	return transmission on said network.	sent on the network.

	'918 Claim 13	Sandberg 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 claim 8.

	'918 Claim 14	Sandberg 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	Sandberg 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	Sandberg 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 claim 11.

5. Figure 1 of the '918 Patent

The specification of the '918 patent discloses a diagram of a prior art file server architecture. Accordingly, as seen in the claim charts provided below, Figure 1 of the '918 patent discloses each and every limitation of—and thus anticipates—'918 claims 7 through 16.

	'918 Claim 7	**'918 Figure 1**
7	A network file server for use with a data network and a mass storage device, said network file server comprising;	The '918 Patent in Figure 1 discloses prior art that discloses network file server for use with an Ethernet 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 '918 Patent in Figure 1 discloses a network controller – network controller 34 – including a network interface that receives 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 '918 Patent in Figure 1 discloses a Host CPU Card 10 that along with SMD Disk Controller 22 or SCSI Host Adapter 26.
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 '918 Patent in Figure 1 discloses a communication path – VME bus 20 – directly connecting the network control module and the file system control module. This path would carry file retrieval requests.
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,	After the "file system control module" has retrieved the requested data from the mass storage device into its Memory 16, it transfers the data to the "network control module" (Network Controller 34) as data structures from its Memory 16.
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	The "network control module" would then prepare the reply messages in the form of serial Ethernet packets.

	said network.	

	'918 Claim 8	'918 Figure 1
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 "file system control module" returns the requested data from the mass storage device via its Memory 16 to the "network control module" (Network Controller 34).

	'918 Claim 9	'918 Figure 1
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 controller 34 will receive NFS requests to retrieve 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 file server via Memory 16.
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 file server will store the specified storage data from Memory 16 to the attached mass storage device.

	'918 Claim 10	'918 Figure 1
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 '918 Patent in Figure 1 discloses a communication path – VME bus 20 – directly connecting the network control module and the file system control module. This path would carry file storage requests.

	'918 Claim 11	'918 Figure 1
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. According to column 5:30-31, "[t]he system communicates over the Ethernets using . . . NFS protocol stacks."

	'918 Claim 12	'918 Figure 1
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:	The '918 Patent in Figure 1 discloses a method for network processing data requests prior art that discloses a network controller module – Network Controller 34 – connected via Ethernet to the network and a file system control module – Host CPU Card 10 that along with SMD Disk Controller 22 or SCSI Host Adapter 26 – is coupled to a 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 network controller module – Network Controller 34 – could receive file system requests from the Ethernet Local Area Network ("LAN") cable, and transfer the data structures into the Memory 16 as file retrieval requests.
12.3	said network control module communicating said file retrieval requests directly to said file system control module;	The network control module is connected directly via a VME bus 20 to the file retrieval module.
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 "file system control module" would retrieve the requested data from the mass storage device, and then it transfers the data to the "network control module" (Ethernet Network Controller 34).
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 network control module– Network Controller 34 – then prepares a reply message that contains the specified data retrieved by the file system control module to be sent on the network.

	'918 Claim 12	'918 Figure 1
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 -- to receive network requests and a network file server connected via the network control module to the mass storage device.
12.2	said network control module preparing file retrieval requests in	The network control module – the Excelan board – "handles the Ethernet, IP, and UDP

	response to received file system requests to retrieve specified retrieval data from said mass storage device;	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 – 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 return transmission on said network.	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 sent on the network.

	'918 Claim 13	**'918 Figure 1**
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 claim 8.

	'918 Claim 14	**'918 Figure 1**
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	**'918 Figure 1**
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	**'918 Figure 1**
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 claim 11. According to column 5:30-31, "[t]he system communicates over the Ethernets using . . . NFS protocol stacks."

Obviousness

6. V-System Manual + Pawlowski/Sandberg/Rosen

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 limitations in claim 3, 6, 11, and 16 that require NFS. It follows that as explained in the Coulouris claim charts above, the V-System Manual discloses each and every limitation of independent claims 1, 4, 7, and 12.

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. Each of these references disclose limitations 3, 6, 11, and 16.1.

The V-System combined with any one or all of the Pawlowski, Rosen, and Sandberg references disclose each and every limitation of—and thus render obvious—claims 3, 6, 11, and 16 of the '918 patent.

C. The Asserted '037 Patent Claims Are Anticipated And/Or Rendered Obvious by the Prior Art

Anticipation

1. 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. This sweeping framework of the V-system included all the functionality that the '037 patent claims as its invention. The V-System Manual discloses each and every limitation of—and thus anticipates—the '037 claims-in-suit, as shown in the claim charts provided below.

	'037 Claim 1	V-System Manual
1.1	A computer system employing a multiple facility operating system architecture, said computer system comprising:	The V-system employs multiple computers to distribute as many of the usual operating system functions outside the V-kernel as efficiency permits, as disclosed in the V-System Manual at 1-1 to 1-6. Figure 1-1 and page 1-2 in particular disclose the distribution of operating system functions to different facilities. These facilities are further disclosed in chapters 3, 4, 6, 12, 28, 29, 31, 33, 39, 43, and appendix C.
1.2	a) a plurality of processor units	Any distributed operating system architecture has a plurality of processing units. For example, Figure 1-1 on page 1-2 of the V-System Manual discloses multiple servers and workstations, each of which is a separate processing unit.

1.3	provided to co-operatively execute a predetermined set of operating system peer-level facilities,	The Court construed "operating system peer-level facility" as "a major functional subsystem of the operating system constituted as a separately executed software entity." The V-System consists of a distributed kernel and a distributed set of server processes running on a plurality of processing units. This plurality of processing units interact with each other to provide a predetermined set of functions usually provided by the operating system, such as file service, print service, gateway service, routing service, etc.
1.4	wherein each said processor units is associated with a respective different one of said operating system peer-level facilities and not another of said operating system peer level facilities;	Each of the processing units in the V-System network provides a distinct function usually provided by the operating system. For example, in the V-System, the file server provides only file service and not print service, while the print server provides only print service and not file storage.
1.5	and wherein each of said operating system peer-level facilities constitutes a respective separately executed software entity	Each of the processing units in the V-System network runs its own respective separately executed software program to implement its respective function. For example, the file server runs a program that implements file service, while the gateway server runs a program that provides gateway service. This division of functionality is disclosed on page 31-8 of the V-System manual: ``One of the principles guiding the V-System design has been to place as many [of] the usual operating system functions outside the kernel as efficiency permits. Moreover, functions have been partitioned as far as practical into separate servers. Consequently, the kernel and each server have been kept reasonably small and independent of each other, which has in turn simplified debugging, maintenance, and experimentation with new servers."
1.6	which includes a respective distinct set of peer-level facility related functions, each said processor unit including:	Each of the processing units in the V-System network provides a unique set of functions. For example, the file server provides functions to create, read, write, and delete files while the print server provides functions to print files, provide duplexing, and select paper trays. The functions provided by these servers do not overlap and are quite distinct from each other.
1.7	i) a processor capable of executing a control program; and	Each of the processing units in the V-System network has a processor capable of executing a

