.]" The minimally configured Epoch-1 is disclosed as having "three or more magnetic disk drives" and "two Ethernet interfaces."
19.2	a network interface, coupleable to said Ethernet network, for receiving from said Ethernet network packets containing NFS requests to read data from or write data to said at least one mass storage device;	The network interfaces on the Epoch-1 file server include at least two Ethernet interfaces that are able to receive Ethernet packets containing NFS read and write requests from the network.
19.3	a parallel bus;	The Motorola 68020 used by the FEP has a 32-bit parallel bus to connect to its peripherals.
19.4	a dedicated file server processor, coupled to said network interface by means of said parallel bus, and coupleable to said at least one mass storage device, for executing	The FEP "handles all I/O operations for the SCSI controllers, [and] two Ethernet interfaces[.]" Therefore, the network interfaces on the Epoch-1 file server – the two Ethernet interfaces – are connected by the Motorola 68020's parallel bus

	essentially only NFS requests from said network interface, including accessing said at least one mass storage device if required.	to the file server processor – the FEP. Likewise, the FEP is connected to the mass storage device through its SCSI controllers. Therefore, FEP processes NFS requests received from its network interfaces and will access a mass storage device as necessary to satisfy such NFS requests. The FEP does not run a general purpose operating system and does not handle the "Unix housekeeping chores."

3. Sandberg

Sandberg discloses a multiple-processor architecture implementation of a NFS network file service. Specifically, Sandberg discloses using an Excelan Board (see generally Excelan reference manual) and a VAX 750 to implement NFS. Accordingly, as seen in the claim charts provided below, the Sandberg reference discloses each and every limitation of—and thus anticipates—claims 1, 18 and 19 of the '366 claims-in-suit.

	'366 Claim 18	**Sandberg Reference**
18.1	A network file server for use with a network and at least one mass storage device, said network file server including:	Sandberg in Figure 1 discloses the architecture and implementation of an NFS network file service, including a data network and a mass storage device.
18.2	a network interface, coupleable to said network, for receiving NFS requests from said network;	The Excelan board "handles the Ethernet, IP, and UDP layers," as disclosed on page 11. Given that the Excelan board handles the Ethernet, the board receives NFS requests from the network.
18.3	a file server processor coupled to the network interface and coupleable to said at least one mass storage device for executing essentially only NFS requests from said network interface including accessing said at least one mass storage device if required.	The VAX 750 acts as the NFS file server processor, which is defined as the "machine that provides resources to the network" on page 1. Those resources are data stored on a mass storage device. The VAX 750 thus is connected to network and mass storage device. The VAX 750 is capable of executing essentially only NFS requests.

	'366 Claim 19	**Sandberg Reference**
19.1	A network file server for use with an Ethernet network and at least one mass storage device, said network	Sandberg in Figure 1 discloses the architecture and implementation of an NFS network file service, including an Ethernet network and a mass storage

	file server including:	device.
19.2	a network interface, coupleable to said Ethernet network, for receiving from said Ethernet network packets containing NFS requests to read data from or write data to said at least one mass storage device;	The Excelan board "handles the Ethernet, IP, and UDP layers," as disclosed on page 11. Given that the Excelan board handles the Ethernet, the board receives Ethernet packets containing the NFS requests, including requests to read data or write data to a mass storage device.
19.3	a parallel bus;	The Excelan board is connected directly to the VAX 750, which has a 32-bit parallel bus interfacing to its peripherals.
19.4	a dedicated file server processor, coupled to said network interface by means of said parallel bus, and coupleable to said at least one mass storage device, for executing essentially only NFS requests from said network interface, including accessing said at least one mass storage device if required.	The VAX 750 is connected through the parallel bus to the Excelan board, the network interface. As shown in Figure 1 and discussed at page 1, the VAX 750 is connected to a mass storage device via this network, so that the VAX 750 can act as the NFS file server processor and satisfy NFS requests. The VAX 750 is capable of executing essentially only NFS requests.

