Big Software for SmallSats: Adapting CFS to CubeSat Missions

Alan Cudmore, Gary Crum, Salman Sheikh, James Marshall
NASA Goddard Space Flight Center
[alan.p.cudmore, gary.a.crum, salman.i.sheikh, james.marshall-1]@nasa.gov
Outline

• Motivation
• What is CFS?
• Experience: CSP / CeREs
• Experience: Dellingr
• Performance
• Future Work
• References

• NOTE: All images courtesy of NASA
Motivation

• Expanding requirements
  – Science
  – Risk tolerance
  – This stresses software (and teams!)

• Budgets are not expanding

• “Small” Satellite does not mean “small” software

• Solution: a trusted framework with reusable components
CFS: Core Flight Software

• NASA recognized a need to move away from “Clone and Own”
• Developed to tackle the very issues that SmallSats now face
• Framework and core services (cFE)
• Common set of applications and libraries
• (McComas, 2012) (Fesq, Dvorak, 2012)

“At Goddard the main driver for changing the development process is cost, [...] An obvious way to reduce cost and schedule is to increase the amount of software reuse.” (Wilmot, 2006)

The CFS follows a product line approach with the goal to support systematic reuse.

(Ganesan, Lindvall, Ackermann, McComas, Bartholomew, 2009)
Framework and Core Services (cFE)

- Layered architecture
- Supports Publish / Subscribe Applications
- Events
- Tables
- Time

CFS Applications / Libraries
- CF
- CS
- DS
- FM
- HS
- HK
- LC
- MD
- MM
- SBN
- SC
- SCH

User Applications / Libraries
- ?

cFE Services
- Exec
- Event
- Bus
- Table
- Time

Operating System Abstraction Layer (OSAL)

Operating System (Linux, RTEMS, VxWorks, FreeRTOS)
Libraries and Applications

- Currently 12 Applications are available (http://cfs.gsfc.nasa.gov/)
- Optional, depends on mission needs.
- Easy to create
  - Sample application demonstrates messaging, events, and application loop
Heritage

- cFE:
  - Lunar Reconnaissance Orbiter
  - Living With a Star / Radiation Belt Storm Probes

- CFS
  - Global Precipitation Measurement
  - Magnetospheric MultiScale
  - Lunar Atmosphere and Dust Environment Explorer
CHREC Space Processor

- Space Test Program, Houston 5 / ISS SpaceCube Experiment Mini
- CHREC Space Processor Experiment
- NSF Center for High-Performance Reconfigurable Computing
- Presented here last year (Rudolph et al, 2014)
- Xilinx Zynq 7020
  - Arm Dual Core Cortex A9
  - Artix-7 FPGA
- Runs CFS!
CFS on the CHREC Space Processor

- Work spread over 3 employees
- Created 11 custom applications / libraries
- Code is in well defined applications
- Vary in level of reusability
- This is in addition to existing CFS functionality

<table>
<thead>
<tr>
<th>Mission Specific</th>
<th>Reusable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSP Health</td>
<td>Image Processing</td>
</tr>
<tr>
<td>FTDP</td>
<td>Telemetry Out</td>
</tr>
<tr>
<td>Injest</td>
<td>Camera Control</td>
</tr>
<tr>
<td>Comm Library</td>
<td>Self Timer</td>
</tr>
<tr>
<td>FTDP Send</td>
<td>Command Injest</td>
</tr>
<tr>
<td>FTDP Receive</td>
<td>File Transfer</td>
</tr>
<tr>
<td>FTDP</td>
<td>Comm Library</td>
</tr>
</tbody>
</table>

National Aeronautics and Space Administration
CeREs

- Compact Radiation Belt Explorer
- MERiT: Miniaturized Electron and pRoton Telescope
- Flight computer is a CSP
- CFS used for flight software
- (Kanekal, 2014)
Dellingr

- **Hardware:**
  - ARM7 processor (40 Mhz 2Mb RAM)
  - Reaction Wheels
  - Magnetorquers
  - Sensors (FSS)

- **Science**
  - INMS
  - Magnetometer
  - Thermal Louvre
Dellingr and CFS

- Work spread over three employees
- Ported OSAL to FreeRTOS
- Integrate with GomSpace software
- Custom
  - Hardware Library
  - Hardware telemetry
  - Radio
  - ACS
  - Science instruments
- Generated using David A. Wheeler's 'SLOCCount'

Custom Code for Dellingr Approx. 10k SLOC

- ACS
- HW Lib
- Radio
- INMS
- GPS
- SHK
- RW
- MAG
- Camera

CFS for CubeSats
Performance

- CFS imposes some performance costs
- Compared build with just FreeRTOS vs CFS
- Code available: https://github.com/jcmarsh/cpek

<table>
<thead>
<tr>
<th></th>
<th>FreeRTOS</th>
<th>CFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whetstone (KWIPs)</td>
<td>10426.4</td>
<td>4448.2</td>
</tr>
<tr>
<td>Dhrystone (per second)</td>
<td>798.6</td>
<td>330.8</td>
</tr>
<tr>
<td>Hardware ping (per second)</td>
<td>753.5</td>
<td>489.7</td>
</tr>
<tr>
<td>Application ping (per second)</td>
<td>1481.9</td>
<td>285.7</td>
</tr>
</tbody>
</table>
Future Work


• CFS SDK

• Man Rated
Summary

- CFS is a mature framework
  - Strong heritage
  - Reduces personnel requirements
  - Available on a variety of platforms
  - Well suited to CubeSat missions
- Already being used on NASA CubeSats
References


