United States District Court,
N.D. California.

QUANTUM CORPORATION,
Plaintiff.
v.
RIVERBED TECHNOLOGY, INC,
Defendant.
and Related Counterclaim,
and Related Counterclaims.

No. C 07-04161 WHA

Aug. 5, 2008.

Amardeep Lal Thakur, Sheppard Mullin Richter & Hampton, San Diego, CA, Steven E. Bledsoe, Arent Fox LLP, Los Angeles, CA, Nathaniel Philip Bruno, Sheppard Mullin Richter & Hampton LLP, San Francisco, CA, for Plaintiff.

CLAIM CONSTRUCTION ORDER

WILLIAM ALSUP**, District Judge.**

INTRODUCTION

This is a claim construction order for United States Patent Nos. 7,116,249 B2 and 5,990,810. This order addresses five phrases selected for construction by the parties. A technology tutorial, as well as a full round of briefing and a hearing, preceded this order.

STATEMENT

Each side is accusing the other of infringement. Plaintiff Quantum Corp., purports to be the exclusive licensee of the '810 patent. Defendant Riverbed Technology, Inc., alleges it is the exclusive owner of the '249 patent. Each patent generally relates to data storage. More specifically, each patent is aimed at techniques for more efficiently storing data.

* * *

The '249 patent is directed to a method of storing a data file using data-compression techniques. As described in the '249 specification (col.1:42-51):

Data compression is a process of representing input data as compressed data such that the compressed data comprises fewer bits or symbols than the input data and is such that the compressed data can be decompressed into at least a suitable approximation of the original data input. Compression allows for more

efficient transmission of data, as fewer bits need to be sent to allow a receiver to recover the original set of bits (exactly or approximately) and compression allows for more efficient storage as fewer bits need to be stored.

According to the purported invention, data is compressed using a process called "segmentation." *First,* a data file is broken into distinct segments- *i.e.,* strings or sequences of data. *Second,* the segments are each assigned reference labels. *Third,* the reference labels are compared to a database containing all previously stored reference labels from other segments. If the reference label is found in the database then it can be used to replace multiple occurrences of the common segment. If the reference label is not found in the database then it is added to the database for future comparisons. By using a single reference label for several segments that are common, less data needs to be stored.

A simple example illustrates the general process. Assume a word file contains the following text:

I plan to go hiking at Yosemite National Park next week. Yosemite National Park is one of the most beautiful locations in the United States. In fact, many people consider Yosemite National Park one of the most beautiful locations in the world.

If the text "Yosemite National Park" were identified as a segment and assigned a reference label, such as "XYZ", the sentence could be represented as follows:

I plan to go hiking at XYZ next week. XYZ is one of the most beautiful locations in the United States. In fact, many people consider XYZ one of the most beautiful locations in the world.

By replacing the longer text string "Yosemite National Park" with the shorter reference "XYZ", therefore, the word file contains less data and requires less space for storage.

A schematic of the compression process for the invention is shown below.