Obviousness

4. <u>V-System Manual + Pawlowski/Sandberg/Rosen</u>

The V-System Manual discloses the V-System, a multiple facility operating system architecture designed to place as many of the usual operating system functions outside the V-kernel as efficiency permits. The V-System Manual discloses the same system as disclosed in the Coulouris Textbook, except in more detail. The V-System Manual discloses using a network file service running on a separate processor, but does not specify which network file system should run on that separate processor. Accordingly, the V-System does not explicitly disclose the claim limitations that require NFS. It follows that as explained in the Coulouris Textbook claim charts above, the V-System Manual discloses limitations 1.1-1.5, 1.7, 3, 5, 5.4, 6.1, 7.1, 8, 8.4, 9, 18.1, 19.1, and 19.3.

A person of skill in the art would have thought to combine references describing NFS with the V-System Manual. The V-System Manual itself provided the motivation to combine, as it explicitly disclosed the use of a network file server in its system. As also disclosed in the Coulouris textbook, the art included several standard network file service software to be run on distributed systems, one of which was NFS. Thus, a person of ordinary skill in the art would have had the motivation to combine the V-System Manual with a reference describing NFS architecture and implementation, such as Pawlowski, Rosen, or Sandberg.

Pawlowski, Rosen, and Sandberg disclose the design and implementation of NFS in distributed operating systems. Pawlowski in Figure 3 and 6 discloses the architecture of a NFS network file service and elsewhere in 295-296 and 301-302 discloses an implementation of NFS. Rosen in section 2.1 and 2.5 discloses the design and implementation of NFS, including how the standard depends on remote procedure calls and external data representation packages for access to transport and network services. Sandberg in Figure 1 discloses the architecture and implementation of an NFS network file service. Indeed, this Figure demonstrates in picture form how a file server processor decodes the NFS request, performs procedures for satisfying that NFS request, and encodes an NFS reply message. Elsewhere at pages 1-6 and 11 Sandberg discloses in the text the NFS protocol and implementation. Each of these references disclose limitations 1.6, 5.1-5.3, 8.1-8.3, 18.2, 18.3, 19.2, and 19.4.

The V-System Manual combined with any one or all of the Pawlowski, Rosen, and Sandberg references disclose each and every limitation of—and thus render obvious—the '366 claims-in-suit.

B. The Asserted '918 Patent Claims Are Anticipated And/Or Rendered Obvious by the Prior Art

Anticipation

1. Coulouris Textbook

The Coulouris textbook is written to "serve as a text for an advanced undergraduate or graduate level course on Distributed Systems[.]" Thus the material covered in the book would be expected to be known to those that would be considered of ordinary skill in the art. The Coulouris textbook discloses various distributed systems, including the V-system. The V-system is a multiple facility operating system architecture designed to place as many of the usual operating system functions outside the V-kernel as efficiency permits, including file service. The Coulouris textbook also discloses various network file service software to be run on such distributed systems, including NFS. In light of Network Appliance's infringement contentions, the Coulouris reference discloses each and every limitation of—and thus anticipates—the '918 claims-in-suit, as shown in the claim charts provided below.

	'918 Claim 1	**Coulouris Reference**
1.1	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	Coulouris discloses an environment made up of a federation of machines each doing a specific task for the rest of the machines. "Loosely-coupled distributed systems are particularly effective in exploiting the power and flexibility of single-user computers or workstations, enabling them to access shared data and resources located in other server computers via high-speed local networks (Figure 1.1)." p. 2. One of the services shown in Figure 1.1 is a file server. Like all of the machines in Figure 1.1 it is connected to the network. The file server machine includes one or more mass storage devices to hold the files that it serves.
1.2	means for decoding file system requests from said network;	Notwithstanding the Court's definition of the "means for decoding" as "microprocessor 210 programmed to convert data from the NFS message format to LNFS message format," NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. Coulouris discloses a "transport software module" that decomposes the request into a form that can be communicated via another, lower level protocol, such as the OSI transport layer disclosed on pages 59 through 65: "The transport layer's task is to provide a network-independent message transport service between pairs of network **ports** (also known as **sockets**). Ports are software-definable destination points for communication within a host computer." Coulouris also discloses communications at the LNFS or presentation layer: "The presentation layer software in each computer transforms data from the formats used in the computer it is running on into a set of standard network representations called **external data representation (EDR)** before transmission. Received data is transformed from the network representations into the local data formats." Therefore, Coulouris fully discloses handling NFS requests at either the transport or presentation layer.