	control program; and	control program. The V-System Manual at page 1-3 discloses the system: ``The distributed kernel consists of the collection of kernels resident on each participating machine (see Figure 1-2)."
1.8	ii) a memory store capable of storing said control program,	Each of the processing units in the V-System network is able to run as a standalone machine that has a memory store capable of storing the control program.
1.9	said processor being coupled to said memory store to obtain access to said control program,	The processor in the processing unit is coupled to the memory store and fetches control program instructions from that memory store.
1.10	said memory store providing for the storage of a first control program portion that includes a one of said respective distinct sets of operating system peer-level facility related functions and that corresponds to a one of said predetermined operating system peer-level facilities	Each of the processing units in the V-System network uses its memory store for storing the server program that implements its distinctive function that is usually provided by the operating system. For example, the memory store will hold the program that implements the file service and provides the create, read, write, and delete functions that are predetermined for that service.
1.11	and a second control program portion that provides for the implementation of a multi-tasking interface function,	The Court has construed "multi-tasking interface" to mean "software that is tailored to each peer-level facility that supports direct communication with other peer-level facilities and capable of managing at least two concurrent or interleaved tasks." Each of the processing units in the V-System network has a kernel – a second control program portion – that "provides process management, interprocess communication, and low-level device management facilities", as disclosed on page 1-3 of the V-System Manual. The interprocess communication functionality of the kernel ``supports direct communication with other peer-level facilities" and the management of the ``collections of processes outside the kernel" shows that the kernel is ``capable of managing at least two concurrent or interleaved tasks." Furthermore, all other operating system services are "implemented as (collections) of processes outside the kernel."
1.12	said multi-tasking interface function being responsive to control messages for selecting for execution a one of said peer-level facility	Each of the kernels implements interprocess communication functionality that responds to control messages by selecting for execution the desired software functions that are implemented by

	related functions of said one of said predetermined operating system peer-level facilities and responsive to said one of said predetermined operating system peer-level facilities for providing control messages to request or in response to the performance of said predetermined peer-level facility related functions of another operating system peer-level facility; and	a distinct software process. Each kernel is also responsive to that software process and can send control messages to request or respond to software functions implemented by a different software process. For example, the kernel will receive a control message from another software process requesting that a file be read. The kernel will select one of the file server processes to run and deliver the control message to that process. The selected file-server process will read the requested data and send a control message containing that data to the kernel. The kernel will then return the control message containing the data to the software process that requested it.
1.13	b) a communications bus that provides for the interconnection of said plurality of processor units, said communications bus transferring said control messages between the multi-tasking interface functions of said predetermined set of operating system peer-level facilities.	The Court construed ``bus" to mean ``a set of signal lines to which two or more devices may be connected and over which information is transferred between those devices." Ethernet fulfills this definition. Each of the kernels is connected to a communications bus (typically Ethernet) that provides for the interconnection of the plurality of machines, as shown in Figure 1-3. This communications bus (Ethernet) transfers control messages between the interprocess communication facilities – the multi-tasking interface functions – resident on each of the processor units that implement their own distinctive function that is usually provided by the operating system. This functionality is disclosed in Figure 1-3 on page 1-3 of the V-System Manual: ``The host kernels are integrated via a low-overhead inter-kernel protocol (IKP) that supports transparent interprocess communication between machines."

	'037 Claim 2	**V-System Manual**
2.1	The computer system of claim 1 wherein a first one of said predetermined set of operating system peer-level facilities includes a network communications facility and a second one includes a filesystem facility.	A network communication facility (the processes that implement the Internet server facility) and a filesystem facility (the processes that implement the file server facility) are disclosed in the V-System Manual on pages 1-2 through 1-4: ``The V-System adheres to the server model: The world consists of a collection of resources accessible by clients and managed by servers. A server defines the abstract representation of its resource(s) and

		the operations on this representation. A resource may only be accessed or manipulated through its server. ... Servers include: Internet server – provides network and transport level support for traditional network architectures, namely ARPA Internet. ... storage server – provides file storage.''

	'037 Claim 3	V-System Manual
3.1	The computer system of claim 2 wherein said network communications facility is coupled to a network to permit the receipt of network requests, said network communications facility providing for the identification of a predetermined filesystem type network request, said multi-tasking interface function of said network communications facility being responsive to said predetermined filesystem type network request to provide a predetermined control message to said filesystem facility to request the performance of a predetermined filesystem function.	The network communications facility discussed above for limitations 2.1 is coupled to a network so that it can receive filesystem server requests. When the network communications facility receives a filesystem server request it forwards that request to the filesystem server to request that that server perform the requested filesystem function. Communication between servers is disclosed in the V-System manual on page 27-1: ``The process and interprocess communication-related functions in the V C library provide services and/or interfaces between processes and the V kernel.'' Further, on page 27-11, the V-System Manual: ``[S]end the message in **msg** to the specified process, blocking the invoking process until the message is both received and replied to.''

	'037 Claim 4	V-System Manual
4.1	The computer system of claim 3 further comprising a data store that provides for the storage of data, said predetermined filesystem type network request directing said network communications facility to transfer predetermined data with respect to said network, said data store being coupled to said network communications facility for storing said predetermined data.	The V-System Manual discloses a jointly accessible data store that provides for the storage of data in response to a predetermined filesystem type network request that directs the network communications facility to transfer the requested data. The network communications facility is able to access the referenced data placed in the jointly accessible data store by the filesystem facility and transfer it on the network. The use of a shared memory to pass large pieces of data between processes (facilities) in the V-System are disclosed in the V-System manual on page 1-5: ``[M]ultiple processes may share the same address space or *team*, to facilitate fine-grain sharing of code and data. ... To facilitate transfer of large amounts of data, a separate transfer facility is provided.

		Specifically, a process can pass, in a message, access to an area in its team space. This facility follows the procedure paradigm in being used primarily to access what are logically "call-by-reference" parameters.'' Thus in the V-System, a process providing the filesystem facility reads in a block of data to a data store. It then sends a message to a process providing the networking facility with a request that the data be sent to a client. The process providing the networking facility then sends the data from the data store across the network to the client.

	'037 Claim 5	**V-System Manual**
5.1	The computer system of claim 3 or 4 wherein said predetermined set of peer-level facilities further includes a storage facility and wherein said filesystem facility provides for the performance of said predetermined filesystem function, said multi-tasking interface function of said filesystem facility being responsive to said filesystem facility to provide control messages to said storage facility to request the performance of a predetermined storage access function.	The filesystem server facility sends control messages to a storage facility to request that the storage facility perform one of its storage access functions. The V-System manual discloses this functionality on page 1-4, stating that the "storage server provides file storage,'' and the "device server interfaces to a specific physical device such as … [a] disk.'' The interprocess communication function of the file system facility – the multi-tasking interface function – sends control message requests to the storage facility.

	'037 Claim 6	**V-System Manual**
6.1	The computer system of claim 5 wherein said predetermined storage access function directs said storage facility to transfer said predetermined data, said data store being coupled to said storage facility for storing said predetermined data.	As discussed above for limitation 5.1, the storage facility has certain access functions. In performing such an access function, the storage facility will transfer the requested data to an attached data store – the buffer pool.