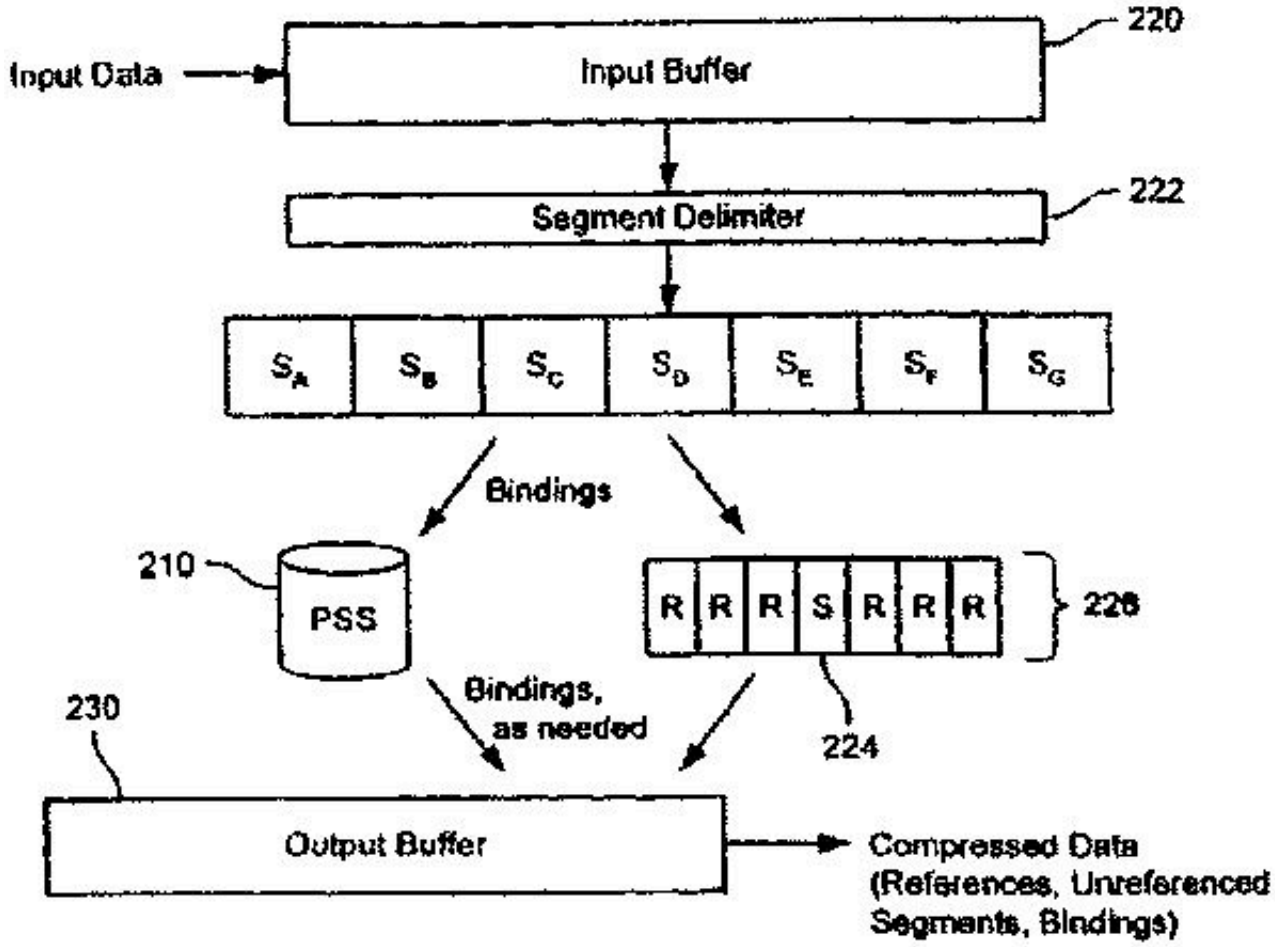

FIG. 3

Input data is buffered at block 220 and then broken up into several segments Sa, Sb, ... Sg by segment delimiter 222. Reference labels 226 are then assigned to the segments and thereafter compared to previously stored reference labels in persistent segment store 210. If a comparable segment is present, then its previously defined reference is used to replace the segment currently being encoded. If the segment is not present in the PSS, then the new reference label is added to the PSS. The data, including both referenced and unreferenced segments, is then outputted through output buffer 230 as compressed data.

* * *

The '810 patent similarly describes a method for identifying and removing duplicative data to save space for storage and transmission. As described in the '810 specification, prior-art systems that identified duplicative data being sent or stored were common. In these systems, data would be broken up into blocks and compared to one another. Where two blocks matched, the data in the block could be replaced with a pointer that directed the system to a common-pool bank where the data was centrally stored. But, those prior-art systems all used fixed-length windows to break the data into blocks. The problem with that process was it would only recognize duplicative data if the data within the two fixed blocks being compared aligned perfectly. In practice, however, this was not always the case. The problem was illustrated in Figure 1 of the patent.

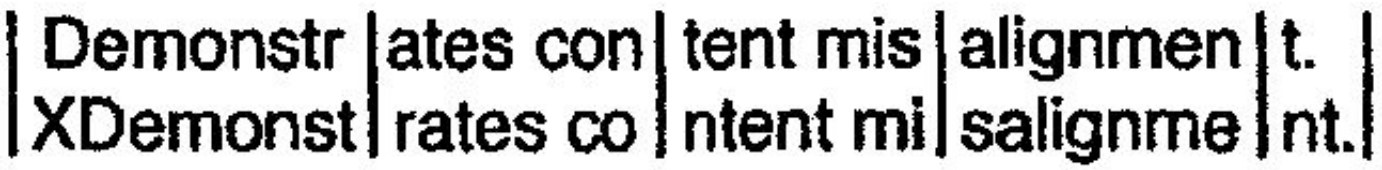

Figure 1

As shown, the only difference between the sentence on the top line and the sentence on the bottom line is one character. But after the sentences are broken up into blocks, it appears as though none of the blocks between the two sentences match.

The '810 purportedly solved this problem by employing a window of varying length to slide across the data being stored or transmitted one position at a time. At each new position the window being analyzed would be reduced to a value, or a checksum. The window would then be compared to a predetermined value in a preexisting algorithm. If the conditions of the test were satisfied, a block at the window being tested would be created.

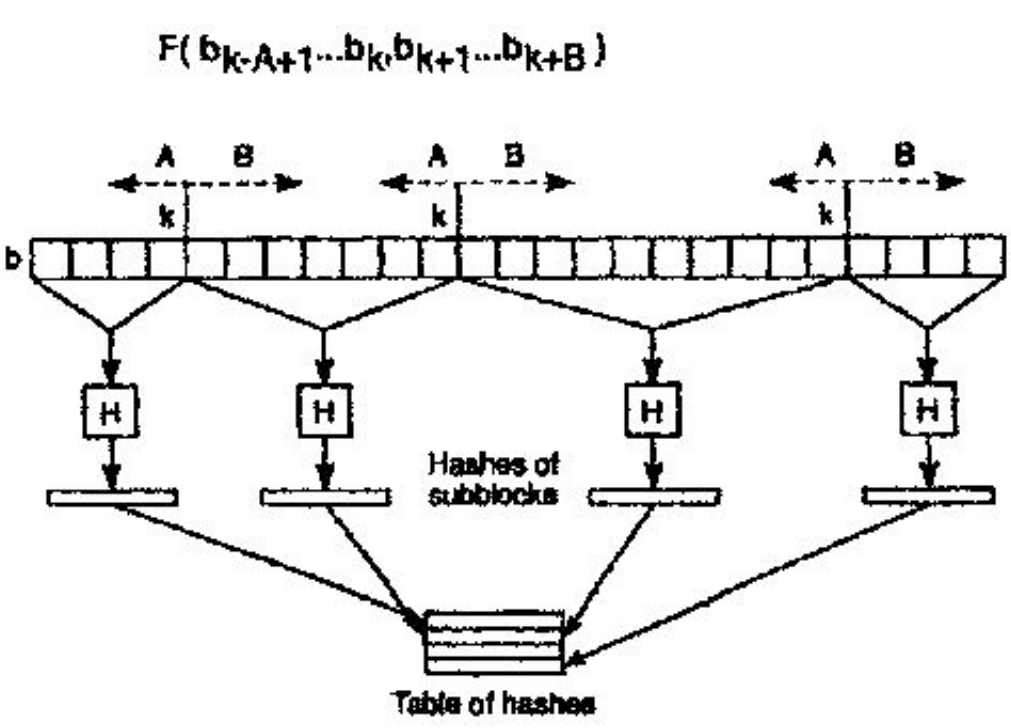

Figure 24

To illustrate with Figure 24 above, a window of length A + B slides across the data set. When a given window meets the predetermined condition, a partition in the data set is created between positions k and k+1. As described by the inventor, Dr. Ross Williams, before the patent application was filed (Bruno Decl. Exh. B):

The important thing is not to define the blocks on fixed boundaries (*e.g.,* every 1024 bytes). By defining the boundaries at arbitrary points based on the data (*e.g.,* where a checksum takes a particular value), common subsections of the old and new version of the file can be detected, even though bytes have been inserted, deleted, or the contents of the file have been rearranged.

