**Partitioned Convolution**

**For Real-Time Audio Effect Processing**

Eric Battenberg, David Wessel, Juan Colmenares

\[ \text{ericb@eecs.berkeley.edu, wessel@cnmat.berkeley.edu, juancol@eecs.berkeley.edu} \]

---

**The Application**
- *First real-time app in the Par Lab.*
- Partitioned Convolution – an efficient way to do low-latency filtering with a long impulse response.
- Used in convolution reverb for environment simulation, creative effect processing, and electronic instrument creation.

**Uniform Partitioning**
- To trade off between complexity and latency we can split the filter into smaller delayed parallel filters.
- This reduces latency to \( N \), but increases the number of FFTs we must compute.

**Frequency Delay Line (FDL)**
- With Uniform Partitioning, we can exploit linearity of FFT's by moving them outside of the parallel delay line.
- In this "FDL", we only have to compute one FFT/IFFT per iteration.

**Multiple FDL’s**
- With a long filter and a small block size, we may end up with hundreds of partitions in an FDL.
- To cover more samples per partition, we can run larger FDL's in parallel.
- This presents us with a tuning problem: *What is the best set of FDL's for a particular filter length and latency?*

**Auto-Tuning for Real-Time**
- We are not only trying to maximize throughput.
- We want to improve the validity of real-time guarantees.
- For now, we estimate a Worst-Case Execution Time (WCET) then combine FDL's that are most likely to meet their deadlines (vertical lines in scheduling diagram).
- We will progress to more robust scheduling as a next step.

---

**Convolution**
- Convolution is a way to do linear filtering using an FIR (finite impulse response).
- Filter length, \( L \), can be \( > 100,000 \) (3 sec)
- Direct Convolution:
  - \( O(L) \) complexity
  - Zero delay
- Block FFT Convolution:
  - \( O(\log(L)) \) complexity
  - \( L \) delay
- We would like the delay to be less than 512 samples (10ms)

---

**Mathematics**

- \( y[n] = \sum h[k]x[n-k] \)
- \( y = \) output, \( x = \) input, \( h = \) filter

---

**Diagram**

- ![Diagram of Partitioned Convolution](image)