	'037 Claim 7	**V-System Manual**

7.1	A computer system implementing a co-operative facility based operating system architecture, said computer system comprising:	See analysis of the prior art for limitation 1.1 (multiple facility operating system architecture).
7.2	a) a plurality of processors, each being coupled to a respective control program store and a respective data store, said plurality of processors being interconnected by a communications bus; and	See analysis of the prior art for limitations 1.2 (plurality of processors), 1.8 (respective control program store), 1.9 (processor coupled to store), 4.1 (respective data store), and 1.13 (interconnected communications bus).
7.3	b) a multiple facility operating system having a kernel and providing for the message based co-operative operation of said plurality of processors, said multiple facility operating system providing for the operating system internal execution of a plurality of operating system peer-level facilities by execution of each of said peer-level facilities by a respective different one of said plurality of processors, each of said peer-level facilities constituting a respective software entity executed separately from said kernel, wherein each of said plurality of facilities implements a multi-tasking interface coupleable between said communications bus and a respective and unique peer-level control function set to permit message transfer between each of said plurality of facilities.	See analysis of the prior art for limitations 1.1 (multiple facility operating system), 1.11 (message-based kernel providing co-operative operation of a plurality of processors), 1.2, 1.3, 1.4, & 1.5 (internal execution of a plurality of operating system peer-level facilities) 1.6, 1.7, & 1.10 (execution of each facility by different processors), 1.11 (peer-level facilities executed separately from the kernel), 1.11, 1.12, & 5.1 (multi-tasking interface with unique peer-level control function set for message transfer between facilities), and 1.13 (facilities coupled to a communications bus).

	'037 Claim 8	**V-System Manual**
8.1	The computer system of claim 7 wherein said plurality of facilities includes a network facility and a filesystem utility, wherein said network facility includes a communications network peer-level control function coupled between a first multi-tasking interface and a network interface and said filesystem facility	See analysis of the prior art for claim 7 and limitations 2.1 (network facility and filesystem utility), 3.1 (network facility includes communications network coupled with a multi-tasking interface to a network interface), and 5.1 & 6.1 (filesystem facility includes a data storage control function coupled to a second multi-tasking interface and filesystem).

	includes a data storage peer-level control function coupled between a second multi-tasking interface and a filesystem.	

	'037 Claim 9	V-System Manual
9.1	The computer system of claim 8 wherein said network facility is coupled through said network interface to a communications network, wherein said network facility is responsive to a predetermined network filesystem message received via said network interface to provide a predetermined filesystem message, and wherein said filesystem facility is responsive to said predetermined filesystem message to transfer data with respect to said filesystem.	See analysis of the prior art for limitation 3.1.

	'037 Claim 10	V-System Manual
10.1	The computer system of claim 9 further comprising a common data store, said network facility providing for the transfer of data between said network interface and said data store, said filesystem facility providing for the transfer of data between said data store and said filesystem, said communications network peer-level control function directing a message to said filesystem peer-level control function identifying a predetermined location of data in said data store with respect to said predetermined filesystem message.	See analysis of the prior art for limitations 4.1 (common data store), 5.1 (transfer of data from network to filesystem via data store), and 4.1 (message from network to filesystem identifying location of data in data store).

	'037 Claim 11	V-System Manual

11.1	A computer system employing a multiple facility operating system to provide for co-operative operation of a plurality of processors,	See analysis of the prior art for limitations 1.1 (multiple facility operating system architecture), 1.3 (co-operative operation), and 1.2 (plurality of processors).
11.2	wherein said operating system includes a kernel and a plurality of additional component facilities executed separately from said kernel, each of said component facilities including a facility sub-component, that defines the execution operation of a one of said component facilities, coupled to a multi-tasking interface sub-component,	See analysis of the prior art for limitations 1.11 (kernel and plurality of additional component facilities executed separately from the kernel), and 5.1 (facility subcomponent that defines operation of component facility coupled to a multi-tasking interface sub-component).
11.3	wherein said computer system comprises: a) a plurality of processors executing said operating system, each of said processors including local memory for the storage and execution of a respective component facility;	See analysis of the prior art for limitations 1.1 (plurality of processors), 1.11 (executing said operating system), and 1.6, 1.7, 1.8, 1.9, & 1.10 (processors including local memory for the storage and execution of their component facility).
11.4	b) a data memory accessible by each of said processors for the storage and retrieval of data blocks exchangeable between said processors; and	See analysis of the prior art for limitation 4.1. The V-System ran on the VAX family of computers some of which were available with two processors that shared a common memory. Accordingly, the V-System could include a file server that ran the filesystem on one processor and the device server on the other processor and communicated through the shared memory. A shared memory system was possible in the V-System through the device server (the device server is disclosed on pages 1-4 and 36-1 to 36-4 of the V system manual).
11.5	c) a communications bus coupling said processors and said data memory to permit the exchange of control messages between said processors and data through said data memory,	See analysis of the prior art for limitation 1.13. NetApp contends that a disk is shared memory. Therefore, the communications bus in the V-System connects the shared data memory – the disk – to the processors and thus permits the exchange of control messages between the processors and data through the disk.

11.6	and wherein said processors each implement a respective different local sub-set of fewer than all of said component facilities that depends through the exchange of control messages on the execution of another sub-set of said componentized facilities by another of said processors to co-operatively implement said operating system.	See analysis of the prior art for limitations 1.4 (each processor implements only a subset of the component facilities), and 1.11 & 5.1 (exchange of control messages between processors to control execution of component facilities).

	'037 Claim 12	**V-System Manual**
12.1	The computer system of claim 11 wherein control messages communicate any of a facility sub-component function request, a facility sub-component function response, and a facility sub-component identifier of a memory space within said data memory to use in connection with said sub-component function request.	See analysis of the prior art for limitations 4.1 (message passing through shared memory) and 5.1 (use of message passing between facility sub-component functions).

	'037 Claim 13	**V-System Manual**
13.1	The computer system of claim 12 wherein said plurality of component facilities includes a network facility and a filesystem facility, wherein a network facility sub-component is executed by a first processor to process network requests and data transfers and a filesystem facility sub-component is executed by a second processor to process filesystem requests and data transfers derivative of said network requests and data transfers.	See analysis of the prior art for limitations 1.2 (multiple processors), and 2.1 (networking and filesystem run separately).

	'037 Claim 14	**V-System Manual**
14.1	The computer system of claim 1, [illegible]	See analysis of the prior art for claim 1. The V- [illegible]

	wherein one of the processor units in said plurality of processor units is provided further to execute a further operating system peer-level facility not in said predetermined set of operating system peer-level facilities.	System provides a workstation environment that appears to the user like a general purpose Unix environment. The workstation environment is created by performing some of the functions locally and sending the rest off to be performed remotely with the result returned to the workstation environment. This functionality is disclosed on page 3-3 of the V system manual: ``If the executive fails to find an appropriate load file for a command, it will attempt to execute the command on the server providing its current context by executing the **fexecute** program. Thus, for example, when a V server on Unix is providing the current context, all the standard Unix commands like **finger**, **man**, or **ps** are available. The output of the Unix command is printed on the standard output file.''

	'037 Claim 15	**V-System Manual**
15.1	The computer system of claim 7, wherein said multiple facility operating system provides further for the operating system internal execution of a further operating system peer-level facility not in said plurality of operating system peer-level facilities, by execution of said further peer level facility by one of the processors in said plurality of processors.	See analysis of the prior art for limitation 14.1.

2. Epoch Reference

The Epoch reference discloses a multiple-processor architecture that breaks down the functional units in the same way as the '037 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.'' Given the scope of the asserted claims indicated by Network Appliance's infringement contentions, the Epoch reference discloses each and every limitation of—and thus anticipates—the '037 claims-in-suit, as shown in the claim charts provided below.

