White Paper

Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

Contents

Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

Overview

Aspera Sync is purpose-built by Aspera for highly scalable, multidirectional asynchronous file replication and synchronization. Sync is designed to overcome the bottlenecks of conventional synchronization tools like rsync and scale up and out for maximum speed replication and synchronization over WANs, for today's largest big data file stores —from millions of individual files to the largest file sizes.

Unlike conventional tools like rsync, Aspera Sync reconciles file system changes with remote peers at extremely high speed (500 files per second+) on commodity hardware and does not degrade in performance as numbers of files increase, or with WAN conditions. Built upon Aspera *fasp* transport, Aspera Sync transfers new data between peers at full bandwidth capacity, regardless of distance and network conditions (100X+ over rsync) over long distance and high bandwidth WANs.

Unlike rsync and conventional replication tools which copy any new data over the WAN, Aspera Sync intelligently recognizes changes and file operations such as moves and renames, instantaneously propagating these to remote peers, avoiding what can be hours of unnecessary copy times. Also, unlike unidirectional-only tools, Sync supports bidirectional and multidirectional synchronization topologies where content is changing on multiple nodes. Replication jobs can be configured to run continuously for real-time synchronization, or one-time, on demand.

Finally, Sync ensures 100% data integrity: users can safely move, rename, and delete files or entire directory structures, without fear of data loss.

Synchronization Challenges over Distance

Transferring, replicating, or synchronizing file-based assets over wide-area networks (WANs) presents serious challenges using traditional TCP-based replication tools. The latency and packet loss created over distance disrupts replication over TCP, especially for larger transmissions such as big files. While TCP can be optimized, it remains an unreliable approach to large scale delivery or acquisition of file-based assets—for businesses and industries that require predictable performance, highly efficient bandwidth utilization, reliability, and security.

TCP Acceleration Appliances

There has been significant innovation in TCP optimization and acceleration techniques using in-band hardware appliances and storage devices. Mainly, these approaches utilize compression and de-duplication to minimize network traffic (e.g., chatty protocols) and incrementally send block-changes in support of application-specific workflows, between appliances.   This approach effectively optimizes certain applications from sending unnecessary information and data over the network, including some file synchronization scenarios.

However, these acceleration appliances do *not* expedite big data freighting and synchronization scenarios where entire files (sometimes very large numbers of files) need to be transferred, replicated, or synchronized, in bulk.

Ad Hoc Software Tools

Today, a number of low-cost TCP-based file replication and synchronization tools are available through the open source community and software vendors. These range from open source tools like rsync to an entire ecosystem of do-it-yourself (often unsupported) tools for Mac and Windows. Typically this software is host-based (i.e., runs on each server) and can be used to replicate files point-to-point (unidirectional or one way) over TCP.

Aspera, Inc.   5900 Hollis Street Suite E   Emeryville, CA 94608   USA   www.asperasoft.com

Rsync

For small data sets across relatively low-latency wide-area connections, rsync provides an efficient unidirectional approach; it can also be configured to run back and forth, unidirectional, between endpoints.

Some history: For years, rsync has been the status quo Linux replication tool, and eventually made its way onto Windows. When Andrew Tridgell designed rsync in the late 90's, file sizes and file systems were relatively small, counted in gigabytes—not terabytes or petabytes! Rsync's approach, while innovative at the time, relies on scanning the file system, and reading all files into memory to acquire information on file changes and deltas. In rsync vernacular, file change information is referred to as "blocks" (not be confused with block changes). Rsync stores this information about each file in memory on the source and target systems. Rsync then communicates over TCP to compare local file chunks on the source system with remote file chunks on the target to make decisions on which files to replicate.

Replication speed is limited by two factors: the time it takes to scan the local file system, and TCP. As files are created, moved, or changed on the source file system, the scanning process takes progressively longer to execute. As distance between source and target servers increases, latency and packet loss worsen on average, limiting the speed of the comparison of the local and remote systems to find the deltas, and the speed of replication over TCP. Thus, rsync works fine for small data sets over low-latency networks. But, as data sets grow and network latency increases, rsync scans become impractical.

Another issue is usability. If a user accidentally moves or renames a directory on the source server, its entire contents must be rescanned and resent (replicated) to the target server. These caveats can be extremely costly when the data set is large.