By using a window of varying length, therefore, the system could more accurately identity duplicative data. Figure 3 shows the improvement.

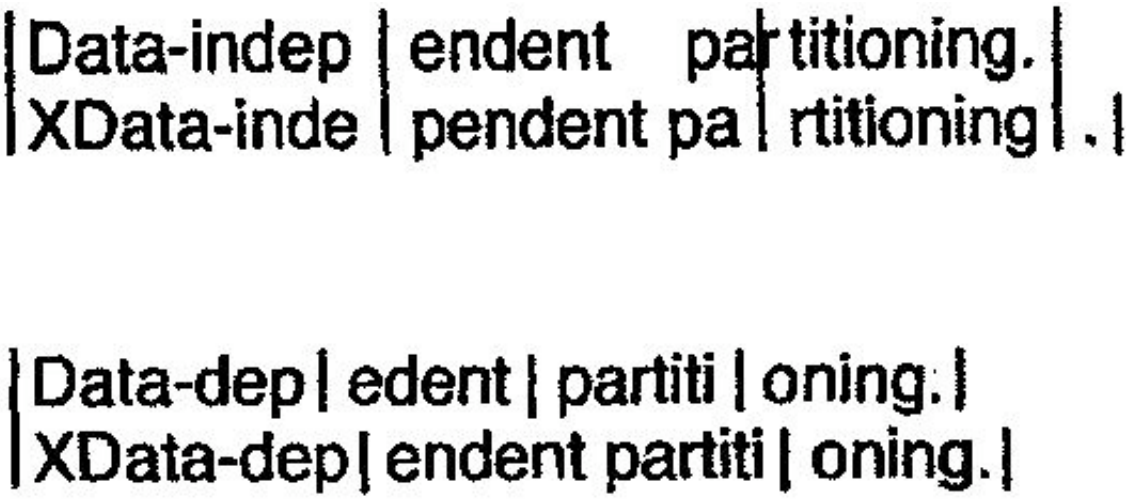

Figure 3

* * *

This action was filed on November 13, 2008, alleging Quantum infringed the '249 patent. Riverbed counterclaimed alleging infringement of the ' 810 patent. The parties jointly selected five phrases for construction at this time. Two phrases appear in the '249 patent and three phrases appear in the '810 patent. A technology tutorial on both patents was held on July 2, 2008, and a hearing was held on July 24, 2008. Trial is set for March 30, 2009.

ANALYSIS

1. LEGAL STANDARD.

Claim construction is a matter of law to be decided by a judge, not a jury. Markman v. Westview Instruments, Inc., 517 U.S. 370, 388, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996). Courts must give words in the claims their ordinary and customary meaning, which "is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention." Phillips v. AWH Corp., 415 F.3d 1303, 1312-13 (Fed.Cir.2005) (en banc).

Where this ordinary and customary meaning is not immediately clear, courts must primarily look to intrinsic evidence (*i.e.,* the claims, the specification, and the prosecution history) to determine the meaning. Id. at 1314. With respect to the specification, although a difficult task, a court must distinguish "between using the specification to interpret the meaning of a claim and importing limitations from the specification into the claim." Id. at 1323. The latter is not permissible.

Although courts have the discretion to consider extrinsic evidence, including expert and inventor testimony, dictionaries and scientific treatises, such evidence is "less significant than the intrinsic record in determining

the legally operative meaning of claim language." Id. at 1317 (citation omitted). "The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." Id. at 1315. "Nonetheless, any articulated definition of a claim term ultimately must relate to the infringement questions it was intended to answer." E-Pass Tech., Inc. v. 3Com Corp., 473 F.3d 1213, 1219 (Fed.Cir.2007) (citing Wilson Sporting Goods Co. v. Hillerich & Bradsby Co., 442 F.3d 1322, 1326 (Fed.Cir.2006)).

2. RIVERBED'S '249 PATENT.

The two phrases in the '249 patent selected for construction are: (A) "reference label" and (B) "back-end storage system."

A. "Reference Label."

The disputed term "reference label" was recited throughout the claims of the ' 249 patent. Claim 1, for example, stated: (col.16:50-17:19):

1. A method for storing data, comprising:

receiving a file from a client at a server;

segmenting the file into one or more segments;

forming a list, comprising:

determining whether each segment is present in a segment store, wherein a segment that is present in the segment store has an assigned *reference label;*

for each segment in the segment store, adding to the list the assigned *reference label;* and

for each segment not present in the segment store, assigning a *reference label* to the segment, storing the segment and the *reference label,* and adding the *reference label* to the list; and

storing an association between the file and the list.

Riverbed proposes that "reference label" should mean "data used to identify a segment." Quantum argues the term should mean "globally unique data used to identify a segment." The parties therefore agree that a reference label is data used to identify a segment. The disagreement is over whether the data must be "globally unique."

Both proposed constructions must be rejected. Quantum's construction improperly imports a limitation from the specification. Significantly, the term "globally unique" only appears once in the '249 specification when describing a specific embodiment of the invention. That embodiment appears below.