	'037 Claim 1	Epoch Reference
1.1	A computer system employing a multiple facility operating system architecture, said computer system comprising:	The Epoch reference discloses the Epoch-1 NFS file server. The Epoch file server ``runs a somewhat modified version of 4.3BSD Unix'' which is a multiple facility operating system.
1.2	a) a plurality of processor units	``The Epoch-1 sports two processors, both Motorola 68020 CPU's running at 25Mhz.''
1.3	provided to co-operatively execute a predetermined set of operating system peer-level facilities,	``The first processor, called the Applications Processor (AP) … is primarily concerned with handling the NFS daemons (detached processes that take care of NFS requests from other Unix hosts) as well as other Unix housekeeping chores. … The other processor – known as the Front-End Processor (FEP) handles all I/O operations for the SCSI controllers, two Ethernet interfaces, modem port and system console port.'' Collectively the two processors cooperatively execute a predetermined set of software functions that make up the NFS service.
1.4	wherein each said processor units is associated with a respective different one of said operating system peer-level facilities and not another of said operating system peer level facilities;	As discussed above for limitation 1.3, each processor handles a different and disjointed subset of the peer-level facilities.
1.5	and wherein each of said operating system peer-level facilities constitutes a respective separately executed software entity	``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[.]'' Both processors run multiple separately executed software entities.
1.6	which includes a respective distinct set of peer-level facility related functions, each said processor unit including:	As discussed above for 1.3, each processor handles a different and disjoint subset of the peer-level facilities.
1.7	i) a processor capable of executing a control program; and	The Motorola 68020 processors used by the Epoch server are capable of executing a control program.

1.8	ii) a memory store capable of storing said control program,	The Motorola 68020 processors used by the Epoch server each have a memory capable of storing a control program.
1.9	said processor being coupled to said memory store to obtain access to said control program,	The Motorola 68020 processors used by the Epoch server are coupled to their memory to gain access to their control programs.
1.10	said memory store providing for the storage of a first control program portion that includes a one of said respective distinct sets of operating system peer-level facility related functions and that corresponds to a one of said predetermined operating system peer-level facilities	Each of the processors uses its memory store for the storage of the server program that includes the distinctive set of software processes that are provided by that server. For example, the AP processor runs an NFS daemon that provides the remote mount service.
1.11	and a second control program portion that provides for the implementation of a multi-tasking interface function,	The Court construed "multi-tasking interface" as ``software that is tailored to each peer-level facility that supports direct communication with other peer-level facilities and capable of managing at least two concurrent or interleaved tasks.'' As discussed above for limitation 1.5, each processor executes such software. The processes both execute their tasks separately from the kernels themselves.
1.12	said multi-tasking interface function being responsive to control messages for selecting for execution a one of said peer-level facility related functions of said one of said predetermined operating system peer-level facilities and responsive to said one of said predetermined operating system peer-level facilities for providing control messages to request or in response to the performance of said predetermined peer-level facility related functions of another operating system peer-level facility; and	Each of the kernels implements interprocess communication functionality that responds to control messages by selecting for execution the desired software functions that are implemented by a distinct software process. Each kernel is also responsive to that software process and can send control messages to request or respond to software functions implemented by a different software process. For example an NFS mount request will be received on the Ethernet interface. The mount daemon will be executed and passed the request. When completed, a response will be composed and returned to the operating system to be returned to the client.
1.13	b) a communications bus that provides for the interconnection of said plurality of processor units, said communications bus	``The two processors use shared memory in order to pass data between each other.'' The processors are connected to the shared memory by a shared communication bus over which control messages

	transferring said control messages between the multi-tasking interface functions of said predetermined set of operating system peer-level facilities.	are passed by their predetermined set of software processes.

	'037 Claim 2	**Epoch Reference**
2.1	The computer system of claim 1 wherein a first one of said predetermined set of operating system peer-level facilities includes a network communications facility and a second one includes a filesystem facility.	The FEP has a predetermined set of operating system peer-level facilities that includes a network communication facility (the processes that implement the Internet server facility). The AP and FEP collectively run a set of processes that provide the filesystem facility.

	'037 Claim 3	**Epoch Reference**
3.1	The computer system of claim 2 wherein said network communications facility is coupled to a network to permit the receipt of network requests, said network communications facility providing for the identification of a predetermined filesystem type network request, said multi-tasking interface function of said network communications facility being responsive to said predetermined filesystem type network request to provide a predetermined control message to said filesystem facility to request the performance of a predetermined filesystem function.	The FEP processor has its network communications facility coupled to its Ethernet interface which permits it to receive requests of the type that are predetermined to be for the filesystem server. When the network communications facility receives a filesystem server request, it forwards that request to the filesystem facility to request that the filesystem facility handle it.

	'037 Claim 4	**Epoch Reference**
4.1	The computer system of claim 3 further comprising a data store that provides for the storage of data,	``The two processors use shared memory in order to pass data between each other. The system unit has at least 8MB of memory, of which 4MB is devoted to

	said predetermined filesystem type network request directing said network communications facility to transfer predetermined data with respect to said network, said data store being coupled to said network communications facility for storing said predetermined data.	the Unix operating system and 4MB is used for filesystem buffers. Unix tends to do much more buffering of disk I/O than is common under VMS and typically devotes 10 percent of main memory for this operation. Main memory can be expanded to 24MB, and all except the first 4MB would be used for file system buffering.'' The filesystem facility places the requested data in the shared memory and requests the network facility to transfer the requested data from the shared memory over the network to the client making the filesystem request.

	'037 Claim 5	**Epoch Reference**
5.1	The computer system of claim 3 or 4 wherein said predetermined set of peer-level facilities further includes a storage facility and wherein said filesystem facility provides for the performance of said predetermined filesystem function, said multi-tasking interface function of said filesystem facility being responsive to said filesystem facility to provide control messages to said storage facility to request the performance of a predetermined storage access function.	The FEP processor that provides part of the filesystem services includes a storage facility (``three or more magnetic disk drives'') wherein the filesystem facility is responsive to and provides for the performance of its predetermined filesystem functions by sending control messages to the storage facility to request that it perform one of the predetermined storage access functions.

	'037 Claim 6	**Epoch Reference**
6.1	The computer system of claim 5 wherein said predetermined storage access function directs said storage facility to transfer said predetermined data, said data store being coupled to said storage facility for storing said predetermined data.	See the description of the shared memory disclosed in limitation 4.1. The storage facility managed by the FEP with predefined access functions has a data store (the shared memory) coupled to it from which data store it gets the data to be stored.

	'037 Claim 7	**Epoch Reference**

7.1	A computer system implementing a co-operative facility based operating system architecture, said computer system comprising:	The Epoch reference discloses the Epoch-1 NFS file server. The Epoch file server ``runs a somewhat modified version of 4.3BSD Unix'' which is a cooperative facility operating system.
7.2	a) a plurality of processors, each being coupled to a respective control program store and a respective data store, said plurality of processors being interconnected by a communications bus; and	See analysis of prior art for limitations 1.2 (plurality of processors), 1.8 (respective control program store), 1.9 (processor coupled to store), 4.1 (respective data store), and 1.13 (interconnected communications bus).
7.3	b) a multiple facility operating system having a kernel and providing for the message based co-operative operation of said plurality of processors, said multiple facility operating system providing for the operating system internal execution of a plurality of operating system peer-level facilities by execution of each of said peer-level facilities by a respective different one of said plurality of processors, each of said peer-level facilities constituting a respective software entity executed separately from said kernel, wherein each of said plurality of facilities implements a multi-tasking interface coupleable between said communications bus and a respective and unique peer-level control function set to permit message transfer between each of said plurality of facilities.	See analysis of prior art for limitations 1.1 (multiple facility operating system), 1.11 (kernel providing co-operative operation of a plurality of processors), 1.2, 1.3, 1.4, & 1.5 (internal execution of a plurality of operating system peer-level facilities) 1.6, 1.7, & 1.10 (execution of each facility by different processors), 1.11 (peer-level facilities executed separately from the kernel), 1.11, 1.12, & 5.1 (multi-tasking interface with unique peer-level control function set for message transfer between facilities), 1.13 (facilities coupled to a communications bus).