Scalability Challenges for Big Data Synchronization

Rsync and other low-cost Windows tools may work fine in support of smaller data sets and simple replication scenarios. But as data sets grow and network conditions degrade over distance, users should investigate alternative tools. Issues to consider include: number of files, file size, file system size, directory structure, network speed, and network distance and quality factors such as latency and packet loss rates.

Number of files

As the sheer number of files stored in a file system grows, it's important to have a highly efficient system for acquiring and propagating file change information (i.e., a change delta) of each file stored on source and target systems. Some customers need to synchronize thousands to millions of files in a single file system. To accomplish this efficiently requires some level of integration with the underlying file system—either through change notification or rapid snapshot capabilities.

TCP appliances and ad hoc software tools such as rsync are typically not integrated, as such.

File size

Today files are measured in multiple megabytes, gigabytes, and terabytes. Customers placing high value on file-based assets require reliable, predictable replication techniques of large files.

The limiting factor to synchronizing many large files using TCP over distance is usually the condition of the network. Conventional TCP-based approaches to synchronization fail transferring large files over long distances: TCP replication throughput does not scale over distance. TCP throughputs are $1/10^{th}$ to $1/1000^{th}$ of the available bandwidth over higher bandwidth networks on typical global WANs.

One provincial solution: Have each replication end-point break up files into smaller chunks, transfer the chunks, and reconstruct the file on the target. This process is slow – limited by the poor TCP throughput over the WAN – and at best error prone and unpredictable.

Directory structure

Directory structure layouts will affect the performance of most replication solutions. The layout and structure needs to be taken into consideration in cases where many files need to be replicated. Some systems will perform well on deep directory structures with fewer directories. Others will perform well on flat directory structures, with fewer files per directory. An ideal replication solution recognizes and transfers directory changes at high speed, independent of structure.

Large file systems

Among data-intensive industries, file systems are often measured in the many gigabytes to terabytes to petabytes. As file system size increases, the synchronization solution needs to be file-aware.

The replication system should able to:

- Efficiently acquire metadata on files, from change deltas to other properties.
- Scale to take advantage of modern scale-up and -out server systems.
- Scale network throughput to fully utilize available bandwidth—in some cases up to 10Gbps—and available computing resources on both source and target endpoints.
- Protect data in the event of failure: If the system crashes or the file transfer is interrupted—the synchronization application should be able to gracefully restart and retransfer all (and only) files that failed to transfer or synchronize.
- Ensure data integrity, through the use of checksums or transactional semantics, in the event of failure or network interruption.

Network utilization—for maximum throughput and predictable replication times

Network bandwidth—especially WAN bandwidth—is a precious resource. Additionally, fast and predictable replication times depend upon efficient, full use of the available WAN bandwidth. An ideal synchronization solution should both maximize network throughput and fully utilize available bandwidth.

In contrast, TCP-based solutions (FTP, SCP, etc.) do not efficiently utilize bandwidth. As conditions degrade over distance, TCP-based tools in some cases utilize merely 1/100th of the available bandwidth over higher speed networks at global distances. Worse, naïve data blasting solutions often fill bandwidth but with wasteful un-needed data, collapsing the effective throughput. Both conventional and naïve blasting solutions offer inferior approaches to bandwidth utilization.

Distance Limits Throughput, Forces Trade-offs with TCP

Conventional TCP replication strategies and design decisions are often the product of working around the constraints of the protocol. As latency and loss increases, replication over TCP degrades or fails, because the effective speed of TCP collapses over distance. Unfortunately, high speed networks from 1-10Gbps are *just as prone* to latency and loss over distance as lower speed networks.

In support of Content Delivery Networks (CDNs), for example, today customers replicate large amounts of data from the center (origin) to the edge closer to users to subvert the distance limitations of TCP. This requires expensive storage, compute, and other overhead, as redundant data must be cached in multiple locations.

A New Model for N-way Synchronization over Distance

Aspera Sync is purpose-built by Aspera for highly scalable, multidirectional asynchronous file replication and synchronization. Sync is designed to overcome the bottlenecks of conventional synchronization tools like rsync and scale up and out for maximum speed replication and synchronization over WANs, for today's largest big data file stores —from millions of individual files to the largest file sizes.

