Scalable CloudTurbine Networks

Matt Miller, Cycronix, 1/23/2018

Overview

CloudTurbine (CT) scalability, that is how many simultaneous users may be supported, depends on a number of factors. Unlike a central-server approach (e.g. DataTurbine), CT enables each data producer and consumer (viewer) to simply write or read local files, respectively. To this extent, data producers and consumers do not directly impact each other and therefore there is unlimited scalability. What impacts CT scalability is the ability to move and access the CT data files across a network.

CT, by design, does not stipulate how your data streams (via files) are distributed and accessed. This task is delegated to a wide selection of third party and standard operating system level utilities, such as network drives, file sharing (e.g. Dropbox), FTP, sneaker-net, etc. It is a system integration task to configure your file-transport network, and this document discusses various options with associated scalability issues.

System Building Blocks

To build a scalable CT network, there are a variety of ways to connect (move files between) distributed systems. Each way has unique performance, ease of use, and security implications. CT enables a "building block" approach to assembling distributed streaming data systems, several examples of which are discussed as follows. In these examples, the "WebScan" browser app is shown as the data viewer, but any CT data consumer (e.g. Matlab) may be used.

In the examples below, it is presumed CT data is distributed over network connections. Keep in mind that CT also works well where data producers and consumers access local files in a common folder.

Single Source to Sink System

Many data acquisition scenarios entail a data source which needs to produce and store data but does not necessarily need to view other distributed data. This type of application may efficiently "push" data over a one-way data transfer mechanism such as UDP or a variant of FTP (SFTP, FTPS, SSHFS).

A simple yet useful system is a single source streaming data to a single CT station supporting one or more viewers:

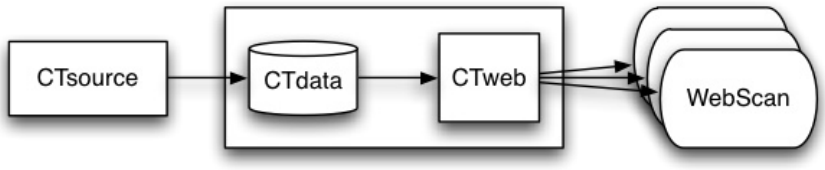

Figure 1: Single Source to Sink CloudTurbine System

Unidirectional data producers, utilizing UDP or FTP for file transfer, are very efficient per-source at sustaining continuous high-rate throughput. The total number of such sources is limited by the administrative practicality and operating-system limits on the number of simultaneous connections. E.g. many commercial web servers limit concurrent FTP connections to a small number, typically on

order of 10 or 100, but not unlimited.

Clusters of Users via Web Interface

For data consumers which only need to ingest but not share or produce their own data, a central distribution server such as the "CTweb" (HTTP) application may provide a convenient last-mile connection for CT streaming data.

Stand-alone data consumers utilizing an HTTP web server may bottleneck at the ability of this server to handle many connections, and the speed with which CT data files may be accessed on a common filesystem. The simple "CTweb" application included with CT open source distribution is designed to be more of a "personal" web service using embedded Jetty[1], with on the order of 10 to 100 users. This approach could potentially expand to handle thousands of concurrent users should it be adapted to use powerful web server technologies such as Apache or IIS.

The figures below show different approaches to scaling up resources. The left panel illustrates mirroring a source to multiple CTdata/CTweb sites, for NxM scalability. The right panel adds a "CTproxy" layer that load-balances and/or aggregates data from multiple sources.

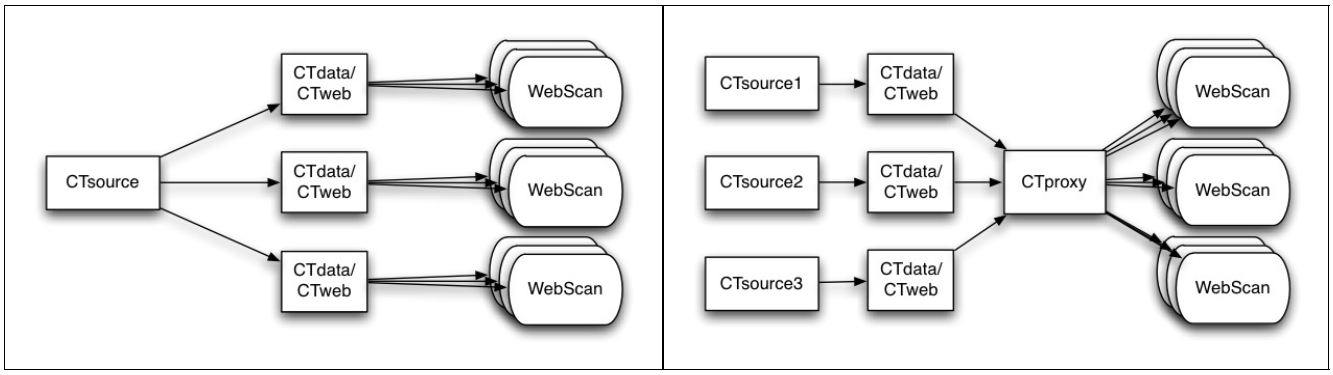

Figure 2: CloudTurbine Scaling via Mirroring and Proxy Servers

Distributed Collaboration via File Sharing

For cases where sites are both data producer and consumer, the broad category of file-sharing services provide convenient options. Third party services such as Dropbox, OneDrive, Google Drive, Syncthing provide easy ubiquitous data access. Within a local network, LAN-based file servers may work well, such as Network-Attached-Storage (NAS) and mapped drives (NFS, SMB, AFP). The figure below shows how a nominally open-ended number of sites may share CT data via file sharing.