	'037 Claim 8	**Epoch Reference**
8.1	The computer system of claim 7 wherein said plurality of facilities includes a network facility and a filesystem utility, wherein said network facility includes a communications network peer-level control function coupled between a first multi-tasking interface and a network interface and said filesystem facility includes a data	See analysis of prior art for claim 7 and limitations 2.1 (network facility and filesystem utility), 3.1 (network facility includes communications network coupled with a multi-tasking interface to a network interface), 5.1 & 6.1 (filesystem facility includes a data storage control function coupled to a second multi-tasking interface and filesystem).

	storage peer-level control function coupled between a second multi-tasking interface and a filesystem.	

	'037 Claim 9	Epoch Reference
9.1	The computer system of claim 8 wherein said network facility is coupled through said network interface to a communications network, wherein said network facility is responsive to a predetermined network filesystem message received via said network interface to provide a predetermined filesystem message, and wherein said filesystem facility is responsive to said predetermined filesystem message to transfer data with respect to said filesystem.	See analysis of prior art for limitation 3.1.

	'037 Claim 10	Epoch Reference
10.1	The computer system of claim 9 further comprising a common data store, said network facility providing for the transfer of data between said network interface and said data store, said filesystem facility providing for the transfer of data between said data store and said filesystem, said communications network peer-level control function directing a message to said filesystem peer-level control function identifying a predetermined location of data in said data store with respect to said predetermined filesystem message.	See analysis of prior art for limitations 4.1 (common data store), 5.1 (transfer of data from network to filesystem via data store), 4.1 (message from network to filesystem identifying location of data in data store).

	'037 Claim 11	Epoch Reference
11.1	A computer system employing a multiple facility operating system to	See analysis of prior art for limitations 1.1 (multiple facility operating system architecture), 1.3 (co-

	provide for co-operative operation of a plurality of processors,	operative operation), 1.2 (plurality of processors).
11.2	wherein said operating system includes a kernel and a plurality of additional component facilities executed separately from said kernel, each of said component facilities including a facility sub-component, that defines the execution operation of a one of said component facilities, coupled to a multi-tasking interface sub-component,	See analysis of prior art for limitations 1.11 (kernel and plurality of additional component facilities executed separately from the kernel), 5.1 (facility subcomponent that defines operation of component facility coupled to a multi-tasking interface sub-component).
11.3	wherein said computer system comprises: a) a plurality of processors executing said operating system, each of said processors including local memory for the storage and execution of a respective component facility;	See analysis of prior art for limitations 1.1 (plurality of processors), 1.11 (executing said operating system), 1.6, 1.7, 1.8, 1.9, & 1.10 (processors including local memory for the storage and execution of their component facility).
11.4	b) a data memory accessible by each of said processors for the storage and retrieval of data blocks exchangeable between said processors; and	See analysis of prior art for limitation 4.1 (describing the use of shared memory).
11.5	c) a communications bus coupling said processors and said data memory to permit the exchange of control messages between said processors and data through said data memory,	See analysis of prior art for limitation 1.13 (processors coupled to a communications bus to permit exchange of control messages and data).
11.6	and wherein said processors each implement a respective different local sub-set of fewer than all of said component facilities that depends through the exchange of control messages on the execution of another sub-set of said componentized facilities by another of said processors to co-operatively implement said operating system.	See analysis of prior art for limitations 1.4 (each processor implements only a subset of the component facilities), 1.11 & 5.1 (exchange of control messages between processors to control execution of component facilities).

	'037 Claim 12	Epoch Reference
12.1	The computer system of claim 11 wherein control messages communicate any of a facility sub-component function request, a facility sub-component function response, and a facility sub-component identifier of a memory space within said data memory to use in connection with said sub-component function request.	See analysis of prior art for limitations 4.1 (message passing through shared memory) and 5.1 (use of message passing between facility sub-component functions).

	'037 Claim 13	Epoch Reference
13.1	The computer system of claim 12 wherein said plurality of component facilities includes a network facility and a filesystem facility, wherein a network facility sub-component is executed by a first processor to process network requests and data transfers and a filesystem facility sub-component is executed by a second processor to process filesystem requests and data transfers derivative of said network requests and data transfers.	See analysis of prior art for limitations 1.2 (multiple processors), 2.1 networking and filesystem processes run separately).

	'037 Claim 14	Epoch Reference
14.1	The computer system of claim 1, wherein one of the processor units in said plurality of processor units is provided further to execute a further operating system peer-level facility not in said predetermined set of operating system peer-level facilities.	See analysis of prior art for claim 1. ``The first processor, called the Applications Processor (AP), runs a somewhat modified version of 4.3BSD Unix[.] … It supports the full complement of TCP/IP networking utilities, and 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 any programs that make calls to a general purpose operating system

		(4.3BSD Unix), not just those that were predefined.

	'037 Claim 15	**Epoch Reference**
15.1	The computer system of claim 7, wherein said multiple facility operating system provides further for the operating system internal execution of a further operating system peer-level facility not in said plurality of operating system peer-level facilities, by execution of said further peer level facility by one of the processors in said plurality of processors.	See analysis of prior art for limitation 14.1.

3. <u>Cheriton 1984</u>

Cheriton 1984 discloses the V-System, a multiple facility operating system architecture designed to place as many of the usual operating system functions outside the kernel as efficiency permits. Cheriton 1984 discloses each and every limitation of—and thus anticipates—at least claims 1 and 7 of the '037 patent, as shown in the claim charts provided below.

	'037 Claim 1	**Cheriton 1984**
1.1	A computer system employing a multiple facility operating system architecture, said computer system comprising:	The V-system employs multiple computers to distribute as many of the usual operating system functions outside the kernel as efficiency permits, as disclosed in Cheriton 1984 page 20 and depicted in figure 1.
1.2	a) a plurality of processor units	Any distributed operating system architecture has a plurality of processing units, as disclosed in Cheriton 1984 page 20 and depicted in Figure 1.
1.3	provided to co-operatively execute a predetermined set of operating system peer-level facilities,	The V-System consists of a distributed kernel and a distributed set of server processes running on a plurality of processing units. This plurality of processing units interact with each other to provide a predetermined set of functions usually provided by the operating system, such as file service, print service, gateway service, routing service, etc., as

		shown in Figure 1.
1.4	wherein each said processor units is associated with a respective different one of said operating system peer-level facilities and not another of said operating system peer level facilities;	Each of the processing units in the V-System network provide one and only one function usually provided by the operating system, as shown in Figure 1 and discussed on page 20. For example, in the V-System, the file server provides only file service and not print service, while the print server provides only print service and not file storage.
1.5	and wherein each of said operating system peer-level facilities constitutes a respective separately executed software entity	Each of the processing units in the V-System network runs a program – a respective separately executed software entity – to implement its respective function, as shown in Figure 1 and discussed on page 20. For example, the file server runs a program that implements file service and only file service, while the gateway server runs a program that provides only gateway service.
1.6	which includes a respective distinct set of peer-level facility related functions, each said processor unit including:	Each of the processing units in the V-System network provides a unique set of functions, as shown in Figure 1. For example, the file server provides functions to create, read, write, and delete files while the print server provides functions to print files, provide duplexing, and select paper trays. The functions provided by these servers do not overlap and are quite distinct from each other.
1.7	i) a processor capable of executing a control program; and	Each of the processing units in the V-System network has a processor capable of executing a control program, as shown in Figure 1 and discussed on page 20. Each of the boxes is disclosed as a machine, e.g., ``file server machine" and ``gateway server machine."
1.8	ii) a memory store capable of storing said control program,	Each of the processing units in the V-System network is able to run as a standalone machine that has a memory store capable of storing the control program.
1.9	said processor being coupled to said memory store to obtain access to said control program,	The processor in the processing unit is coupled to the memory store and fetches control program instructions from that memory store.