1.3	means for performing procedures for satisfying said file system requests, including accessing said mass storage device if required; and	The Court has defined the means for "performing procedures for processing NFS [or file system] requests" to include the "network controller 110, file controller 112, storage processor 114, system memory 116, and VME bus 120" programmed to "perform such procedures based on the algorithms disclosed" in sections 17:20-18:7, 19:24-21:12, 26:26-28:67 of the specification. Again, notwithstanding the Court's definition, NetApp contends that the network interface card in BlueArc's products meet this limitation because it "performs NFS request dispatching and IP routing by programming the microprocessor . . . to dispatch and route information using standard dispatching and routing procedures." Coulouris discloses a file server that would perform such standard dispatching and routing procedures of an NFS server in section 10.3 on pages 264 through 268: "The NFS service is implemented in terms of remote procedure calls between kernels. …If the file is in a remote directory the local kernel makes an [NFS] service call to the appropriate remote kernel." Coulouris further discloses the method of performing a procedure in response to a request on page 84 and in Figure 4.3. Figure 4.3 shows a remote procedure call as the receipt of a request from the network that is handled as local function (in this case as a local filesystem access), and the return of the result to the network. Coulouris also discloses the mechanism for accessing mass storage to satisfy NFS file requests in chapter 6 on pages 135 to 153. Specifically, on page 146, Coulouris states: "The block service supports block read and write operations, implementing them as more primitive disk operations[.]" Coulouris fully discloses this limitation based on NetApp's contention.
1.4	means for encoding any file system reply messages for return transmission on said network,	Notwithstanding the Court's definition of the "means for encoding" as "microprocessor 210 programmed to convert data from the LNFS message format to NFS message format." NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meet this limitation. Coulouris discloses a "transport software module" that decomposes the request into a form that can be communicated via another, lower level protocol,

		such as the OSI transport layer disclosed on pages 59 through 65: "The transport layer's task is to provide a network-independent message transport service between pairs of network **ports** (also known as **sockets**). Ports are software-definable destination points for communication within a host computer." Coulouris also discloses communications at the LNFS or presentation layer: "The presentation layer software in each computer transforms data from the formats used in the computer it is running on into a set of standard network representations called **external data representation (EDR)** before transmission. Received data is transformed from the network representations into the local data formats." Therefore, Coulouris fully discloses handling NFS requests at either the transport or presentation layer.
1.5	said first unit lacking means in said first unit for executing any programs which make calls to any general purpose operating system.	Coulouris discloses the construction of a system built from a lightweight kernel on pages 39 and 40: "A simpler system kernel providing lightweight processes and interprocess communication is a better basis for building workstation software. Server software may also be built on such a basis." Thus while a client workstation runs a general purpose operating system, the file server runs just the lightweight kernel and the additional software needed to handle NFS requests.

	'918 Claim 2	**Coulouris Reference**
2	A network file server according to claim 1, further including a second unit comprising means for executing programs which make calls to a general purpose operating system.	As shown in Figure 1.1 on page 3, Coulouris discloses a second unit – a client workstation – that runs a general purpose operating system and that can access a file server.

	'918 Claim 3	**Coulouris Reference**
3	A network file server according to claim 1, wherein said file system requests from said network comprise NFS requests.	See analysis of prior art for claim 1. Coulouris discloses the details of an implementation of an NFS server in section 10.3 on pages 264 through 268: "The NFS service is implemented in terms of remote procedure calls between kernels. …If the file is in a remote directory the local kernel makes an [NFS] service

		call to the appropriate remote kernel."

