Audio Rendering/Processing and Control Ubiquity

a solution built using the Faust dynamic compiler and JACK/NetJack

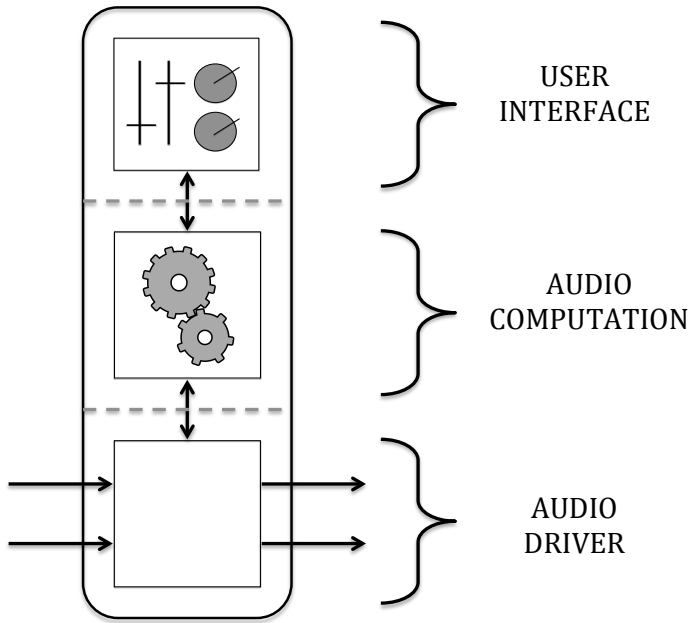
Stéphane Letz, Sarah Denoux, Yann Orlarey

journée FEEVER, 13 octobre 2014

The "control, compute, communicate" model

Separation of concern

- A typical audio processing application can be separated in three parts: *control*, *compute*, *communicate* (with the audio card)



- The three different parts can be deployed on different machines or control devices

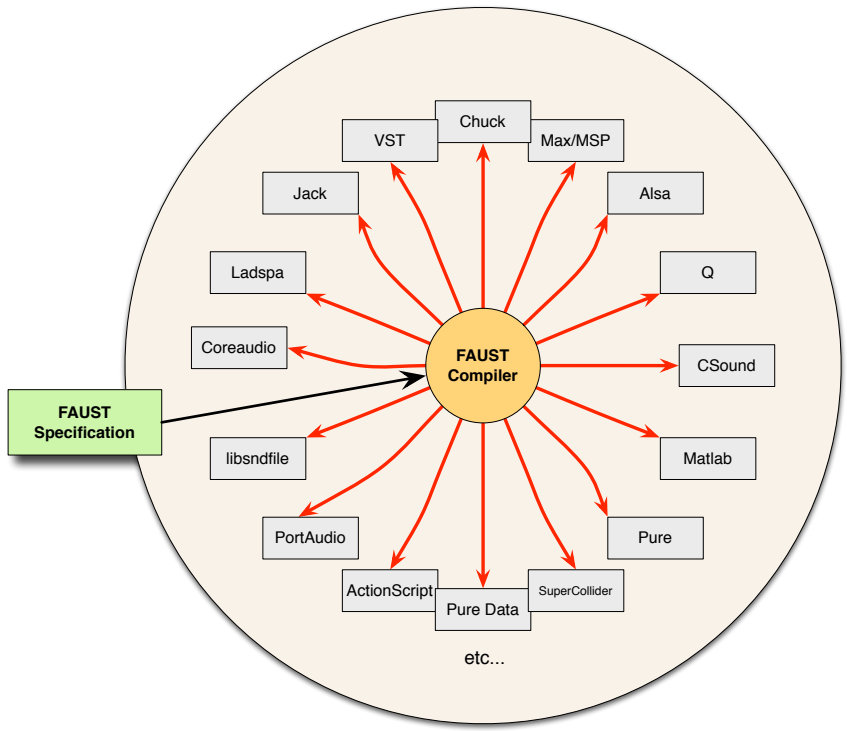
Using Faust for DSP processing, JACK/NetJack for audio

How to develop/deploy audio applications following this scheme:

- using easily reusable an combinable libraries, to be deployed on machines, tablets, smartphones...
- distributed as open-source components : Faust as a Domain Specific Language for DSP processing ("compute"), JACK/NetJack for audio processing/rendering ("communicate")