1.10	said memory store providing for the storage of a first control program portion that includes a one of said respective distinct sets of operating system peer-level facility related functions and that corresponds to a one of said predetermined operating system peer-level facilities	Each of the processing units in the V-System network uses its memory store for storing the server program that implements its distinctive function that is usually provided by the operating system. For example, the memory store will hold the program that implements the file service and provides the create, read, write, and delete functions that are predetermined for that service.
1.11	and a second control program portion that provides for the implementation of a multi-tasking interface function,	The Court construed ``multi-tasking interface" to mean ``software that is tailored to each peer-level facility that supports direct communication with other peer-level facilities and capable of managing at least two concurrent or interleaved tasks." Each of the processing units in the V-System network, as disclosed in limitation 1.7, has a kernel – a second control program portion – that provides process and memory management, interprocess communication, and device management facilities, as disclosed on page 20. The interprocess communication functionality provides "the basic glue for connecting the different system components and is implemented with conventional 'system call' traps." The interprocess communication facility is "capable of managing at least two concurrent or interleaved tasks," given that it handles communications between multiple other system components.
1.12	said multi-tasking interface function being responsive to control messages for selecting for execution a one of said peer-level facility related functions of said one of said predetermined operating system peer-level facilities and responsive to said one of said predetermined operating system peer-level facilities for providing control messages to request or in response to the performance of said predetermined peer-level facility related functions of another operating system peer-level facility; and	Each of the kernels implements interprocess communication functionality that responds to control messages by selecting for execution the desired software functions that are implemented by a distinct software process. Each kernel is also responsive to that software process and can send control messages to request or respond to software functions implemented by a different software process. For example, the kernel disclosed in 1.7 and 1.11 will receive a control message from another peer-level facility requesting that a file be read. The kernel will select one of the file server processes to run and deliver the control message to that process. The selected file-server process will read the requested data and send a control message containing that data to the kernel. The kernel will then return the control message containing the data to the peer-level facility that requested it. The details of how the interprocess communication facility performs these tasks is detailed on pages

		20-29.
1.13	b) a communications bus that provides for the interconnection of said plurality of processor units, said communications bus transferring said control messages between the multi-tasking interface functions of said predetermined set of operating system peer-level facilities.	The Court construed ``bus" to mean ``a set of signal lines to which two or more devices may be connected and over which information is transferred between those devices." Ethernet fulfills this definition. Each of the kernels is connected to a communications bus (typically Ethernet) that provides for the interconnection of the plurality of machines, as shown in Figure 1. This communications bus (Ethernet) transfers control messages between the interprocess communication facilities – the multi-tasking interface functions – resident on each of the processor units that implement their own distinctive function that is usually provided by the operating system. This functionality is disclosed on page 19: "the V kernel was thought of as a software chassis . . . and Similar[] [to a backplane], with the V kernel interprocess communication, interface procedures provide a 'clean' interface to applications."

	'037 Claim 7	**Cheriton 1984**
7.1	A computer system implementing a co-operative facility based operating system architecture, said computer system comprising:	See the analysis of prior art for limitation 1.1 (multiple facility operating system architecture).
7.2	a) a plurality of processors, each being coupled to a respective control program store and a respective data store, said plurality of processors being interconnected by a communications bus; and	See the analysis of prior art for limitations 1.2 (plurality of processors), 1.8 (respective control program store), 1.9 (processor coupled to store), 4.1 (respective data store), and 1.13 (interconnected communications bus).
7.3	b) a multiple facility operating system having a kernel and providing for the message based co-operative operation of said plurality of processors, said multiple facility operating system providing for the operating system internal execution of a plurality of operating system peer-level facilities by execution of each of said peer-level facilities by a	See the analysis of prior art for limitations 1.1 (multiple facility operating system), 1.11 (message-based kernel providing co-operative operation of a plurality of processors), 1.2, 1.3, 1.4, & 1.5 (internal execution of a plurality of operating system peer-level facilities) 1.6, 1.7, & 1.10 (execution of each facility by different processors), 1.11 (peer-level facilities executed separately from the kernel), 1.11, 1.12, & 5.1 (multi-tasking interface with unique peer-level control function

	respective different one of said plurality of processors, each of said peer-level facilities constituting a respective software entity executed separately from said kernel, wherein each of said plurality of facilities implements a multi-tasking interface coupleable between said communications bus and a respective and unique peer-level control function set to permit message transfer between each of said plurality of facilities.	set for message transfer between facilities), and 1.13 (facilities coupled to a communications bus).

D. The Asserted Claims Lack Adequate Support Based On The Disclosure Provided In The Patent Specification

1. <u>'366 and '918 Patents</u>

The specification of the '366 patent does not disclose the invention of claim 1 of the '366 patent (and its dependent claims) as asserted by NetApp. Claim 1 recites an apparatus having a first and second processing unit each coupled to the network. In its infringement contentions, NetApp asserts that BlueArc's System Management Unit, which is directly connected to the network, is the first processing unit required by claim 1. Thus, based on NetApp's contentions, claim 1 would cover apparatuses where the first processor unit is a stand-alone unit that is connected to the network directly, without regard to the second processing unit.

But the specification does not disclose a first processing unit directly connected to the network. The only item disclosed in the patent corresponding to a first processing unit is the local host. As seen in Figure 2, the local host is connected via a VME bus to the network controller; it does not operate on its own on the network. Indeed, having a local host connected via the network controller is key to the purported invention. The invention is premised upon a division of labor, wherein the first processor handles a predefined set of non-NFS requests, while the second processor handles NFS requests. The specification discloses only one mechanism for distinguishing between a request falling within the predefined set of non-NFS requests and a NFS request and routing the request appropriately – the network controller.

An architecture that relies upon a first processing unit that is completely independent from the second processing unit, and is a separate device on the network, is quite different in structure and design from the architecture disclosed in the '366 patent.

2. '037 Patent

Claim 14 is irreconcilable with the other claims in the '037 patent. Claim 14 depends on claim 1, which requires a "plurality of processor units" wherein each processing unit is associated with a different operating system function and not another. As discussed above, this means that the file server provides the file service function but not print service. Yet, in claim 14, one of the processor units that is associated with a different operating system function and not another somehow also executes another operating system function. This language contradicts itself, and as such it is practically impossible to discern what is the invention recited in claim 14.

