Experiences with using GNU Radio for Real-time Wireless Signal Classification

Christopher Becker (University of Utah)
Aniqua Baset (University of Utah)
Sneha Kasera (University of Utah)
Kurt Derr (Idaho National Laboratory)
Samuel Ramirez (Idaho National Laboratory)
Outline

- Introduction
- System Overview
- Challenges and Approaches
- Setup
- Evaluation
- Conclusion
Outline

- Introduction
- System Overview
- Challenges and Approaches
- Setup
- Evaluation
- Conclusion
Real-time Wireless Monitoring in a Variety of Environments

- High-security/control system
  - Power plants
  - Military bases
  - Water treatment plants

- Shared spectrum
  - 3.5 GHz band
Motivation

<table>
<thead>
<tr>
<th>Safety</th>
<th>Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Interference Identification and Mitigation</td>
</tr>
</tbody>
</table>
Outline

➢ Introduction
➢ System Overview
➢ Challenges and Approaches
➢ Setup
➢ Evaluation
➢ Conclusion
System Architecture

Note: For Reference Only; part of a paper under submission
Energy-based Classifier (EBC)

Feedback: cut-off multiplier, m

Signal Classifier 1
- mean of lowest n values, μ
- std of lowest n values, σ
- cut-off = μ + mσ
- Peak Detector
- Bandwidth Analyzer
- Pattern Matcher

Signal Classifier 2

Signal Classifier n

Signal Classifier
- One-class SVM
- Multi-class SVM

Machine Learning-based Classifier (MLBC)

Data Reducer
- 1 in C chunks
- C is determined based on Usage Control
- Feature Set Calculator (SCF and α-profile)

Usage Control

Performance Monitor

Note: For Reference Only; part of a paper under submission
Outline

➢ Introduction
➢ System Overview
➢ Challenges and Approaches
➢ Setup
➢ Evaluation
➢ Conclusion
Message Passing

• Challenges
  – Limited message rate
  – Data encapsulation/unencapsulation expensive

• Approaches
  – Fixed-size formatted data strings
  – Qt Signals and Slots/Message Passing Wrappers
Message Passing Wrapper

1. Receive Input
2. Process the Input
3. Prepare the Output Data
4. Generate Message (if needed)
5. Forward the Data
6. Convert to Internal Format (if needed)
7. Update Internal Data
8. Request for Data
9. Forward Request
10. Prepare and Return Data
11. Send Output
Overflow and Dropped Packet Detection

• Challenges
  – Run-time performance tuning
  – Timestamps set for recovery from Overflow/Dropped Packet
  – UHD driver sends information to stderr output (not accessible by GNU Radio API)

• Approach
  – Separate program to monitor performance
    • Monitor stderr for overflow/dropped packet indicators
    • Collect information from operating system
    • Send adjustment information to rest of our system
Non-optimized Block Implementations

- **Challenge**
  - Some of the standard blocks not always optimized correctly (at least for our use cases)
    - Examples: add_cc, Log Power FFT, analog_const_source

- **Approaches**
  - add_cc and Log Power FFT: use Vector Optimized Library of Kernels (VOLK) to improve speed
  - Log Power FFT and analog_const_source: change interface/internal code to add capabilities and/or reduce overhead for our specific use case
Dynamic Decimation

• Challenge:
  – Dynamically change decimation rates at run-time
    • Pausing and reconfiguring our flowgraph causes errors due to buffer size issues

• Approach
  – Delete and re-initialize the immediately preceding blocks to have buffers re-initialize correctly
Scheduler

- Challenge
  - GNU Radio scheduler thread can become a bottleneck for performance when a large number of blocks (threads) are used
  - Still an open problem

- Current Approach
  - Limit the number of threads
    - Combine functionality of multiple blocks into new blocks
    - Allow more inputs to a single block
GPU Acceleration

• Challenges
  – Limited data buffer sizes
    • Does not lend itself to GPU Acceleration due to memory copying overhead
    • See also: Hitefield & Clancy GRCon 2016, Piscopo GRCon 2017

• Approach
  – Use VOLK whenever possible
  – Use other processing-optimization techniques
    • e.g., heaps and lookup tables
Outline

- Introduction
- System Overview
- Challenges and Approaches
- Setup
- Evaluation
- Conclusion
Laptop Configurations

<table>
<thead>
<tr>
<th></th>
<th>Development and Testing</th>
<th>Testing and Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operating System</strong></td>
<td>Ubuntu 14.04.1</td>
<td>Ubuntu 18.04</td>
</tr>
<tr>
<td><strong>UHD Version</strong></td>
<td>3.11.0</td>
<td>3.11.1</td>
</tr>
<tr>
<td><strong>GNU Radio Version</strong></td>
<td>3.7.11</td>
<td>3.7.12</td>
</tr>
<tr>
<td><strong>VOLK Version</strong></td>
<td>1.3.0</td>
<td>1.4.0</td>
</tr>
</tbody>
</table>

- Dell M4800 Laptops with increased RAM
  - 2.8 GHz 4-core 64-bit Intel i7-4810MQ processors
  - 32 GB of RAM
- Build GNU Radio script (slightly modified for Ubuntu 18.04) used for installing software
### X310 Configurations

<table>
<thead>
<tr>
<th></th>
<th>Development and Testing</th>
<th>Testing and Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hardware Revision</strong></td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td><strong>Firmware Version</strong></td>
<td>5.1</td>
<td>6</td>
</tr>
<tr>
<td><strong>FPGA Version</strong></td>
<td>33</td>
<td>35</td>
</tr>
</tbody>
</table>

- 1 Gbps Ethernet Link
  - 25 MHz sample rate
Outline

➢ Introduction
➢ System Overview
➢ Challenges and Approaches
➢ Setup
➢ Evaluation
➢ Conclusion
Message Passing

• Used wrapped sender and receiver blocks
• 4096 (long) integers via either GNU Radio messages or Qt Signals and Slots
• Ran 100 times

<table>
<thead>
<tr>
<th></th>
<th>Median Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNU Radio Messages</td>
<td>0.26392602</td>
</tr>
<tr>
<td>Qt Signals and Slots</td>
<td>0.00337398</td>
</tr>
</tbody>
</table>

> 78x performance improvement
Non-optimized Implementations

- 4096 Samples
- Ran 100 tests

<table>
<thead>
<tr>
<th></th>
<th>Median Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Complex Add</td>
<td>0.00143647</td>
</tr>
<tr>
<td>VOLK Complex Add</td>
<td>0.00141096</td>
</tr>
<tr>
<td>Log Power FFT</td>
<td>0.01590848</td>
</tr>
<tr>
<td>PSD</td>
<td>0.01587605</td>
</tr>
</tbody>
</table>
Combining Blocks/Adding Inputs (Setup 1)

- Separate
- Combined
- Somewhat Combined
Combining Blocks/Adding Inputs (Results 1)
Combining Blocks/Adding Inputs (Setup 2)

Separate

Combined
Combining Blocks/Adding Inputs (Results 2)
Outline

➢ Introduction
➢ System Overview
➢ Challenges and Approaches
➢ Setup
➢ Evaluation
➢ Conclusion
Conclusion

• Many challenges for using GNU Radio in real-time spectrum monitoring applications
• Many ways to approach these challenges
  – Some provide large performance improvements
  – No one-size fits all solution, however