	'918 Claim 4	**Coulouris Reference**
4	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	See analysis of prior art for limitation 1.1.
4.1	means for decoding file system requests from said network;	See analysis of prior art for limitation 1.2.
4.2	means for performing procedures for satisfying said file system requests, including accessing said mass storage device if required; and	See analysis of prior art for limitation 1.3.
4.3	means for encoding any file system reply messages for return transmission on said network,	See analysis of prior art for limitation 1.4.
4.4	said first unit lacking means to execute any user-provided application programs on said first unit.	See analysis of prior art for limitation 1.5.

	'918 Claim 5	**Coulouris Reference**
5	A network file server according to claim 4, further including a second unit running a user-provided application program.	See analysis of prior art for claim 2.

	'918 Claim 6	**Coulouris Reference**
6	A network file server according to claim 4, wherein said file system requests from said network comprise NFS requests.	See analysis of prior art for claim 3.

	'918 Claim 7	**Coulouris Reference**
7	A network file server for use with a data network and a mass storage device, said network file server comprising:	Coulouris discloses an environment made up of a federation of machines each doing a specific task for the rest of the machines. "Loosely-coupled distributed systems are particularly effective in

	comprising;	exploiting the power and flexibility of single-user computers or workstations, enabling them to access shared data and resources located in other server computers via high-speed local networks (Figure 1.1)." p. 2. One of the services shown in Figure 1.1 is a file server. Like all of the machines in Figure 1.1 it is connected to the network. The file server machine is connected to one or more mass storage devices to hold the files that it serves.
7.1	a network control module, including a network interface coupled to receive file system requests from said network;	The file server in Figure 1.1 on page 3 receives requests from the network. These requests are delivered into the machine through its network interface. The network control module is the network interface hardware.
7.2	a file system control module, including a mass storage device interface coupled to said mass storage device; and	The file server in Figure 1.1 on page 3 includes a filesystem control module – the file server – that accesses one or more mass storage devices for servicing remote requests.
7.3	a communication path coupled directly between said network control module and said file system control module, said communication path carrying file retrieval requests prepared by said network control module in response to received file system requests to retrieve specified retrieval data from said mass storage device,	The file server in Figure 1.1 on page 3 receives requests from its network interface and passes them to its filesystem control module for handling. The decoding of a filesystem request from a network is disclosed on page 84 and shown in Figure 4.3: "The idea behind remote procedure calls is simple: to allow a program to call procedures in a different computer or in a different address space in the same computer. The remote procedure call is sent in the form of a **request message** to a remote process that is able to receive the call, execute the procedure, and send back a **reply message**." The figure shows a remote procedure call and its receipt of a request from the network and its being passed to the filesystem control module for processing.
7.4	said file system control module retrieving said specified retrieval data from said mass storage device in response to said file retrieval requests and returning said specified retrieval data to said network control module,	The mechanism of accessing mass storage to satisfy local file requests is given in chapter 6 on pages 135 to 153, specifically disclosed on page 146: "The block service supports block read and write operations, implementing them as more primitive disk operations[.]" The receipt of the request and return of the results is done by the remote procedure call functionality disclosed in 7.3.
7.5	and said network control module preparing reply messages containing said specified retrieval data from said file system control	The return of the result from the filesystem module to the client by the network control module over the network is the conclusion of the remote procedure call disclosed in 7.3.

	module for return transmission on said network.	

	'918 Claim 8	Coulouris Reference
8	A network file server according to claim 7, wherein said file system control module returns said specified retrieval data directly to said network control module.	The passing of data between modules or machines is accomplished using a remote procedure call which is disclosed by Coulouris on page 84 and shown in Figure 4.3: "The idea behind remote procedure calls is simple: to allow a program to call procedures in a different computer or in a different address space in the same computer. The remote procedure call is sent in the form of a **request message** to a remote process that is able to receive the call, execute the procedure, and send back a **reply message**." The figure shows a remote procedure call and its data being passed to the filesystem control module for processing. On pages 107 and 108 Coulouris discloses how the remote procedure call between modules that can share memory can be implemented using shared memory so that those modules may communicate directly with each other. "This section discusses the implementation of RPC facilities using lightweight processes (or threads) with shared memory[.] … Therefore a single process in each computer receives all incoming messages and distributes them to other processes by using shared variables. Concurrent processes are required in a server because it should be able to execute remote procedures for several clients at once."