Unlike conventional tools like rsync, Aspera Sync reconciles file system changes with remote peers at extremely high speed (500 files per second+) on commodity hardware and does not degrade in performance as numbers of files increase, or with WAN conditions. Built upon Aspera fasp transport, Aspera Sync transfers new data between peers at full bandwidth capacity, regardless of distance and network conditions (100X+ over rsync) over long distance and high bandwidth WANs.

Scalable, Location-independent Transport—the Foundation of File Delivery

Aspera solutions are designed for customers requiring big bulk data file transfer, replication, and synchronization. Ideal scenarios include: content ingest and acquisition (on-premise and into the cloud), large-scale content distribution, remote collaboration, mobile upload and download, and replication across workflows.

Often, customers share common problems: reliably and securely transferring large high-value assets (file) in more predictable ways. Aspera Sync has advanced capabilities for replicating many small files—and large files, concurrently. Replication jobs can be comprised of a mix of very large files or many small files.

The approach Aspera has taken is founded on the following design principles:

- Open, cross-platform: Working across deployed industry-standard infrastructure (networks, storage, servers) and mainstream operating systems such as Linux, UNIX (Solaris, BSD), Mac, and Windows.
- Integrated: Rapid acquisition and propagation of file changes requires both integration with source and target systems—and the underlying transport, Aspera *fasp*.
- Location-agnostic: Building on advancements in *fasp*, synchronization and other applications fully utilize the available bandwidth of any IP network, across any distance.
- Optimally scaled—for a variety of file types—and compute, storage, and network performance to enable fast replication of bulk data sets over long distances.
- Transparent and easy to use: Tools are familiar to administrators and easy to use. Users should interact with the system just like their familiar OS environment (Linux, Windows, Mac, etc.).

- Available, resilient, and robust: All aspects of the solution should operate continuously and be resilient to failures, without compromising data integrity.
- Secure: All aspects of security should be considered: Network, access, authorization, identity management, encryption, and antivirus.
- Predictable: through the design principles above, customers will be able to obtain predictability in support of a wide range of file transfer and freighting scenarios, including synchronization.

Open, Cross-platform Software

Aspera's solutions are compatible with industry-standard networks, storage, servers, and operating systems.[1] This enables utilizing any type of storage supported by the host operating system—and interoperating across multiple vendor solutions.

Aspera also provides an SDK for embedding and integrating the *fasp*™ transport in workflows, web applications, and a number of collaborative 3rd-party line-of-business applications.

Integrated, Location-agnostic Synchronization

At its core, Aspera Sync builds on Aspera *fasp* to provide an integrated, high-speed transport used for synchronization and other file transfer operations.

Built on Aspera *fasp* Transport

Aspera's *fasp* transfer technology provides the core innovative transport, eliminating network-layer bottlenecks, irrespective of distance. Aspera *fasp* scales transfer speeds over any IP network, linearly. The approach achieves complete throughput efficiency, independent of the latency of the path and robust to packet losses. For known available bandwidth, file transfer times can be guaranteed, regardless of the distance between endpoints or network conditions—over satellite and long distance or unreliable international links.

Aspera Sync inherits all lower- network-level capabilities from *fasp*. These include speed and bandwidth efficiency, bandwidth rate control, robustness, security, file-specific metadata support, and extensibility. Through the *fasp* SDK, APIs are available to extend applications and embed *fasp* capabilities. [2]

Performance Results Compared to Rsync

Aspera Sync (async) performance improvements over rsync and other synchronization systems by 10-100x, both in stable networks or in the most extreme conditions (such as 500 msec delay; 10% packet-loss). Aspera performed a set of tests to baseline performance compared to rsync.

The tests provided the following:

- Reproducing real-world WAN conditions—with 100mls delays and 1% packet loss over a 1Gbps link.
- Small files: measuring the performance and results of replicating many small files—into the millions.
- Large files: measuring the performance and results of replicating large files—into terabytes.
- Time: measuring how long it takes to perform both small and large file replication scenarios.
- Throughput: measuring the overall throughput utilized during the replication job.
- Change replication: changing a fraction (10%) of the files in the data set and measuring the time and throughput to replicate the changes.

 Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

Many smalls files

In conditions with 100mls latency and 1% packet loss, millions of very small files can be replicated from a source to a target. File sizes were extremely small averaging 100KB per file, but varying in size from a few KBs to 2MBs. Async completed the replication of the file set in 2.8 hours, whereas rsync would have required 9.4 days given its steady state throughput of less than 1 Mbps for transferring a fraction of the files.