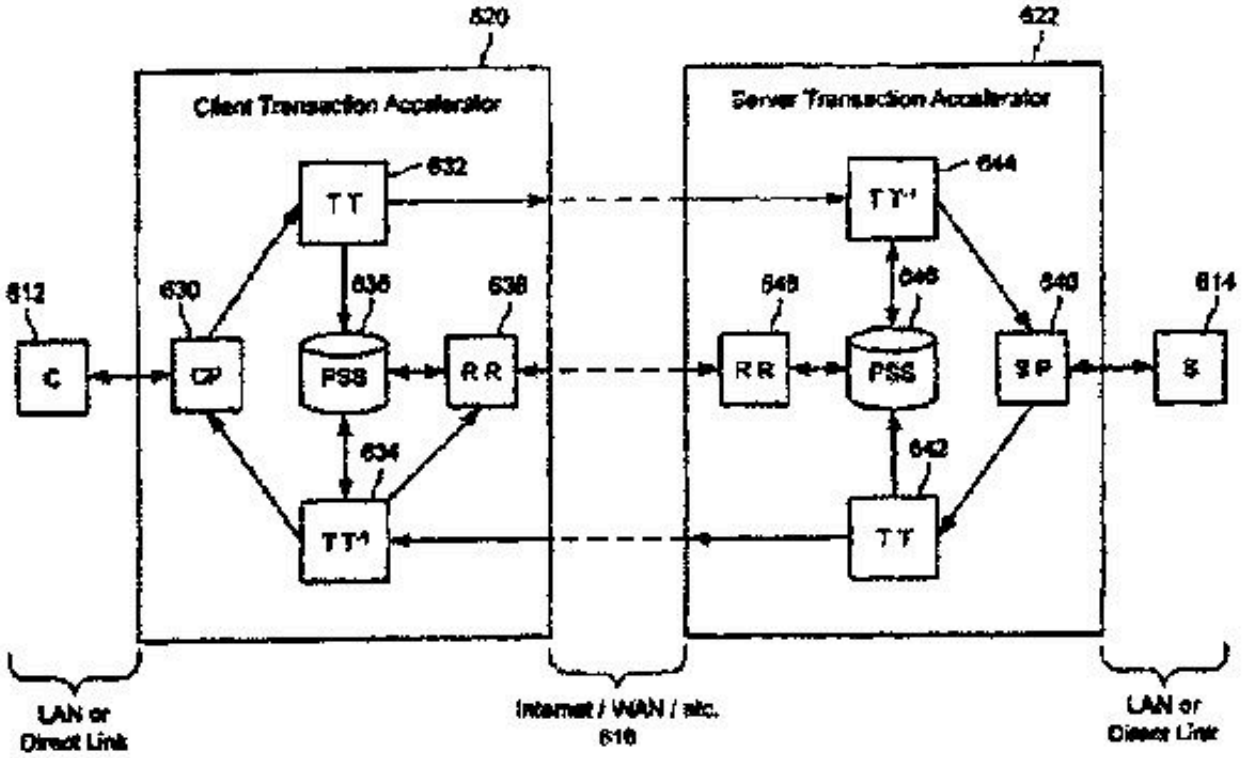

FIG. 12

In the context of this embodiment, the specification stated, "[a]s each [transaction transformer] creates bindings, it assigns globally unique reference labels so that a recipient of a reference label will always be able to replace it with unambiguous segment data" (col.15:19-21). Unlike the other embodiments disclosed in the '249 specification, the embodiment shown in Fig. 12 has two separate transaction transformers (the Client Transaction Accelerator 620 and the Server Transaction Accelerator 622) that could each separately assign reference labels to segmented data. In light of this fact, it makes perfect sense that segmented data would need to be assigned a "globally unique" reference label because if the client transaction accelerator assigned a reference label to a data segment, the server transaction accelerator should not be able to assign that same reference label to different data. In other words, the reference labels assigned must be global- *i.e.*, both transaction transformers must be consistent in their usage. The remaining embodiments of the '249 patent, however, do not contain two structures that can independently assign reference labels to segmented data. They only have one. It would thus be unnecessary to describe a reference label as "global" to the system. Accordingly, Quantum's construction must be rejected because it reads in a limitation only required by one specific embodiment of the invention. *See* SuperGuide Corp. v. DirectTV Enterprises, Inc., 358 F.3d 870, 875 (Fed.Cir.2004) ("[A] particular embodiment appearing in the written description may not be read into a claim when the claim language is broader than the embodiment.")

Nonetheless, reference labels assigned to segments should be "unique" in a different, less demanding sense. Indeed, the purported invention behind the ' 249 patent requires reference labels to be unique. If they were not, data would be compressed with the same label attached to different segments. When the data is ultimately decoded after compression it would be a highly inaccurate representation of the original. There are multiple references of this throughout the specification. For example, the specification recited (emphasis added) (col.9:29-34):

Continuing the description of the encoding process, if an acceptable segment (usually an identical segment, but in special cases, complete identity is not required) is not present, *a new unique name* is assigned to the segment and the binding (reference label, segment data) is entered into the PSS.

Riverbed argues that a reference label cannot be 100% unique because it would be impossible to guarantee strict exclusivity at all times. As explained by Riverbed, "[t]hese rare cases are inherent in using such a scheme for compressing data" (Br.5). Notably, one of the methods disclosed in the '249 specification for assigning reference labels to segmented data is "hashing." Under this approach, data is put through a function that assigns a "unique" number to represent the data being referenced. In this way, data is "hashed" to a specific value. But the assignment process is not perfect. Sometimes two different data segments inadvertently get assigned the same number- *i.e.,* they both hash to the same value.