Faust Language

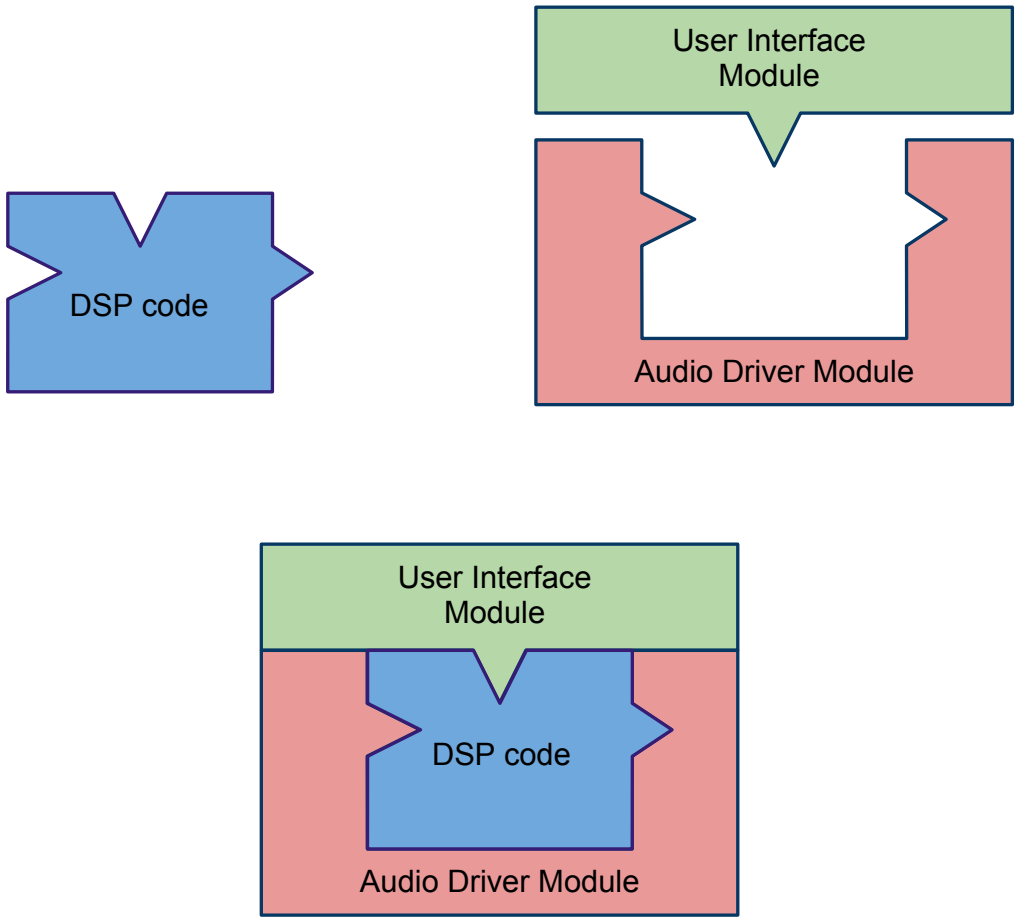
- Faust as a *Domain-Specific Language* for real-time signal processing and synthesis. A Faust program denotes a *signal processor*.
- Faust as a *high-level description language* and a *compiler*
- Faust as a flexible *multiple targets* DSP application/plug-ins deployment ecosystem



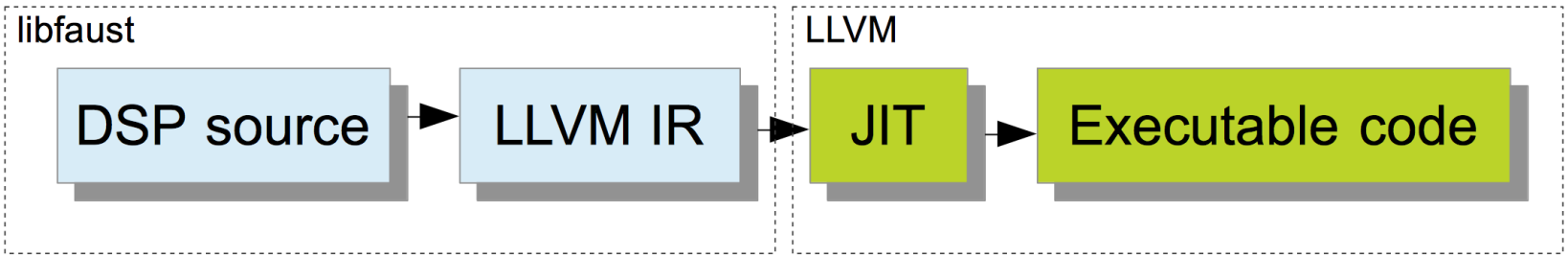
Faust architecture file system

Separation of concern

The *architecture file* describes how to connect the audio computation to the external world.