In this case, async provided a performance improvement of 81x.

Table 1 - Performance Comparison Synchronizing Many Small Files (Average size 100 KB) over WAN of 100 ms / 1%

Small file performance	Number of files	Data set size (GB)	Sync Time	Throughput (Mbps)
Async	978944	93.3	9968 sec (2.8 hours)	80.4
Rsync	978944	93.3	814,500 sec (9.4 days)	0.99
			Aspera Speed Increase	**81X**

Large files

Large file replication performance scales linearly with available bandwidth. Due to *fasp*, async maintains the same speed for files of *any large size*. As a baseline, on a 1 Gbps WAN with 100ms latency and 1% packet loss, a 500GB volume comprised of thousands of larger files can be replicated from a source to a target in under one-and-a-half hours. By contrast, an rsync transfer (stabilizing at just under 1 Mbps throughput) will take 50 days to complete.

In this example, Aspera Sync improves performance over rsync by 940X.

Table 2 - Performance Comparison Synchronizing Large Files (Average Size 100MB) over WAN of 100 ms / 1%

Large file performance	Number of files	Data set size (GB)	Sync Time	Throughput (Mbps)
Async	5194	500.1	4664 sec (1.3 hours)	921
Rsync	5194	500.1	1222 sec (50 days)	0.98
			Aspera Speed Increase	**940X**

File Consistency and Fast Synchronization of New Files & Changes

Aspera Sync quickly captures file changes through snapshots and file change notifications. This enables Aspera Sync to quickly acquire and replicate files based on changes. Jobs can quickly restart if the host is shut down. Async both quickly captures and synchronizes file changes within a very large and dense directory tree, whereas rsync is dramatically slower. Aspera tested the time to synchronize for cases in which a proportion of new files were added to an existing file set. To an initial set of nearly 1 million small files, approximately *30,000 new files were added*. Async synchronized the changes in 16 minutes; rsync requires over 10 hours.

Table 3 - Synchronization time when adding 31,056 files to 1 million small files (100 KB each) over WAN of 100ms/1%

Change File Performance	No. existing files	No. files added	Total size (GB)	Sync Time	Throughput (Mbps)
Async	978944	31056	2.97	947 sec (16 min)	26.9 Mbps
Rsync	978944	31056	2.97	37076 sec (10.3 hours)	0.68 Mbps
				Aspera Speed Increase	**39X**

 Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

In a large directory consisting of thousands of larger files, 10% more files were added. On a 1Gbps WAN of 100ms/1%, Async is able to synchronize the changes in 54 seconds, a throughput of 870 Mbps, as compared to rsync, which requires 15 hours, 1000X slower.

Table 4 - Synchronization time when adding new files to set of large files (100 MB each) over WAN of 100ms/1%

Change File Performance	No. existing files	No. files added	Total size (GB)	Sync Time	Throughput (Mbps)
Async	5194	54	5.49	54 sec	871 Mbps
Rsync	5194	12	5.49	54,573 sec (15 hours)	0.86 Mbps
				Aspera Speed Increase	**1000X**

Moreover, file consistency is maintained during all aspects of remote replication. Native replication checksum and error correction are built into the system to ensure file consistency and integrity during all sync operations.

When content is transferred, data integrity verification ensures each transmitted block is protected. Integrity verification is provided natively through *fasp*, Aspera's high-speed transport. *fasp* accumulates a cryptographic hashed checksum (using 128-bit AES) for each datagram, appending the digest-hash to the secure datagram before it goes on the wire, to be then checked at the receiver to verify message integrity, preventing man-in-the-middle attacks and ensuring data integrity.

Error Checking During File Transfer

In addition to the integrity verification performed on every data block after reception, the Aspera *fasp* transfer protocol guarantees a bit-for-bit identical transfer of the source file to the destination file, if the transfer session finishes with a successful status. This property results from the overall design of the protocol. Automatically resumed transfers verify by default that the fragment of the file on disk matches the fragment at the source by checking not only file attributes, but also performing a checksum on the fragment contents.

Maximum Throughput over Distance

At the transport level, *fasp* enables full utilization of network bandwidth through the concepts of rate control and policies. While *fasp* can fill any available bandwidth, *fasp* also includes an intelligent adaptive rate control mechanism that allows transfer rates to throttle down for precision fairness to standard TCP traffic, but automatically ramps up to fully utilize unused bandwidth.