This concept was further eluded to at the deposition of Michael Demmer, one of the inventors listed on the '249 patent (Demmer Dep. 45:18-46:7):

Q. Okay. Would it be accurate to say that where the hash values ... were the same, that would give you a pretty good idea that data would be the same, or a certain probability?

A. Generally, the properties of ... what's known as a good hash function are that conclusions can be held; that there's a high probability that if two hash values are the same, that the originated data was the same.

Q. But it's not a hundred percent probability?

A. It cannot be a hundred percent probability by the laws of information theory.

Because data is being processed in a mathematical system in which there are a discrete number of possibilities to represent data there will always be some inaccuracy when assigning reference labels to segments. U.S. Patent No. 7,120,666, which is incorporated by reference in the '249 specification, stated (col.14:42-62) (emphasis added):

Several schemes are usable to ensure that each named segment has a unique name throughout the system at any given time (*i.e.,* that no two segments with different data are erroneously assigned the same name). In one approach, every segment reference is generated as a large random number, where the number of unique references is much lower than the space of all possible large random numbers. *This scheme is less than desirable because a small possibility exists that two segments will have the same segment reference but different segment data, which would case a receiving [transaction accelerator] to erroneously reconstruct a message with the wrong data.*

Another scheme is to generate hashes from the segment data so that each segment reference is a hash of the segment data and different segment data will result in different hashes, except in very rare cases. *Yet again, the rare case will always be problematic, as long as two degenerate segments with the same reference but different segment data exist in the system.* Unlike the random number case, this problem will recur every time the particular data pattern exists in the data stream.

This order therefore does recognize that a reference label will not necessarily be 100% unique given the mathematical constraints of the system. To be clear though, the goal of the system is that each reference

label be unique in practice, *i.e.*, almost always unique accepting the small mathematical chance that two different sequences will hash to the same value and thus have the same reference label. The construction for "reference label" must thus accommodate both circumstances.

Quantum argues that the '249 specification teaches away from any practicality constraint on the system. FN1 In particular, Quantum relies on the following passage (col.9:21-28) (emphasis added):

FN1. After the claim construction hearing, Riverbed filed a motion for leave to file a statement in response to this argument on the ground that it was raised for the first time at the hearing. This order disagrees. Quantum's opposition should have easily put Riverbed on notice as to this argument.

Thus, to determine if a segment exists in the PSS, the encoder computes the hash for the segment in question and performs a lookup on the segment index using said hash. If the lookup fails, the segment cannot be in the PSS. If the lookup succeeds, the encoder *can* compare each segment returned by the lookup to the segment in question to check for an exact match. *This handles the rare case that multiple segments hash to the same hash index.*

It is not entirely clear how this passage should be interpreted in the context of the rest of the specification. Quite significantly, the passage stated that if a lookup does succeed, the system "can" compare each segment individually to check for a perfect match- *not* that it must. But it does go on to later state that "this handles" (present tense) the rare case as opposed to "could handle" (conditional sense). The '666 specification though does clearly disclose that hash functions can be used that yield imperfect references. In light of the ambiguity in the passage and the broad claim language used, this order finds that "reference label" covers both those circumstances where a reference label is not perfectly unique as well as when it is perfectly unique. It cannot be overlooked that the claim language itself only calls for a "reference label." The claim drafter could have easily chosen to include some type of modifier to the claim language to indicate that the scope of the invention only included reference labels that were perfectly unique. This was not done. No express limit was placed to surrender that subject matter.

Accordingly, this order holds that "reference label" means: "Unique data used to identify a segment. Unique, however, does not mean that each assigned data is absolutely one of a kind."

B. "Back-End Storage System."

The term "back-end storage system" appears in independent claims 19 and 25. Claim 19 recited (col.19:27-36):

19. A system for storing files, comprising:

a front-end file system for receiving from a client a file command for a file, where the front-end file system includes a front-end file server;

a *back-end storage system* including logic to segment the file; and a segment store for storing the segments;

a network file system interface for sending or receiving contents of the file between the front-end file system and the *back-end storage system.*

Quantum contends that this term should mean "a non-local, HCS-enabled *storage system* that accepts commands from a front-end file server to store and retrieves [sic] whole files." HCS is an acronym standing for "hierarchical content-induced segmentation." Riverbed argues the term should be given its plain and ordinary meaning. Because Quantum's construction includes "storage system," it would appear as if the parties agree that "storage system" should be given its ordinary and plain meaning.

Other than in the claims, the term "back-end storage system" appears nowhere in the '249 patent. Nor does it appear in the prosecution history. Not even the term "back-end" is ever stated. But Figure 14, shown below, does illustrate an exemplary system including both a "front" and a "back" system.