Embedding the dynamic compilation chain : *libfaust + LLVM*



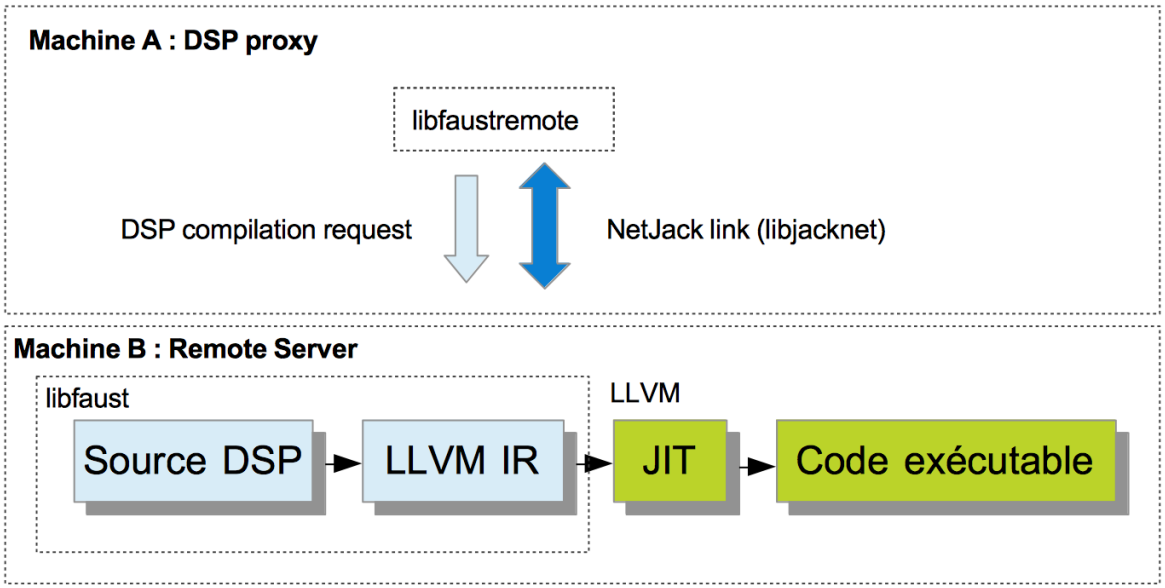
- the Faust compiler is now available as an embeddable library called *libfaust*
- a LLVM IR backend is added in Faust compiler
- linked with LLVM JIT libraries to produce native executable code in memory
- *createDSPFactory(...)*, *createDSPInstance(...)*

JACK/NetJack audio server system

JACK/NetJack system used to establish remote communication and compuation setup:

- NetJack is low-latency master/slave audio/MIDI communication protocol integrated in JACK2 as a set of standalone components (*netmanager*, *net backend*, *netadapter*)
- NetJack can also be used as a embeddable library called *libjacknet* (without having to run under the JACK server)
- *master* part of the protocol is triggered by the audio card
- *slave* part of the protocol will be synchronized (buffer size, sample rate...) with the master through the network link