The adaptive rate control algorithm is an original equation-based approach. When TCP's rate is self-limited on an uncongested link, *fasp* detects the unclaimed bandwidth and ramps up to fill it. When congestion builds up, *fasp* reduces rate to the TCP rate and equally shares the link with multiple TCP flows. This approach has fundamental benefits over the window-based flow control algorithm used by standard TCP and even new accelerated or "high-speed" TCP's.

Compared to TCP/Accelerated TCP, Aspera *fasp* adaptive rate control is:

- Loss tolerant. Reacts only to true congestion, while remaining immune to inherent channel loss.
- TCP fair. Quickly stabilizes at a TCP-friendly rate when links are congested without squeezing small traffic.
- Perfectly efficient. Ramps up to fill unclaimed bandwidth, regardless of latency and packet loss.
- Stable. Zeroes in on the available bandwidth, and runs "flat" without oscillation, for stability.

 Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

Ease of Use

Aspera provides exceptionally easy to use administration tools. To the Linux administrator, the command line tools look very much like Linux tools such as rsync. Aspera Console, by contrast, provides a centralized, web-based single-point of management for all Aspera products, including Aspera Sync. The console enables administrators to centrally manage, monitor, and control transfers and replications across endpoints, called nodes.

Honoring directory semantics

A key capability of Aspera Sync is the ability to *rename*, *move*, *delete*, and copy files and directories on any source and target endpoint included in a synchronization job. This means end users can safely interact with files and directories exactly as they would on Linux, Windows, or Mac. If a user accidentally deletes or renames a file or directory, the file or directory can be re-sync'd to restore.

Aspera Console

Aspera Console provides real-time notification, logging and reporting capabilities, while maintaining a centralized transfer history database for detailed auditing and customized reporting.

Aspera Console's role-based access control allows users to monitor, start transfers, and generate fully customizable reports. From the Aspera Console, users can also automate files transfers—including multi-site synchronization with Aspera Sync and full business process orchestration with the Aspera Orchestrator—and integrate with third-party applications. Administrators have the ability to centrally configure all managed Aspera nodes, create and manage users and user groups, initiate and automate transfer jobs, and precisely control and monitor all transfer and bandwidth utilization parameters.

Complete Security

The *fasp* protocol provides complete built-in security without compromising transfer speed. The security model, based on open standards cryptography, consists of secure authentication of the transfer endpoints using the standard secure shell (SSH), on-the-fly data encryption using strong cryptography (AES-128) for privacy of the transferred data, and integrity verification per data block, to safeguard against man-in-the-middle and anonymous UDP attacks. The transfer preserves the native file system access control attributes between all supported operating systems, and is highly efficient.

Secure endpoint authentication

Each transfer session begins with the transfer endpoints performing a mutual authentication over a secure, encrypted channel, using SSH ("standard secure shell"). SSH authentication provides both interactive password login and public-key modes. Once SSH authentication has completed, the *fasp* transfer endpoints generate random cryptographic keys to use for bulk data encryption, and exchange them over the secure SSH channel. These keys are not written to disk, and are discarded at the end of the transfer session.

For public key authentication, the private keys are stored encrypted on disk using a secure, private passphrase and authentication is done using RSA only (SSH-v1) or RSA/DSA (SSH-v2) public key exchange. The ssh-keygen program is distributed with the Windows version of Aspera Scp for generating DSA and RSA keys. The default key length is 1024 bits although the user may request longer key lengths.

Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

On-the-fly data encryption

Using the exchanged keys, each data block is encrypted on-the-fly before it goes on the wire. *fasp* uses a 128-bit AES cipher, re-initialized throughout the duration of the transfer using a standard CFB (cipher feedback) mode with a unique "initialization vector" for each block. CFB protects against all standard attacks based on sampling of encrypted data during long-running transfers.

Directory services and identity management

Aspera offers an option to integrate with LDAP servers and Active Directory. Both users and groups can be created in LDAP and used in conjunction with Aspera's client and server products.

Network-level security and VPNs

Aspera transfers and synchronization jobs can be tunneled within Virtual Private Networks (VPNs) over any standard IP network. Firewalls and other perimeter security measures supporting UDP such as Network Address Translation (NAT) are fully compatible.