	'918 Claim 9	Coulouris Reference
9.1	A network file server according to claim 7, wherein said network control module further prepares file storage requests in response to received file system requests to store specified storage data on said mass storage device, said network control module communicating said file storage requests to said file system control module,	NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. Coulouris discloses this functionality on pages 59 through 65 in his description of the OSI networking model: "The transport layer's task is to provide a network-independent message transport service between pairs of network **ports** (also known as **sockets**). Ports are software-definable destination points for communication within a host computer." Coulouris also discloses communications at the LNFS or presentation layer: "The presentation layer software in each computer transforms data from the formats used in the computer it is running on into a set of

		standard network representations called **external data representation (EDR)** before transmission. Received data is transformed from the network representations into the local data formats." Therefore, Coulouris fully discloses handling NFS requests at either the transport or presentation layer.
9.2	and wherein said file system control module further stores said specified storage data on said mass storage device in response to said file storage requests.	Coulouris discloses the mechanism for accessing mass storage to store file requests in chapter 6 on pages 135 to 153. Specifically, on page 146, Coulouris states: "The block service supports block read and write operations, implementing them as more primitive disk operations[.]"" The mechanism of responding to file storage requests is disclosed in limitation 7.3.

	'918 Claim 10	**Coulouris Reference**
10	A network file server according to claim 9, wherein said file storage requests are communicated to said file system control module via said communication path.	See analysis of prior art for limitation 7.3.

	'918 Claim 11	**Coulouris Reference**
11	A network file server according to claim 7, wherein said received file system requests to retrieve specified retrieval data comprise NFS requests.	Coulouris discloses the details of an implementation of an NFS server in section 10.3 on pages 264 through 268: "The NFS service is implemented in terms of remote procedure calls between kernels. ...If the file is in a remote directory the local kernel makes an [NFS] service call to the appropriate remote kernel.

	'918 Claim 12	**Coulouris Reference**
12.1	A method for processing requests from a data network, for use by a network file server including a network control module coupled to receive file system requests from said network and a file system control module coupled to said mass storage device, comprising the steps of:	Coulouris discloses a file server as one of the services shown in Figure 1.1. A network control module on the file server receives requests from the network. These requests are delivered into the machine through its network interface. The network control module is the software that manages the network interface. The file server also includes a filesystem control module for servicing remote requests from its attached mass storage device.

12.2	said network control module preparing file retrieval requests in response to received file system requests to retrieve specified retrieval data from said mass storage device;	The file server in Figure 1.1 on page 3 receives requests from its network interface and passes them to its filesystem control module for handling. The decoding of a filesystem request from a network is disclosed on page 84 and shown in Figure 4.3 as: "The idea behind remote procedure calls is simple: to allow a program to call procedures in a different computer or in a different address space in the same computer. The remote procedure call is sent in the form of a **request message** to a remote process that is able to receive the call, execute the procedure, and send back a **reply message**." The figure shows a remote procedure call and its receipt of a request from the network and its being passed to the filesystem control module for processing.
12.3	said network control module communicating said file retrieval requests directly to said file system control module;	The communication of the request from the network control module to the filesystem control module is done by the remote procedure call functionality disclosed in limitation 12.2.
12.4	said file system control module retrieving said specified retrieval data from said mass storage device in response to said file retrieval requests and returning said specified retrieval data to said network control module; and	The mechanism of accessing mass storage to satisfy local file requests is given in chapter 6 on pages 135 to 153, specifically disclosed on page 146: "The block service supports block read and write operations, implementing them as more primitive disk operations[.]" The return of the results from the filesystem control module to the network control module is done by the remote procedure call functionality disclosed in limitation 12.2.
12.5	said network control module preparing reply messages containing said specified retrieval data from said file system control module for return transmission on said network.	The return of the result from the filesystem module to the client by the network control module over the network is the conclusion of the remote procedure call disclosed in limitation 12.2.