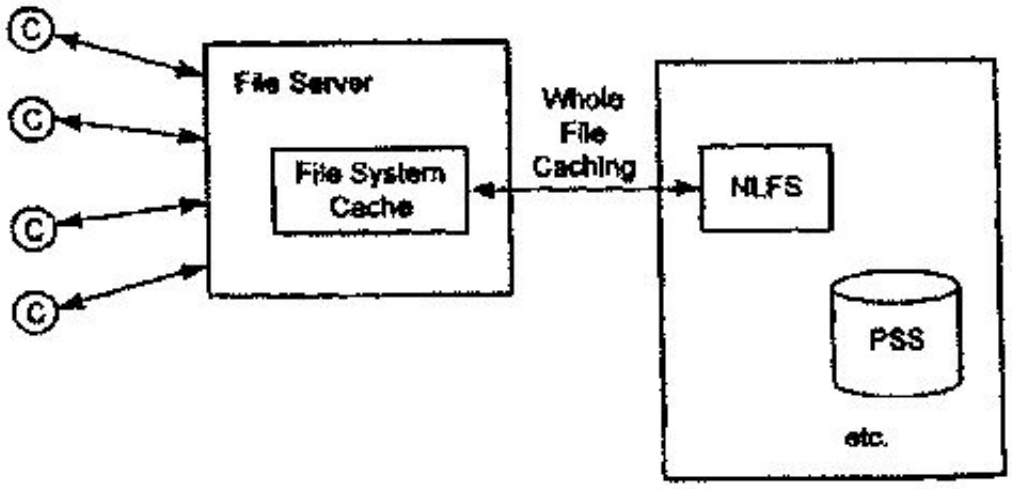

FIG. 14

As shown in the figure, clients (Cs) are connected to a front-end file system, which includes a file server and a file system cache. The front-end system is then connected to the back-end system, which contains a near-line file system ("NLFS") and the PSS. The NLFS is the component that segments and references the data being stored.

This order rejects Quantum's complex and unduly detailed construction. None of the limitations Quantum seeks to import into its construction are required by the claims or the specification. *First,* the term "non-local" does even appear in the specification. Notably, the term "local" is defined (col.4:48-54), but is not used when explaining the embodiment shown in Figure 14 above. Besides a few generally broad quotes indicating that the back-end system is distinct from the front-end system, Riverbed cites to no support for its construction on this specific point. *Second,* although an embodiment discussed in the '249 specification is "HCS-enabled," nothing in the claims or specification requires it to be. To the contrary, the specification stated, "*[a] better approach* is to apply HCS to a file system that employs whole-file accesses rather than block-based accesses" (emphasis added) (col.15:44-49). The specification further recited, "[i]n another embodiment of the invention, HCS *can* be used as a complementary component of a more standard, operating system-based file system by combining the NLFS described above with a file system front-end" (emphasis added) (col.16:37-40). *Third,* requiring the back-end system to "store and retrieve *whole files*" is

only preferred. Again, the specification stated, "[a] better approach is to apply HCS to a file system that employs whole-file accesses rather than block-based accesses" (col.15:44-49). This language makes clear that a whole-file system was thought to be a superior approach to implementing the system-not that it was the only approach. Indeed, the claim language itself recited that the interface between the front and back end be capable of "sending or receiving *contents of the file*."

The term "backend," however, is not so common that it requires no construction at all. The Institute of Electrical and Electronics Engineers, an established authority in the art, defines "backend" as "pertaining to one part of a process which has two parts, the frontend and the backend; the frontend usually denotes what the user sees and the backend denotes some special process" (Briggs Reply Decl. Exh. D). Accordingly, this order finds the term "back-end storage system" to mean "a storage system not directly accessible by the user."

3. QUANTUM'S '810 PATENT.

The three phrases selected for construction in the '810 patent are: (A) "subblock;" (B) "predetermined constraint;" and (C) "an additional hierarchy of subblocks is formed from at least one group of contiguous subblocks."

A. "Subblock."

The term "subblock" appears throughout the claims of the '810 patent. For instance, claim 1 recited:

1. A method for organizing a block b of digital data for storage, communication, or comparison, comprising the step of:

partitioning said block b into a plurality of *subblocks* at at least one position k[k+1 within said block, for which b[k-A+1 ... k+B] satisfies a predetermined constraint, and

wherein A and B are natural numbers.

Quantum proposes the term should mean "a part of a block." Riverbed argues the term should mean "a part of a block consisting of a series of bits or bytes of the digital data from the block." The parties therefore agree that a subblock is "part of a block." The only difference in the construction is Riverbed's added explanatory language.

Both parties rely on the same language in the specification for defining the term "subblock." That language stated (col.2:51-62) (emphasis added):

The term block and subblock both refer, without limitation, to finite blocks or infinite blocks (sometimes called streams) of zero or more bits or bytes of digital data. Although the two different terms ("blocks" and "subblock") essentially describe the same substance (digital data), the two difference terms have been employed in this specification to indicate the role that a particular piece of data is playing. The term "block" is usually used to refer to raw data to be manipulated by aspects of the invention. The term "subblock" is usually used to refer to a part of a block. "*Blocks*" *are partitioned into* "*subblocks*."

Significantly, in the joint claim construction statement, the parties stipulated that the term "block" meant "finite or infinite streams of zero or more bits or bytes of digital data" (Bruno Decl. Exh. C). As such, there

is no need restate that subblock, like blocks, are composed of bits or bytes of digital video. This conclusion is a necessary consequence of the statement that a subblock is part of a block.

Accordingly, this order finds the term "subblock" means "a part of a block."

B. "Predetermined Constraint."

The parties next spill much ink over the term "predetermined constraint." The term predetermined condition appears in every independent claim of the '810 patent. Again, claim 1 recited:

1. A method for organizing a block b of digital data for storage, communication, or comparison, comprising the step of:

partitioning said block b into a plurality of subblocks at at least one position k k+1 within said block, for which b[k-A+1 ... k+B] satisfies a *predetermined constraint,* and

wherein A and B are natural numbers.

Quantum argues the term should mean "criteria for identifying a subblock boundary." Riverbed contends the term should mean:

a function that accepts a window of bits or bytes of length A + B and returns a boolean value to indicate whether a boundary should be placed between the A and B parts of the window. Only the bits or bytes in the window of bit or bytes of length A + B can affect the decision of whether a boundary should be placed at position k k +1. FN2

FN2. In its opening claim construction brief, Quantum argued that Riverbed's proposed construction was incorrect because it limited the window to "bytes" when it was previously stipulated that the system covered "bytes" and "bits." In its opposition brief, Riverbed stipulated to also include "bits" in its proposed construction.