1 http://www.eclipse.org/jetty/documentation/9.3.x/embedding-jetty.html

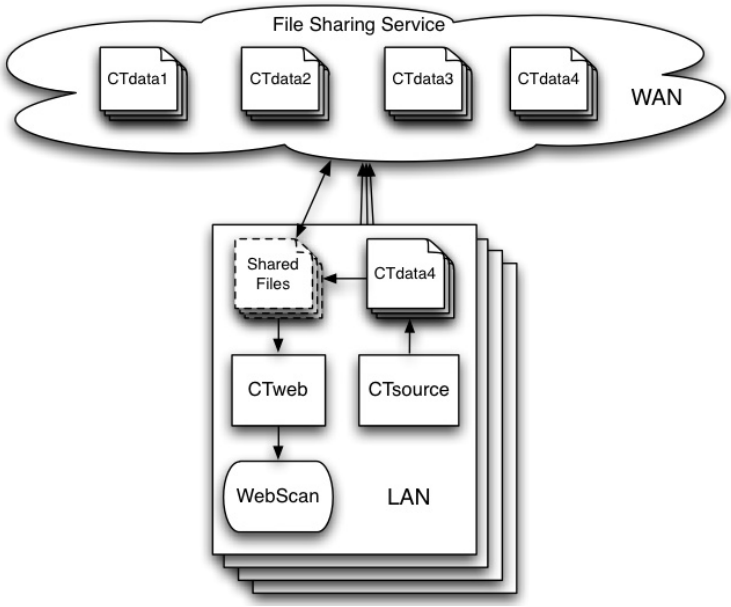

Figure 3: Distributed CloudTurbine Streaming via File Sharing

Third party file sharing services (e.g. Dropbox) in one sense scale globally to millions of users. On the other hand, how many simultaneous users can share a common file folder is typically not well defined. For example, Dropbox answers the question with "There are no limits on how many people you can share your files with[2]", then states elsewhere "You should be able to invite several hundreds of users[3]". The practical answer may be difficult to ascertain, but the file-sharing paradigm of most third party services seems to presume on the order of "hundreds" of users, not truly unlimited global scale distribution of commonly shared data.

Hybrid Approaches

For systems with a variety of participants, combinations of the above connections may be appropriate. For example, a limited number of trusted unidirectional data producers may pool data for an open-ended number of anonymous data viewers. Or when the participants are comprised of a limited number of fixed sites, several unidirectional connections (perhaps one in each direction) may produce the most efficient and secure solution.

The figure below shows how the "CTrollaball" demo[4] can scale to support groups of users on low-latency local networks (LAN), while being connected to remote users over higher-latency (WAN) connections via file-sharing. Note the bi-directional "Player" connections; here the data producers (game players) are also data consumers in that they see and interact with other players.

2 https://www.dropboxforum.com/t5/Sharing-and-collaboration/Is-there-a-limit-to-how-many-people-I-can-invite-to-view-a/td-p/159745

3 https://www.dropboxforum.com/t5/Dropbox/Max-users-for-a-shared-folder/idi-p/50739

4 http://www.cloudturbine.com/ctrollaball-demo/

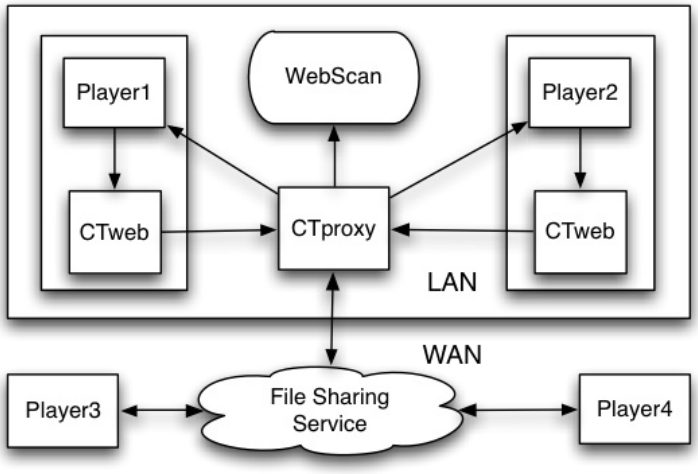

Figure 4: Hybrid CloudTurbine System for Distributed Multiplayer Gaming

Scalability Metrics

In each system discussed above, end-to-end performance and scalability often depends on the weakest-link or choke-point in the flow of data from producer to consumer. There is no such thing as a single-site million-user streaming data system; scalability is determined as much by the connected-system topology as by the performance of any one module.

The following table illustrates general scalability issues and examples of how to select file data transport mechanisms depending on the number of sources (producers) and sinks (consumers) of data. This table is notional only; metrics for specific system configurations will be gathered as CT systems are deployed and tested.

	Approach	Lower Latency (<1s)	Higher Latency (~10s)
Sources			
1 - 10	UDP, FTP	X	
10+	File Sharing		X
Sinks			
1 - 10	Local CTweb	X	
10+	File Sharing		X

There are ways to exponentially expand the number of supported systems via hierarchical system topologies, such as web server farms or clusters that spread load between multiple systems. For example, 10 data sources per UDP data acquisition site, aggregated via file sharing with 10 other sites, each with 10 web portals grouped as a web cluster, might provide 10^3 or a thousand data sources available to a given web viewer. Increase the number from 10 to 100 at each step (requiring significant yet practical resources), and the potential number of users exponentially expands to a global scale.

There are trade offs between scalability and performance, most notably latency, as the type and number of layers of connections between data producer and consumer vary. As CT is deployed for a range of applications, best-practice and performance/scalability numbers will continue to be gathered. CT is a system integration tool; what might be built with it is an on-going exploration.