	'918 Claim 13	**Coulouris Reference**
13.1	A method according to claim 12, wherein said file system control module returns said specified retrieval data directly to said network control module.	See analysis of prior art for limitation 8.

	'918 Claim 14	**Coulouris Reference**
14.1	A method according to claim 12,	

	further comprising the steps of:	
14.2	said network control module preparing file storage requests in response to received file system requests to store specified storage data on said mass storage device	See analysis of prior art for limitation 9.1.
14.3	said network control module communicating said file storage requests to said file system control module;	See analysis of prior art for limitation 9.1.
14.4	and said file system control module storing said specified storage data on said mass storage device in response to said file storage requests.	See analysis of prior art for limitation 9.2.

	'918 Claim 15	**Coulouris Reference**
15.1	A method according to claim 14, wherein said file storage requests are communicated directly to said file system control module.	See analysis of prior art for limitation 8.

	'918 Claim 16	**Coulouris Reference**
16.1	A method according to claim 12, wherein said received file system requests to retrieve specified retrieval data comprise NFS requests.	See analysis of prior art for limitation 11.

2. Epoch Reference

The Epoch reference titled "How to Satisfy a Voracious NFS Appetite" discloses a multiple-processor architecture that breaks down the functional units in the same way as the '918 patent. The novelty of the Epoch-1 is made clear in the section of the reference describing its performance: "Because this is the first NFS server that DR Labs has tested, we did not have any other servers to compare it with." In light of Network Appliance's infringement contentions, the Epoch reference discloses each and every limitation of—and thus anticipates—claims 1 through 6 of the '918 patent, as shown in the claim charts provided below.

	'918 Claim 1	Epoch Reference
1.1	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	According to the Epoch reference, "[t]he Epoch-1 Infinite Storage Server from Epoch Systems can truly provide … a virtually inexhaustible supply of data storage for Unix users[.]" Even minimally configured, the Epoch-1 is disclosed as having "three or more magnetic disk drives" and "two Ethernet interfaces."
1.2	means for decoding file system requests from said network;	Notwithstanding the Court's definition of the "means for decoding" as "microprocessor 210 programmed to convert data from the NFS message format to LNFS message format." NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. The Front-End Processor is a Motorola 68020 CPU's running at 25Mhz. The FEP "handles all I/O operations for the SCSI controllers [and] two Ethernet interfaces," and thus is connected to the network through its two Ethernet interfaces and to the mass storage devices. The FEP thus decodes the incoming Ethernet packets containing NFS requests.
1.3	means for performing procedures for satisfying said file system requests, including accessing said mass storage device if required; and	The Court has defined the means for "performing procedures for processing NFS [or file system] requests" to include the "network controller 110, file controller 112, storage processor 114, system memory 116, and VME bus 120" programmed to "perform such procedures based on the algorithms disclosed" in sections 17:20-18:7, 19:24-21:12, 26:26-28:67 of the specification. Again, notwithstanding the Court's definition, NetApp contends that the network interface card in BlueArc's products meet this limitation because it "performs NFS request dispatching and IP routing by programming the microprocessor . . . to dispatch and route information using standard dispatching and routing procedures." The Front-End Processor is a Motorola 68020 CPU's running at 25Mhz. The FEP "handles all I/O operations for the SCSI controllers [and] two Ethernet interfaces," and thus is connected to the network through its two Ethernet interfaces and to the mass storage devices. The FEP thus decodes the incoming Ethernet packets containing NFS requests. After decoding the request from the network, the FEP next runs a procedure to satisfy

		the request. If necessary, the FEP will access the mass storage device to read or write data.
1.4	means for encoding any file system reply messages for return transmission on said network,	Notwithstanding the Court's definition of the "means for encoding" as "microprocessor 210 programmed to convert data from the LNFS message format to the NFS message format," NetApp contends that BlueArc's network interface card which communicates at the OSI transport layer meets this limitation. The Front-End Processor is a Motorola 68020 CPU's running at 25Mhz. The FEP "handles all I/O operations for the SCSI controllers [and] two Ethernet interfaces," and thus is connected to the network through its two Ethernet interfaces and to the mass storage devices. The FEP thus encodes the reply messages to be sent across the network into Ethernet packets.
1.5	said first unit lacking means in said first unit for executing any programs which make calls to any general purpose operating system.	"The other processor – known as the Front-End Processor (FEP) – runs Ready Systems' VRTX real-time multitasking kernel and handles all I/O operations for the SCSI controllers, two Ethernet interfaces, modem port and system console port." The processor managing the mass storage and network interfaces is running a real-time operating system that is not able to execute any programs that make calls to a general purpose operating system.