At the claim construction hearing, Quantum indicated that its only objection to the first sentence of Riverbed's construction was that it is unduly long and complicated. The central debate therefore is over the second sentence in Riverbed's construction. Riverbed argues that the only criteria that can be used in determining whether or not a boundary should be drawn consists exclusively of data within the specific window being analyzed. Quantum disagrees. They argue that other criteria can be used to determine whether or not a boundary should be drawn-namely, the amount of positions that have passed since the last boundary was drawn.

This order sides with Quantum. The specification and claims make clear that other data besides the exact bits or bytes in the window being analyzed are considered. Significantly, if only the data within the window could be considered then there is a possibility that the system would draw boundary lines at positions that are right next to one another thereby making the entire process inefficient and unneeded. The '810 recognized this problem and taught that artificial bounds could be created that depended, in part, on the amount of bytes that have passed since the last boundary was drawn (col.14:54-67) (emphasis added):

The user of data-dependent subblock boundaries provides a way to deterministically partition similar portions of data in a context-independent way. *However, if artificial bounds are not placed on the subblock size, particular kinds of data will yield subblocks that are too large or too small to be effective....* A solution to this problem is artificially to impose an upper bound U and a lower bound L on the subblock size. *There seem to be a limitless number of ways of doing this.*

The specification went on to state some examples of how this could be achieved (col.15:6-20):

Lower bound: Of the set of boundaries that bound subblocks less than L bytes long, remove those boundaries that are closer to their neighboring boundaries than their neighboring boundaries are to their neighboring boundaries.

Lower bound: If the block is being scanned sequentially do not place a boundary unless, at least L bytes have been scanned since the previous boundary.

Lower bound: Of the set of boundaries that bound subblocks less than L bytes long, remove those boundaries that satisfy some secondary hash function.

Lower bound: Of the set of boundaries that bound subblocks less than L bytes long, remove randomly chosen boundaries until all the resulting subblocks are at least L bytes long.

Many other schemes could be devised.

This language demonstrates that many other factors could be considered in determining whether or not a boundary should be drawn. In particular, one principal factor was the number of bytes that have passed since the last boundary line was drawn. If a boundary line was drawn at the directly previous byte, it may be less desirable to draw a new second boundary line given the close proximity to the previous boundary.

Claim 4, which depends on claim 1, allows for a boundary to be placed regardless of whether the predetermined condition is met based on the artificial bounds discussed above:

4. The method of claim 1, wherein at least one bound is imposed on the size of at least one of said plurality of subblocks.

Riverbed contends that this language is irrelevant to construing "predetermined constraint" because it goes above and beyond whatever boundaries the predetermined constraint requires. This cannot be so. *First,* the claim language itself dictates that certain boundary lines are drawn regardless of what bits or bytes are in the window. This is a direct contradiction to Riverbed's proposed construction where "only the bits or bytes in the window of bit or bytes of length A + B can affect the decision of whether a boundary should be placed at position k k +1." Even if this construction only refers to the "predetermined constraint" limitation, it only acts to confuse the matter. *Second,* and more importantly, Riverbed gives no response to the fact that its construction makes little sense when read in conjunction with the specification. As already stated, one example the '810 specification taught for imposing artificial bounds was (col.15:10-12):

Lower bound: If the block is being scanned sequentially do not place a boundary unless, at least L bytes have been scanned since the previous boundary.

If only those bits or bytes within the window could be considered, then how would the system know the number of bytes since the last partition (a piece of information not disclosed by the data in the window)?

In support of its construction, Riverbed primarily relies on a paper written by the inventor listed on the '810 patent, Dr. Ross Williams, over six years after the patent issued. That paper was written as a marketing tool for companies and individuals who were interested in purchasing Dr. Williams' company at the time. As Dr. Williams testified (Williams Dep. 156:11-23):

The companies that we were talking to would sometimes want to as about the ['810] patent, and they'd want to know more about it, may go off and try and read the claims.... They'd sometimes have difficulty with it. So we wanted a marketing document that would provide, you know, and easier to understand perspective on the patent for executives who are exposed to it.... And nonexecutives as well. Everyone.

The paper stated in detail exactly how the claims of the '810 patent should be read and what the actual invention was. It stated in relevant part (Briggs Decl. Exh. A) (emphasis added):

The interesting part of the claim is the phrase 'for which b[kA+1 ... k+b] satisfies a predetermined constraint, and where A and B are natural numbers.' This sounds more complicated than it really is. Figure 2 shows what's happening here. *To evaluate whether a partition should be made at the boundary between the bytes at position k and k+1, we evaluate the bytes within a window consisting of A bytes to the left of the position and B bytes to the right. Only the bytes in this window can affect the decision of whether to put a boundary at position k k+ 1.*

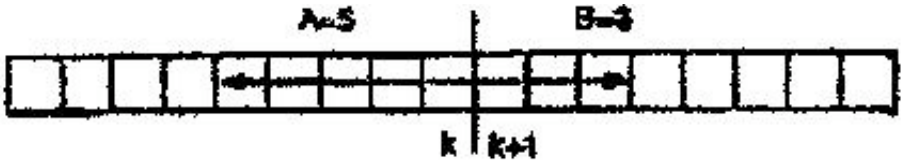

Figure 2: The parsing window.

'A and B are natural numbers" (i.e., 0, 1, 2, 3, ...). This phrase is simply intended to indicate that we are dealing in whole multiples of bytes. In most implementations, A and B will be constants (though the claim does not explicitly constrain them to be).
It may seem overly restrictive to restrict the window being evaluated to A bytes to the left and B bytes to the right (i.e., a window of A + B bytes) of the position being evaluated. However, this is an important aspect of the claim because it captures in the broadest possible way the locality of the window evaluation process. Blocklet boundaries must, above all, be positioned based on data local to the position being evaluated. The degree of locality is left unspecified (in the parameters A and B), but the locality is important because without it, the same blocklets would not be created in the same way when the same data arises in diferent global context.

