**Jetsons Theme Playback Using Multithreading Concepts, C++**

**By**

**Charan Teja Allada,**

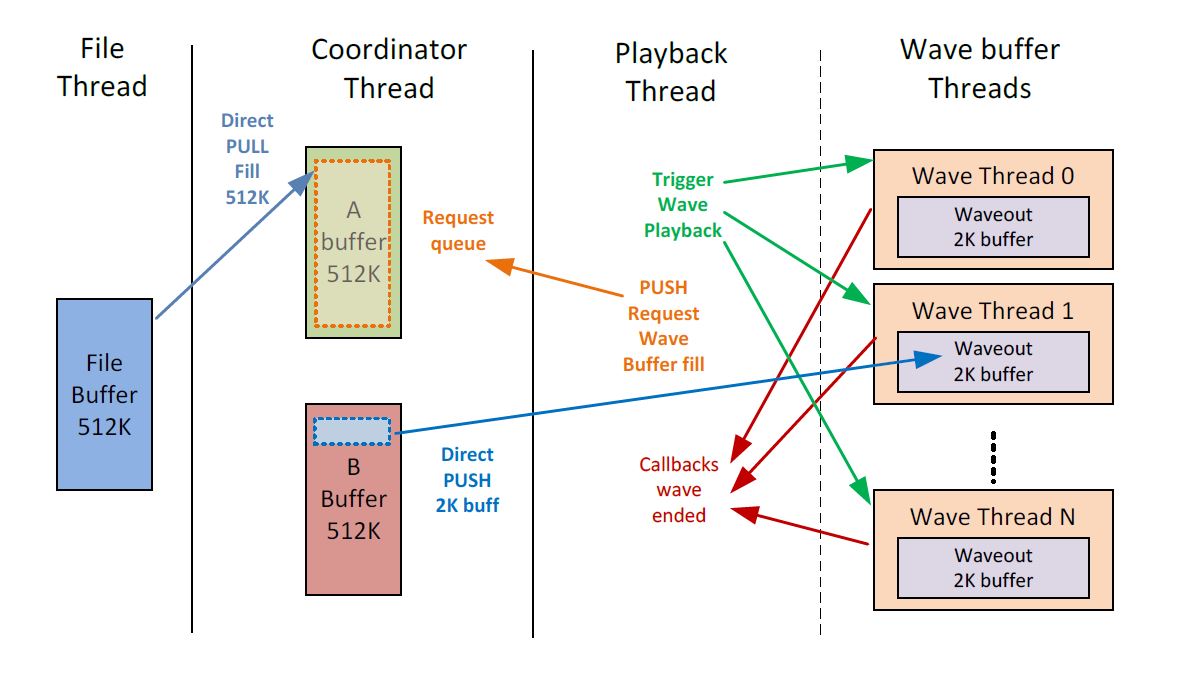
**charantejallada@gmail.com**

**Summary:** Playing familiar Jetsons theme song which is broke in to 23 individual pieces without any interruptions using Multithreading concepts. Each of 23 chunks are further split in to 2k buffer each, and all these chunks will be played simultaneously without any stuttering.

**What is used?**

* Simplified threading model using C++ 11.
* Multi-threading primitives.
* Fixed threads, Mutexes, Call backs, Futures, promises, async, conditional concepts.

**Architecture of project:**



**Description:**

* A total of 24 threads are used.

1. Main thread
2. File thread
3. Co-Ordinator thread
4. Playback thread
5. 20 Wave buffer threads

* Once created they stay alive for the duration of the application (until termination of the application.
* Source wave data is raw wave data.

1. Header less, just raw data.
2. Broken into 23 separate files (wave\_0.wav - wave\_22.wav).
3. Sizes of files vary between 512K to 128K bytes.
4. Sample rate: 22050, PCM, 16-bit, stereo wav file format.

**File thread:**

* Single Buffer - 512K
* Only one dynamic buffer allocation.
* Some type of flag, living in the thread to communicated that it is filled
* Need a mutex for access
* Refill buffer with SYNCHRONOUS file loads such as fopen(), fread()

**Coordinator Thread:**

* Double buffer
* Two separate buffers each 512K, dynamically allocated in beginning once
* Two major roles of the coordinator thread:

1. Retrieves data from the File thread, places in the A/B buffer
2. Push data mode
3. Copies data from A/B to the wave buffer thread

**Playback thread:**

* This thread keeps music playing between all of its wave buffers.
* When a Wave out buffer is done playing 1. It sends a callback to the playback thread.
* This thread then points to the next Wave out buffer and continues playing.
* The playback thread cycles indefinitely between all the Wave out buffers in a continuous circular buffer fashion.
* Futures, promises, async, conditional concepts are used for synchronization.