Resilience and Availability

File replication and synchronization are resilient to end-to-end failures, from the host system (source and target) and across the network. If a failure occurs on the source, jobs quickly restart where the job left off. Aspera Sync can also be load-balanced across end-points. If a synchronization job is interrupted, Aspera Sync will resume at the point at which the last file was transferred. (Some tools like rsync require the entire job to start over.)

Deployment Modes, Models, and Use Cases

Modes

Aspera Sync provides two modes of operation: one-time synchronization and continuous.

One-time Synchronization

Ideal for testing and replicating in ad hoc ways, one-time sync can be scheduled or manually initiated. In this mode, all endpoints defined in the job will synchronize once.

When the job is complete, no further action will be taken.

Continuous Synchronization

Most ongoing replication operations will run in continuous mode. Continuous sync can be scheduled or manually initiated. Once scheduled, synchronization occurs transparently in the background and continues as files or directories are added, updated, or changed.

Models

Unidirectional

Much like the rsync model, Aspera Sync supports replicating files and directories from a source to a target. If updates are being maintained at a single location and the sole goal is to propagate the updates to a target server, this scenario may be adequate. Two-way unidirectional is also supported.

 Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com

Bidirectional

Bidirectional synchronization occurs between two endpoints.

While any number of use cases could be supported, some examples include:

- Point-to-Point Synchronization: Files are kept current and in sync between two servers or sites.
- Disaster Recovery: A primary site replicates to a remote secondary site, used for backup.

Multidirectional

Multidirectional replication can be used in support of many advanced synchronization scenarios. These include but are not limited to:

- Hub-and-spoke synchronization, where a core source (hub) server or cluster replicates to n-number of endpoints (spokes). This configuration could support distributed workflows, remote or branch office replication, and any other scenario requiring replication to multiple endpoints concurrently.
- Content Distribution: In cases where a customer needs to synchronize downstream caching points, regional trunks, or other peering points, Aspera Sync can be used exclusively or in combination with other products.
- Cloud Ingest and Distribution: Aspera Sync can be used in situations where local files need to uploaded and subsequently synchronized with multiple repositories within the cloud data center.
- Collaborative File Exchange: Aspera Sync can also be used in support of collaboration, where one user may upload a file and need to synchronize the file with multiple users or remotely located sites.

Content Delivery Networks

In support of Content Delivery Networks (CDNs), through location-independence for replication operations Aspera supports consolidation of caching points. Aspera Sync enables scaling a core (source) server or cluster, the network, and delivering data directly to (n-number of target) endpoints. What’s required is the ability to scale the core (for compute, storage, memory), scale the bandwidth between the core and endpoints, and use Aspera Sync to scale the transfer and replication speed, linear to available bandwidth, over distance, to n-number of endpoints.

Summary

Aspera Sync is purpose-built by Aspera for highly scalable, multidirectional asynchronous file replication and synchronization of big data. Sync is designed to overcome the bottlenecks of conventional synchronization tools like rsync and scale up and out for maximum speed replication and synchronization over WANs, for today’s largest big data file stores – from millions of individual files and to the largest file sizes.

Unlike conventional tools like rsync, Aspera Sync reconciles file system changes with remote peers at extremely high speed (500 files per second+) on commodity hardware and does not degrade in performance as numbers of files increase, or with WAN conditions. Built upon Aspera *fasp* transport, Aspera Sync transfers new data between peers at full bandwidth capacity, regardless of distance and network conditions (100X+ over rsync) over long distance and high bandwidth WANs.

Aspera Sync intelligently recognizes changes and file operations such as moves and renames, instantaneously propagating these to remote peers, avoiding what can be hours of unnecessary copy times. Aspera Sync supports bidirectional and multidirectional synchronization topologies where content is changing on multiple nodes. Replication jobs can be configured to run continuously for real-time synchronization, or one-time, on demand.

Finally, Sync ensures 100% data integrity: users can safely move, rename, and delete files or entire directory structures, without fear of data loss.

[1] Aspera Sync is currently supported on major versions of Linux. Windows and Mac versions are targeted for a follow-on release. Aspera Enterprise Server, *faspex*, Connect Server, and Aspera Console support Linux, Windows, Mac, Solaris, BSD, and Isilon OneFS.

[2] An SDK specifically for Aspera Sync is not yet available but is planned in a post-version 1.0 release.

Aspera, Inc. 5900 Hollis Street Suite E Emeryville, CA 94608 USA www.asperasoft.com