Likewise, claim 15 depends on claim 7, which requires each of "said plurality of operating system peer-level facilities" to be executed by "a respective different one of said plurality of processors," where each such facility is a "respective software entity executed separately from said kernel" and is an "independently, i.e., separately executed, software entity." As discussed above, this means that the file server provides the file service function but not print service. Yet, in claim 15, one of those "said plurality of processors" that separately executes its particular operating system function somehow also executes to an additional operating system function not in "said plurality of operating system peer-level facilities." Because this language contradicts itself, it is practically impossible to discern what is the invention recited in claim 15.

Dated: March 4, 2005

By: [signature]
MARSHALL KIRK MCKUSICK

EXHIBIT A

THE CHATHAM GROUP
101 First Street, Suite 477, Los Altos, CA 94022
Phone 650-948-1243, Fax 650-949-0534
www.chathamlaw.com

MARSHALL KIRK MCKUSICK, Ph. D.

EDUCATION

Ph.D., Computer Science, U.C. Berkeley, December 1984
M.S., Computer Science, U.C. Berkeley, December 1980
M.S., Business Administration, U.C. Berkeley, December 1979
B.S., Electrical Engineering (with distinction), Cornell University, June 1976

WORK EXPERIENCE

1982 - present
Consulting and design review on application development, porting, and performance enhancement for UNIX-based systems.

1983 - present
Teaching tutorials on UNIX kernel internals and system administration for The Usenix Association, several University of California Extension locations, and on-site at numerous companies throughout the world.

1993 - present
Expert witness on software patent, trade secret, and copyright issues particularly those related to operating systems and filesystems.

1984 - 1993
Research Computer Scientist in charge of the Computer Systems Research Group with responsibility for research and development of the 4.3 and 4.4 Berkeley Software Distributions.

Summer 1981
Computer Systems Research Group - Implementation of a fast filesystem for UNIX.

Winter 1980
U.C. Berkeley Teaching Assistant - First year graduate course on Compiler Construction.

Summers 1977-1979
Hughes Aircraft Company - Design, implementation, and integration of a microprocessor based peripheral interface card for an airborne radar system.

AWARDS

1992 - Usenix Association Lifetime Achievement Award
1991 - UNIX International, Inc. Industry Award

1977-1980 - Howard Hughes Fellowship
1976 - U.C. Berkeley Fellowship

PROFESSIONAL SOCIETIES AND RESPONSIBILITIES

- Member of the Board of Directors of Usenix Association (1986 - 1992, 2000 - present)
- Member of the Editorial Board of ACM Queue (2003 - present)
- President of the Board of Directors of Usenix Association (1990 - 1992, 2002 - 2004)
- Chairman of the Board of Directors iXsystems Inc. (2001 - 2002)
- Member of the Board of Directors iXsystems Inc. (2001 - 2002)
- Chairman of the Board of Directors Berkeley Software Design (1998 - 2001)
- Member of the Board of Directors Berkeley Software Design (1991 - 2001)
- Member of the Editorial Board of Usenix Computing Systems (1987 - 1996)
- Member of the Editorial Board of UNIX Reviews (1985 - 1990, 1992 - 1996)
- IEEE POSIX Standardization Committee (1986 - 1998)
- ACM - Special Interest Group on Operating Systems
- IEEE - Computer Society

PUBLICATIONS

M. McKusick ``Running Fsck in the Background", Proceedings of the BSDCon 2002 Conference, pp 55-64, February 2002.

M. Seltzer, G. Ganger, M. McKusick, K. Smith, C. Soules, C. Stein, ``Journaling versus Soft Updates: Asynchronous Meta-data Protection in File Systems", Proceedings of the San Diego Usenix Conference, pp 71-84, June 2000.

G. Ganger, M. McKusick, C. Soules, Y. Patt, ``Soft Updates: A Solution to the Metadata Update problem in File Systems", ACM Transactions on Computer Systems 18, 2. pp 127-153, May 2000.

M. McKusick, G. Ganger, ``Soft Updates: A Technique for Eliminating Most Synchronous Writes in the Fast Filesystem", Proceedings of the Freenix Track at the 1999 Usenix Annual Technical Conference, pp 1-17, June 1999.

K. McKusick, Chapter 2 - Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable in Open Sources: Voices from the Open Source Revolution, Editors: C. DiBona, S. Ockman, and M. Stone, O'Reilly & Associates, Inc, 101 Morris Street, Sebastopol, CA 95472 ISBN 1-56592-582-3 pp 31-46, 1999.

M. McKusick, Chapter 81 - Secondary Storage and Filesystems, The Computer Science and Engineering Handbook, CRC Press, Boca Raton, FL. ISBN 0-8493-2909-4. pp 1761-1775, 1997.

M. McKusick, K. Bostic, M. Karels, J. Quarterman, The Design and Implementation of the 4.4BSD Operating System Addison-Wesley Publishing Company, Reading, MA. April 1996, ISBN 0-201-54979-4. http://www.awprofessional.com/titles/0-201-54979-4 French Translation published 1997, International Thomson Publishing, Paris, France, ISBN 2-84180-142-X.

M. McKusick ``Secondary Storage and Filesystems", ACM Computing Surveys 28, 1. pp 217-219, March 1996.

J. Pendry, M. McKusick, ``Union Mounts in 4.4BSDLite", Proceedings of the New Orleans Usenix Conference, pp 25-33, January 1995.

M. McKusick, K. Bostic, ``4.4 Berkeley Software Distribution", Five volumes of documentation and software, comprised of: User's Reference Manual, ISBN 1-56592-075-9, User's Supplementary Documents, ISBN 1-56592-076-7, Programmers's Reference Manual, ISBN 1-56592-078-3, Programmers's Supplementary Documents, ISBN 1-56592-079-1, System Manager's Manual, ISBN 1-56592-080-5, CD-ROM Companion, ISBN 1-56592-081-3 (Domestic), ISBN 1-56592-092-9 (International). O'Reilly & Associates, Inc. 103 Morris Street, Suite A Sebastopol, CA 94572, April 1994.

M. McKusick, ``Networking Comes to UNIX", UNIX Review 12, 5. pp 40, May 1994.

M. McKusick, ``What's New in 4.4BSD", UNIX Review 12, 1. pp 51-56, January 1994.

M. McKusick, ``Le prospettive di UNIX su PC", Informatica Oggi and UNIX 13, 90. pp 34-35, Gennaio, Italy 1993.

M. Seltzer, K. Bostic, M. McKusick, C. Staelin, ``An Implementation of a Log-Structured File System for UNIX", Proceedings of the San Diego Usenix Conference, pp 201-218, January 1993.

M. McKusick, ``UNIX Variants: BSD", UNIX Review 10, 4. pp 46-47, April 1992.

S. Leffler, M. McKusick, The Design and Implementation of the 4.3BSD UNIX Operating System Answer Book Addison-Wesley Publishing Company, Reading, MA. April 1991, ISBN 0-201-54629-9. Japanese Translation published January 1992, ISBN 4-8081-8039-5.

M. McKusick, M. Karels, K. Sklower, K. Fall, M. Teitelbaum, K. Bostic, ``Looking Ahead to 4.4BSD", Bit Magazine, pp 10-19, September 1990.

M. McKusick, M. Karels, K. Bostic, ``A Retrospective of 4.3BSD", Bit Magazine, pp 4-9, September 1990.

M. McKusick, M. Karels, K. Bostic, ``A Pageable Memory-Based Filesystem", Proceedings of the Anaheim Usenix Conference, pp 137-144, June 1990. Reprinted in Proceedings of the United Kingdom UNIX Users Group Meeting, London, England, pp 109-115, July 1990.