* * *

The phrase 'predetermined constraint' is purposefully broad, as (assuming we are dealing with bytes), for a given A and B, there are 22^(^8*(A+B)) possible constraint functions that could be used to implement the

method. *All that is meant by the term "constraint" is a function that accepts a block of bytes of length A + B and returns a boolean value to indicate whether a boundary should be placed between the A and B parts of the window.*
In summary, the intent of claim 1 is to capture the concept of dividing a block of data into subblocks of data at data dependent positions where each position is evaluated for boundaryness by examining only those bytes near to an adjacent to the position.

In light of the strong conflict between Dr. Williams' paper and the intrinsic evidence, this order accords little weight to Dr. Willaims' extrinsic statements. As stated by the Federal Circuit in Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1584 (Fed.Cir.2004):

Nor may the inventor's subjective intent as to claim scope, when unexpressed in the patent documents, have any effect. Such testimony cannot guide the court to a proper interpretation when the patent documents themselves do so clearly.

In addition, it would be difficult to confer any probative value on Dr. Williams' former statements given the uncertain context in which they were made. The mere fact that Dr. Williams used certain language to describe the patent to potential non-technical investors sheds little light on how the patent should be construed in the context of litigation. Where the specification is unambiguous, as here, such statements are highly irrelevant.

The only other support Riverbed relies on for its proposed construction is that the only example described in the '810 patent uses only the bytes within the window to determine whether or not a boundary should be drawn. But examples and embodiments in patents should always be read with respect to the specification as a whole. Texas Instruments, Inc. v. U.S. Int'l Trade Comm'n, 805 F.2d 1558, 1563 (Fed.Cir.1986) ("This court has cautioned against limiting the claimed invention to preferred embodiments or specific examples in the specification."). When this is done here, Riverbed's proposed construction fails.

Nonetheless, Quantum's proposed construction of "criteria for identifying a subblock boundary" is overly vague and unhelpful. As already conceded by Quantum, the first sentence of Riverbed's proposed construction is well in line with the '810 specification. This order agrees. Accordingly, the term "predetermined constraint" is held to mean "a function that accepts a window of bits or bytes of length A + B defined by position k and returns a boolean value to indicate whether a boundary should be placed between the A and B parts of the window."

C. "An Additional Hierarchy Of Subblocks Is Formed From At Least One Group Of Contiguous Subblocks."

The term "an additional hierarchy of subblocks is formed from at least one group of contiguous subblocks" only appears in dependent claim 6, which stated:

6. The method of claim 1, wherein additional subblocks are formed from at least one group of contiguous subblock.

Quantum argues that the term be given its plain and ordinary meaning with hierarchy meaning "a system or series of terms of successive rank." Riverbed proposed construction is:

multiple layers of subblocks are constructed from one or more initially contiguous subblocks by progressively restricting a constraint function to generate the multiple layers of subblocks, where the boundaries of the higher layer subblocks are aligned with the subblock boundaries of all the subblock layers below it.

The term only appears once in the written description. That sentence only restates the claim language, "[i]n a further aspect of the invention, the invention provides a method according to aspect 1, wherein an additional hierarchy of subblocks is formed from one or more groups of contiguous subblocks" (col.3:42-45). Figure 9, however, does illustrate the concept graphically.

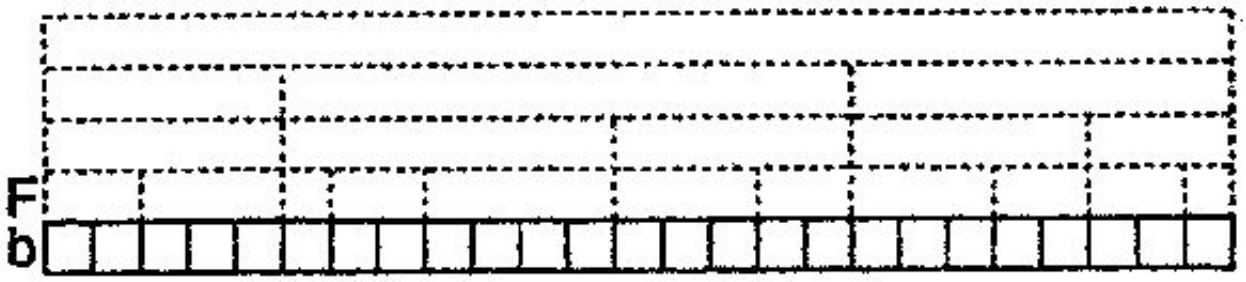

Figure 9

The only sentence in the specification describing the figure stated (col.7:49-50):

Fig. 9 shows how subblocks can be organized into a hierarchy. Such a hierarchy can be constructed by progressively restricting a constraint F.

As seen in Figure 9, as you move up the hierarchy of subblocks, new subblocks are formed by combing two or more subblocks from the previous level in the hierarchy.

Riverbed has not shown why the entirety of claim 6 requires construction. In particular, Riverbed's construction is more complex and confusing than the words directly recited in the claim. Therefore, this order finds the term "an additional hierarchy of subblocks is formed from at least one group of contiguous subblocks" requires no construction and may be sufficiently understood by a layperson.

CONCLUSION

This claim construction order will govern for the remainder of this action, except that, on the motion of the Court (but not the parties) a construction may be altered in the event the Court reaches a more informed understanding of the terms in context, as sometimes occurs in trial or on summary judgment. For the reasons stated above, Riverbed's motion for leave to file a supplemental statement is **DENIED.**

IT IS SO ORDERED.

N.D.Cal.,2008.
Quantum Corporation v. Riverbed Technology, Inc.

Produced by Sans Paper, LLC.