	'918 Claim 2	Epoch Reference
2	A network file server according to claim 1, further including a second unit comprising means for executing programs which make calls to a general purpose operating system.	"The first processor, called the Applications Processor (AP), runs a somewhat modified version of 4.3BSD Unix[.] ... Epoch users can have use of the *telnet, ftp, rlogin, rsh,* and *rcp* utilities for transferring files between the Epoch and other hosts that support TCP/IP or for logging in to a remote host. In addition, a few new commands and utilities have been added for the management of the Epoch-1's files and disks." The AP processor is able to execute programs that make calls to a general purpose operating system (4.3BSD Unix).

	'918 Claim 3	Epoch Reference
3	A network file server according to claim 1, wherein said file system requests from said network	The Epoch reference titled "How to Satisfy a Voracious NFS Appetite" discloses the Epoch-1 NFS file server.

	comprise NFS requests.	NFS file server.

	'918 Claim 4	**Epoch Reference**
4	A network file server for use with a data network and a mass storage device, said network file server including a first unit comprising:	See analysis of prior art for limitation 1.1.
4.1	means for decoding file system requests from said network;	See analysis of prior art for limitation 1.2.
4.2	means for performing procedures for satisfying said file system requests, including accessing said mass storage device if required; and	See analysis of prior art for limitation 1.3.
4.3	means for encoding any file system reply messages for return transmission on said network,	See analysis of prior art for limitation 1.4.
4.4	said first unit lacking means to execute any user-provided application programs on said first unit.	See analysis of prior art for limitation 1.5.

	'918 Claim 5	**Epoch Reference**
5	A network file server according to claim 4, further including a second unit running a user-provided application program.	See analysis of prior art for limitation 2.

	'918 Claim 6	**Epoch Reference**
6	A network file server according to claim 4, wherein said file system requests from said network comprise NFS requests.	The Epoch reference titled "How to Satisfy a Voracious NFS Appetite" discloses the Epoch-1 NFS file server.

3. V-System Manual

The V-System Manual discloses the V-System, a multiple facility operating system architecture designed to place as many of the usual operating system functions outside the V-kernel as efficiency permits. The V-System Manual discloses the same system as disclosed in the Coulouris Textbook, except in more detail. The V-System Manual thus discloses each and every limitation of —and thus anticipates—'918 claims 1, 2, 4, 5, 7, 8, 9, 10, 12, 13, 14, and 15, as explained above in the Coulouris claim charts.

4. Sandberg

Sandberg discloses a multiple-processor architecture implementation of a NFS network file service. Specifically, Sandberg discloses using an Excelan Board (see generally Excelan reference manual) and a VAX 750 to implement NFS. Accordingly, as seen in the claim charts provided below, Sandberg discloses each and every limitation of—and thus anticipates—'918 claims 7, 8, 9, 10, 12, 13, 14, 15, and 16.

	'918 Claim 7	**Sandberg Reference**
7	A network file server for use with a data network and a mass storage device, said network file server comprising;	Sandberg in Figure 1 discloses the architecture and implementation of an NFS network file service, including a data network and a mass storage device.
7.1	a network control module, including a network interface coupled to receive file system requests from said network;	The Excelan board "handles the Ethernet, IP, and UDP layers," as seen on page 11. Accordingly, the board includes a network interface that receives NFS requests from the network.
7.2	a file system control module, including a mass storage device interface coupled to said mass storage device; and	The VAX 750 acts as the file system control module and includes an interface that is connected through the network to a mass storage device as seen in Figure 1.
7.3	a communication path coupled directly between said network control module and said file system control module, said communication path carrying file retrieval requests prepared by said network control module in response to received file system requests to retrieve specified retrieval data from said mass storage device,	The network control module – the Excelan board – is connected directly to the file system control module – the VAX 750 through a communication path, its 32-bit Unibus. The Excelan board will receive NFS requests to retrieve data from a mass storage device, will process those requests to generate a file retrieval request and then will send that file retrieval request to the VAX 750.
7.4	said file system control module retrieving said specified retrieval data from said mass storage device in response to said file retrieval	Upon receiving the file retrieval request, the file system control module – the VAX 750 – will retrieve the specified data from the mass storage device, as seen in Figure 1 and disclosed in the