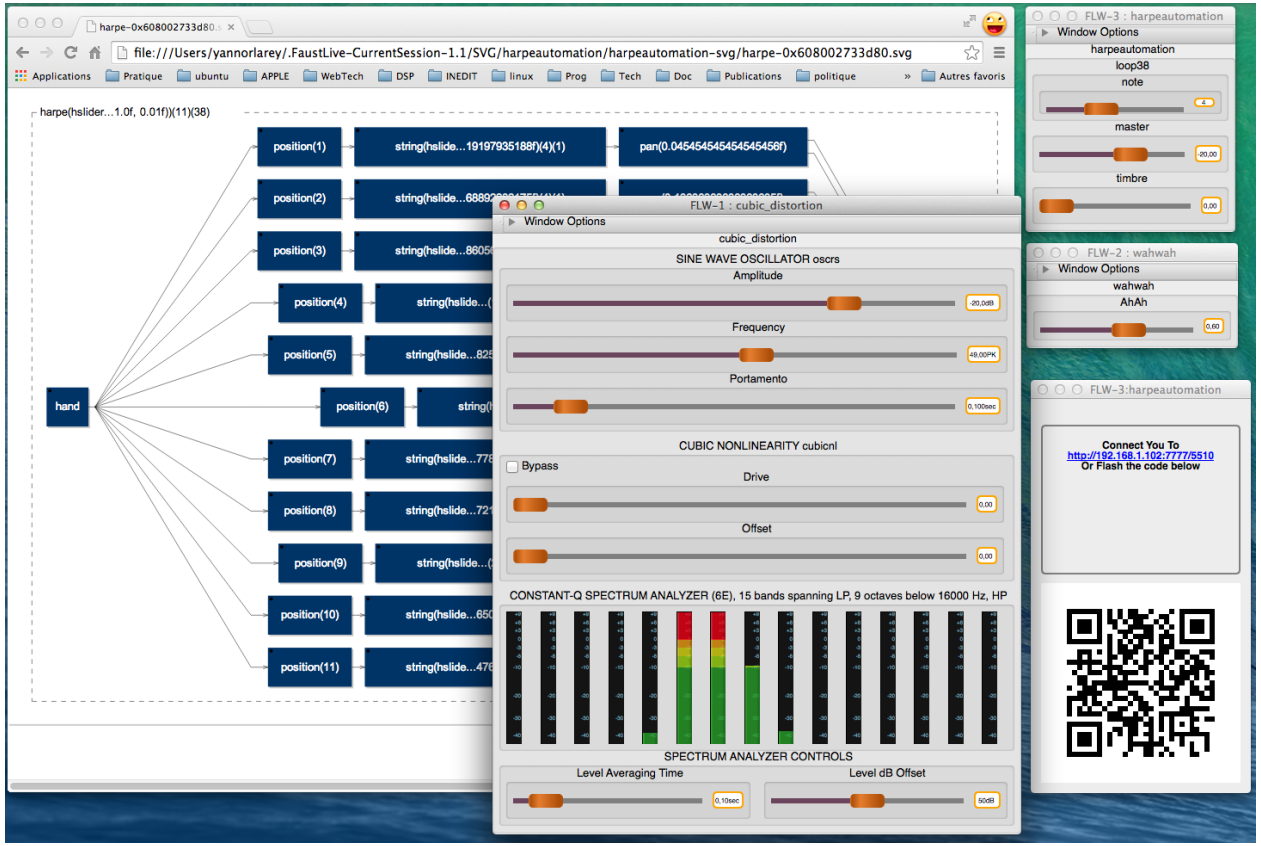
DSP proxy via remote DSP computation



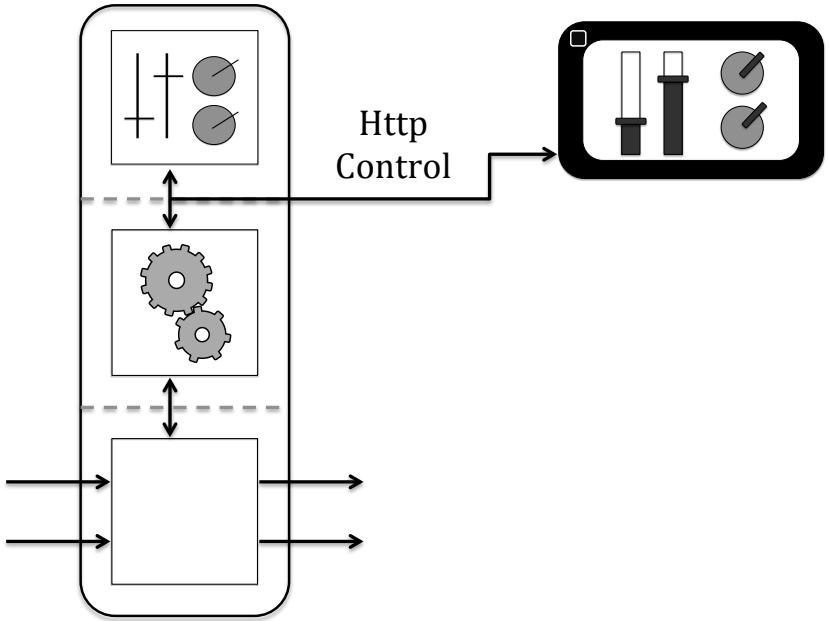
- Remote Server : waiting for compilation requests (Faust DSP source)
- server side : uses *libfaust* + LLVM for dynamic compilation
- server/client : audio + control link with *libjacknet*
- client side : *libfaustremote* proxy library with *createRemoteDSPFactory(...)*, *createRemoteDSPInstance(...)*

DEMO using *FaustLive* application

FaustLive (developed at GRAME by Sarah Denoux) aims to create a dynamic environment for Faust prototyping

