Strings Synchronization Model

Synopsis

This document describes a collection of beads and conventions developed to support multi-host synchronization in Strings.

Overview

The synchronization model consists of the following components:

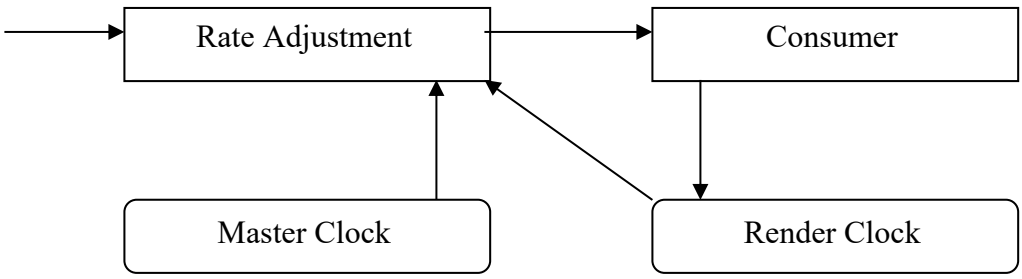
i) An inter-host clock monitoring protocol, implemented in **timesync**, which determines the relative system-clock offset for each synchronized host.

ii) A sample-clock implementation (**sampleclock**), which distributes stream-position information across all cooperating hosts. Each playout node has a 'master' clock, which records the idealized playout position, and a 'render' which records the actual playout position. The **clocksync** bead provides a mechanism for propogating the path clocks across network boundaries, using data collected by **timesync** to convert remote system clock values into local system clock values.

iii) A media-specific mechanism for adjusting the playout rate of a given media type. For rgb video, the **rgbvideo** times frame deliver based on the master clock, and sets the render clock. For pcm audio, the **speaker** bead sets the render clock, and the **audiosync** bead adjusts the playout rate of the sample stream to try to minimize the error between the master and render clocks.

One playout node must be identified as the time master. For simple audio/video playout, generally the master will be the audio stream, since audio playout rates are effectively regulated by the consumption rate of the audio output device.

The master clock for all synchronized streams is a reference to the single render clock for the master time source. Thus the time master's clocks will always match exactly. All other streams vary playout rate to try to minimize the error between the render and master clocks.



TimeSync

The TimeSync bead is used to create a database of clock offsets for all strings machines on the network. The protocol used to estimate clock offsets is based on the NTP protocol.

Periodically each host broadcasts the current time, plus a list of all known remote hosts. The broadcast includes:

1. A psuedo-random host identifier,
2. The local system time at the time of sending.

For remote host known by the sender, the broadcast also includes:

1. The unique identifier of the remote host,
2. The send time from the most recently received broadcast from that host.
3. The local system time when the most recent broadcast was received.

Computation of inter-host clock offset occurs whenever a broadcast is received from a remote host containing a host entry for the local host. Timing is derived from the round-trip consisting of a prior broadcast from the local machine (B0) followed by the remote machines broadcast (B1) which includes values from B0.

T0 is local time when B0 was sent,
T1 is the remote time when B0 was received,
T2 is the remote time when B1 was sent,
T3 is the local time when B1 was received.

We want to compute an estimation for Toffset, such that Tremote + Toffset = Tlocal. Network latency is unknown and variable.

L0 is the latency for B0,
L1 is the latency for B1.

such that

T1+Toffset = T0 + L0
T2+Toffset = T3 – L1

From this we can determine bounds for Toffset:

(T1+Toffset) >= T0
T2 >= T1
T3 >= (T2+Toffset)

And so:

Toffset > T0 – T1
Toffset < T3 – T2

Since we have no way to devolve the effects of L0 and L1, we assume they are approximately equal, and so

Toffset ~= ((T3-T2) – (T0-T1)) / 2

Since latency does vary substantially over time, we reduce the error by averaging subsequent observations. Currently timesync uses the average of the last 8 observations.

Related Issues

Use of Durable Values

To minimize the effects of latency on synchronization accuracy, all shared information is exchanged in terms of stationary or durable values – i.e. values that do not change rapidly over time, and do not assume instantaneous delivery. For example, sample clocks are computed from the following information:

1. The local system clock value at which the stream position was recorded,
2. The sample position at that time,
3. The nominal frequency at which the sample count is expected to be progressing.

From these three values the current playout position of a remote stream can be estimated based on the most recently received sample clock information. Assuming the actual playout rate is close to the nominal playout rate, the accuracy of the estimate does not degrade greatly with increased delivery latency. Use of this principal greatly increases the robustness of the synchronization model.

Clock Uncertainty

The following types of timing uncertainties complicate synchronization:

1. System clock values between machines may have a large relative offset, and a small drift relative to each other.
2. Output devices do not necessarily consume media at exactly the correct rate. The output rate for audio is typically regulated by a DSP clock, not the system clock, and there may be substantial differences between them.