	requests and returning said specified retrieval data to said network control module,	article.
7.5	and said network control module preparing reply messages containing said specified retrieval data from said file system control module for return transmission on said network.	The file system control module – the VAX 750 – is connected to the internet only through the network control module – the Excelan board – so that Excelan board prepares the reply message to be sent across the network, as seen in Figure 1. Indeed, the Excelan board handles the IP and UDP encoding.

	'918 Claim 8	Sandberg Reference
8	A network file server according to claim 7, wherein said file system control module returns said specified retrieval data directly to said network control module.	The network control module – the Excelan board – is connected directly to the file system control module – the VAX 750 through a communication path, a 32-bit Unibus, and thus the file control module returns the specified retrieval data directly to the network control module.

	'918 Claim 9	Sandberg Reference
9.1	A network file server according to claim 7, wherein said network control module further prepares file storage requests in response to received file system requests to store specified storage data on said mass storage device, said network control module communicating said file storage requests to said file system control module,	The network control module – the Excelan board – will receive NFS requests to store data on a mass storage device, will process those requests to generate a file storage request and then will send that file storage request to the VAX 750.
9.2	and wherein said file system control module further stores said specified storage data on said mass storage device in response to said file storage requests.	In response to such a file storage request, the VAX 750 will store the specified storage data, as seen in Figure 1.

	'918 Claim 10	Sandberg Reference
10	A network file server according to claim 9, wherein said file storage requests are communicated to said file system control module via said communication path.	The network control module – the Excelan board – is connected directly to the file system control module – the VAX 750 through a communication path, a 32-bit Unibus, and thus the network control module will send file storage request over that communication path to the network control

		module.

	'918 Claim 11	Sandberg Reference
11	A network file server according to claim 7, wherein said received file system requests to retrieve specified retrieval data comprise NFS requests.	See analysis of prior art for claim 7.

	'918 Claim 12	Sandberg Reference
12.1	A method for processing requests from a data network, for use by a network file server including a network control module coupled to receive file system requests from said network and a file system control module coupled to said mass storage device, comprising the steps of:	Sandberg on page 11 discloses a method for processing requests from a data network using a network file server – the VAX 750 – connected to a network control module – the Excelan board, wherein that network control module receives file system requests from the network.
12.2	said network control module preparing file retrieval requests in response to received file system requests to retrieve specified retrieval data from said mass storage device;	The network control module – the Excelan board – "handles the Ethernet, IP, and UDP layers," as disclosed on page 11. Given that the Excelan board handles the Ethernet, the board receives NFS requests from the network. Given that the Excelan board handles the IP and UDP layers, it decodes the file system requests and prepares a file retrieval request.
12.3	said network control module communicating said file retrieval requests directly to said file system control module;	The network control module – the Excelan board – is connected directly to the network file server – the VAX 750 – through its 32-bit Unibus and thus network control module passes file retrieval requests directly to the network file server.
12.4	said file system control module retrieving said specified retrieval data from said mass storage device in response to said file retrieval requests and returning said specified retrieval data to said network control module; and	In response to file retrieval requests, the network file server – the VAX 750 – retrieves the requested data from the mass storage device and returns that data to the network control module, as depicted in Figure 1.
12.5	said network control module preparing reply messages containing said specified retrieval data from said file system control module for	The network control module – the Excelan board – then prepares a reply message that contains the specified data retrieved by the file system control module – the VAX 750 – to be