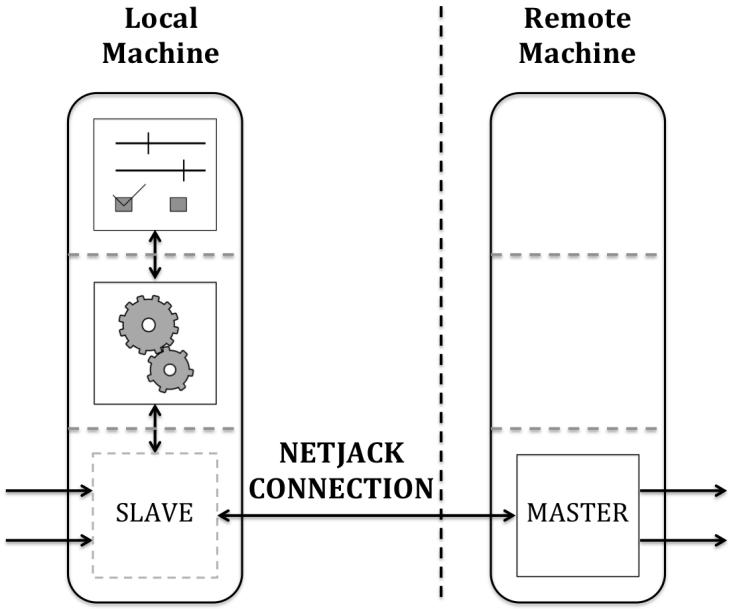


Remote control using OSC, HTTP, WebSocket...



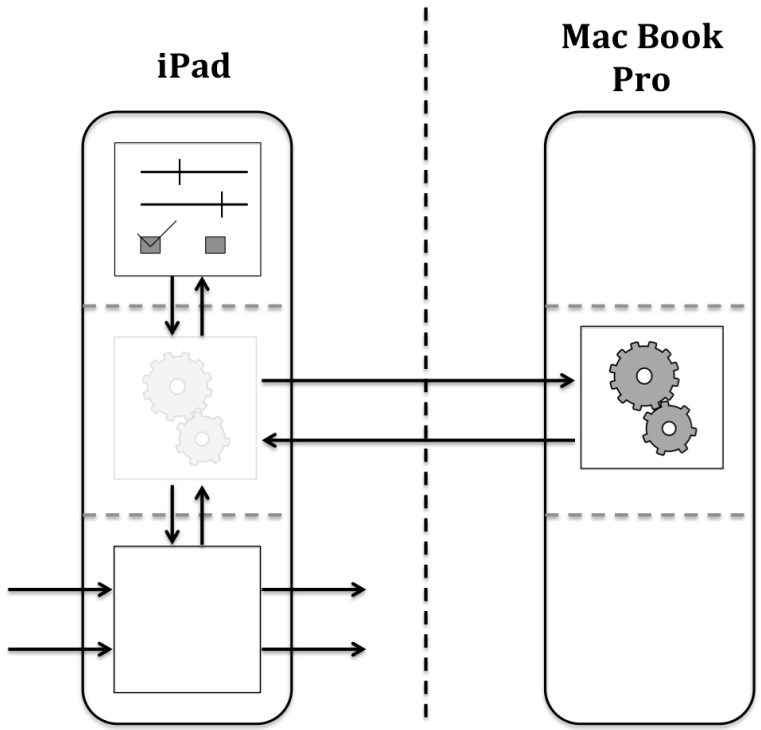
- the running application starts an embedded HTTP server (developed using *libmicrohttpd* library)
- the remote control code runs is delivered as an HTML page
- to be executed on a remote device

Remote communication using JACK/NetJack



- starting the JACK server on another machine, loaded with the *netmanager* master component
- in FaustLive, switching the audio driver to NetJack (*slave* mode)

Remote computation using libfaustremote



- quickly experiment Faust applications that use accelerometers
- on iOS where embedded dynamic compilers are not allowed...

Application final export using the FaustWeb compilation service

FaustWeb offers a multiple targets compilation service :

- list of available targets is returned by the server
- then the DSP code is sent and compiled on the server, and delivered back as a binary
- directly usable on the iPad as an autonomous application

Software components for the Control, Compute, Communicate model

Part of the Faust project :

- *libfaust* : library version of the Faust compiler
- *libfaustremote* : proxy access to remote computation server code
- Look at **http://sourceforge.net/projects/faudiostream/** on the faust2 branch

Part of the JACK2 project :

- *libjacknet* : library version of the master/slave NetJack protocol
- Look at **https://github.com/jackaudio/jack2**