M. McKusick, M. Karels, K. Sklower, K. Fall, M. Teitelbaum, K. Bostic, ``Current Research by the Computer Systems Research Group of Berkeley", Proceedings of the European UNIX Users Group Meeting, Brussels, Belgium, April 1989.

M. McKusick, M. Karels, K. Bostic, ``The Release Engineering of 4.3BSD", Proceedings of the New Orleans Usenix Workshop on Software Management, pp 95-100, April 1989.

S. Leffler, M. McKusick, M. Karels, J. Quarterman, The Design and Implementation of the 4.3BSD UNIX Operating System Addison-Wesley Publishing Company, Reading, MA January 1989, ISBN 0-201-06196-1. German Translation published June 1990, ISBN 3-89319-239-5. Japanese Translation published June 1991, ISBN 4-621-03607-6.

M. McKusick, M. Karels, ``Design of a General Purpose Memory Allocator for the 4.3BSD UNIX Kernel", Proceedings of the San Francisco Usenix Conference, pp 295-304, June 1988.

M. McKusick, M. Karels, ``Directions of UNIX at Berkeley", Digest of Papers of the Thirty-second IEEE Computer Society International Conference, Compcon, San Francisco, pp. 196-199, February 23-27, 1987.

M. McKusick, M. Karels, ``A New Virtual Memory Implementation for Berkeley UNIX", Proceedings of the European UNIX Users Group Meeting, Manchester, England, pp. 451-460, September 1986.

M. Karels, M. McKusick, ``Towards a Compatible Filesystem Interface", Proceedings of the European UNIX Users Group Meeting, Manchester, England, pp. 481-496, September 1986.

M. Karels, M. McKusick, ``Network Performance and Management with 4.3BSD and IP/TCP", Proceedings of the Atlanta Usenix Conference, pp 182-188, June 1986.

M. McKusick, M. Karels, J. Bloom, ``4.3 Berkeley Software Distribution", Six volumes of documentation and tape of software, Computer Systems Research Group, 463 Evans Hall, Dept of EECS, Berkeley, CA 94720 April 1986.

M. McKusick, ``UNIX Unleashed", UNIX Reviews 3, 10. pp 28-33, October 1985.

M. McKusick, M. Karels, S.Leffler, ``Performance Improvements and Functional Enhancements in 4.3BSD" Proceedings of the Portland Usenix Conference, pp 519-531, June 1985.

M. McKusick, ``A Berkeley Odyssey: Ten Years of BSD History", UNIX Reviews 3, 1. pp 30-42, January 1985.

M. McKusick ``Register Allocation and Data Conversion in Machine Independent Code Generators", Ph.D. Dissertation, University of California, Berkeley. November 1984.

M. McKusick, W. Joy, S. Leffler, R. Fabry, ``A Fast File System for UNIX", ACM Transactions on Computer Systems 2, 3. pp 181-197, August 1984.

S. Leffler, M. Karels, M. McKusick, ``Measuring and Improving the Performance of 4.2BSD", Proceedings of the Salt Lake City Usenix Conference, pp 237-252, June 1984.

P. Aigrain, S. Graham, R. Henry, M. McKusick, E. Pelegri-Llopart, ``Experience with a Graham-Glanville Style Code Generator", Proceedings of the ACM SIGPLAN 1984 Symposium on Compiler Construction, SIGPLAN Notices 19, 6, Montreal, Quebec. pp. 13-24. June 1984.

M. McKusick, ``Using gprof to Tune the 4.2BSD Kernel", Proceedings of the European UNIX Users Group Meeting, Nijmegen, Netherlands. April 1984.

M. McKusick, T. Kowalski, ``Fsck - The UNIX File System Check Program", 4.4BSD System Manager's Manual, Document #3, O'Reilly & Associates Inc, Sebastopol, CA, April 1994, ISBN1-56592-080-5.

W. Joy, E. Cooper, R. Fabry, S. Leffler, M. McKusick, D. Mosher, ``4.2BSD System Manual", 4.2BSD UNIX Programmer's Manual, Vol 2c, Document #68 August 1983.

B. Kridle, M. McKusick, ``Performance Effects of Disk Subsystem Choices for VAX Systems Running 4.2BSD UNIX", Proceedings of the Toronto Usenix Conference, July 1983.

S. Graham, P. Kessler, M. McKusick, ``An Execution Profiler for Modular Programs", Software - Practice and Experience, Vol 13. pp 671-685, 1983.

S. Graham, P. Kessler, M. McKusick, ``gprof: A Call Graph Execution Profiler", Proceedings of the ACM SIGPLAN 1982 Symposium on Compiler Construction, SIGPLAN Notices 17, 6, Boston, MA. June 1982.

W. Joy, S. Graham, C. Haley, M. McKusick, P. Kessler, ``Berkeley Pascal User's Manual", UNIX Programmer's Manual, Vol 2c, Document #48 October 1981.

W. Joy, M. McKusick, ``Berkeley Pascal PX Implementation Notes, Version 2.0", Computer Systems Research Group, U.C. Berkeley. January 1979.

KEKER & VAN NEST, LLP
ROBERT A. VAN NEST - #84065
BRIAN L. FERRALL - #160847
LEO L. LAM - #181861
RAVIND S. GREWAL - #220543
710 Sansome Street
San Francisco, CA 94111-1704
Telephone: (415) 391-5400
Facsimile: (415) 397-7188

BROMBERG & SUNSTEIN LLP
LEE CARL BROMBERG (Admitted *Pro Hac Vice*)
125 Summer Street
Boston, MA 02110-1618
Telephone: (617) 443-9292
Facsimile: (617) 443-0004

Attorneys for Defendant
BLUEARC CORPORATION

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

NETWORK APPLIANCE, INC., Plaintiff, v. BLUEARC CORPORATION, Defendant.	Case No. C 03-05665 MHP **PROOF OF SERVICE** Date: September 30, 2004 Time: 9:30 a.m. Courtroom: 15, 18th Floor Judge: Hon. Marilyn Hall Patel

I am employed in the City and County of San Francisco, State of California in the office of a member of the bar of this court at whose direction the following service was made. I am over the age of eighteen years and not a party to the within action. My business address is Keker & Van Nest, LLP, 710 Sansome Street, San Francisco, California 94111.

On March 4, 2005, I served the following document:

EXPERT REPORT OF
MARSHALL KIRK MCKUSICK, Ph.D.

☑ by e-mail in pdf format.

☑ by regular **UNITED STATES MAIL** by placing a true and correct copy in a sealed envelope addressed as shown below. I am readily familiar with the practice of Keker & Van Nest, LLP for collection and processing of correspondence for mailing. According to that practice, items are deposited with the United States Postal Service at San Francisco, California on that same day with postage thereon fully prepaid. I am aware that, on motion of the party served, service is presumed invalid if the postal cancellation date or the postage meter date is more than one day after the date of deposit for mailing stated in this affidavit.

Scott Wales
Howrey Simon Arnold & White, L.L.P.
525 Market Street, Suite 3600
San Francisco, CA 94105
bunsowh@howrey.com
waless@howrey.com

Executed on March 4, 2005, at San Francisco, California.

I, Ryan M. Kent, declare under penalty of perjury under the laws of the State of California that the above is true and correct.

RYAN M. KENT