



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: Dellinger
 - 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.”

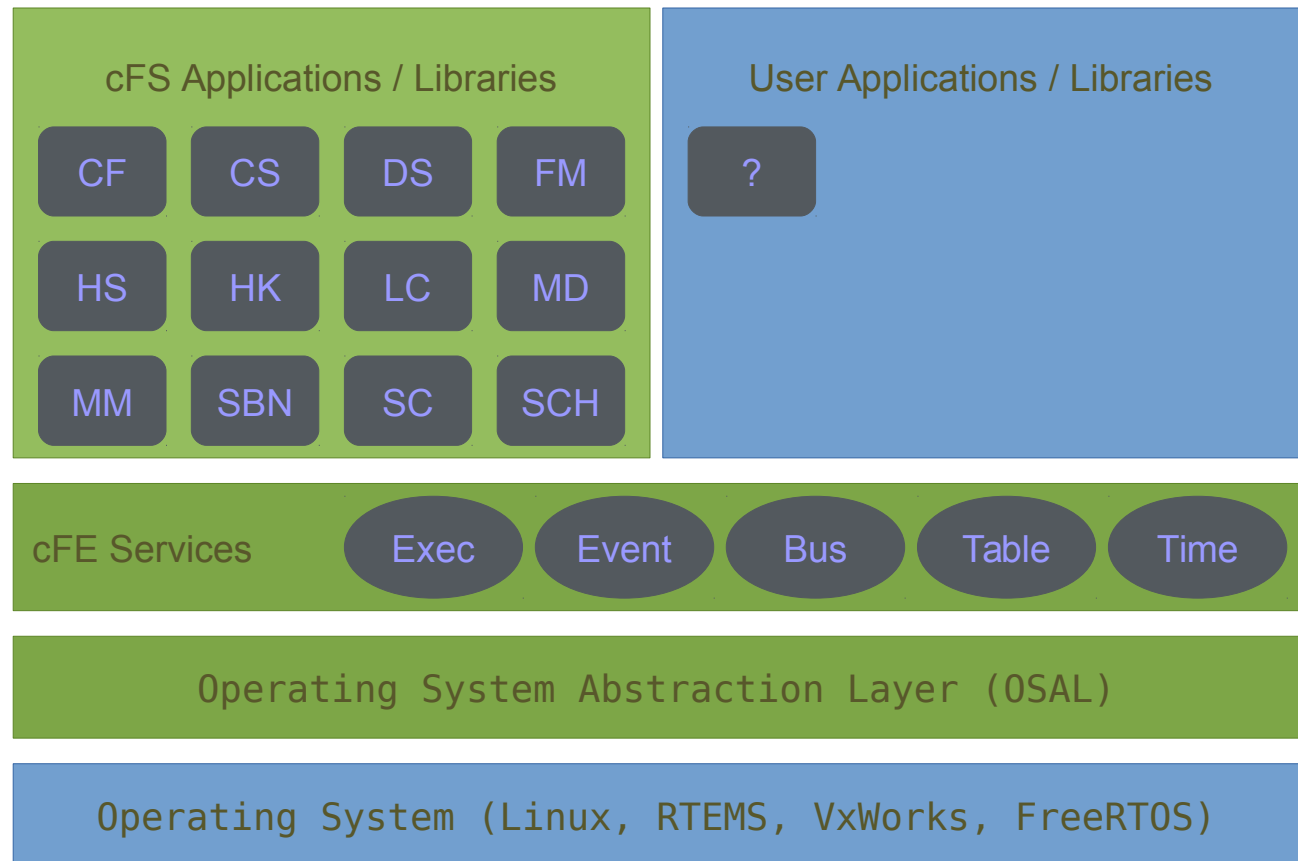
(Wilmot, 2006)

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



Libraries and Applications

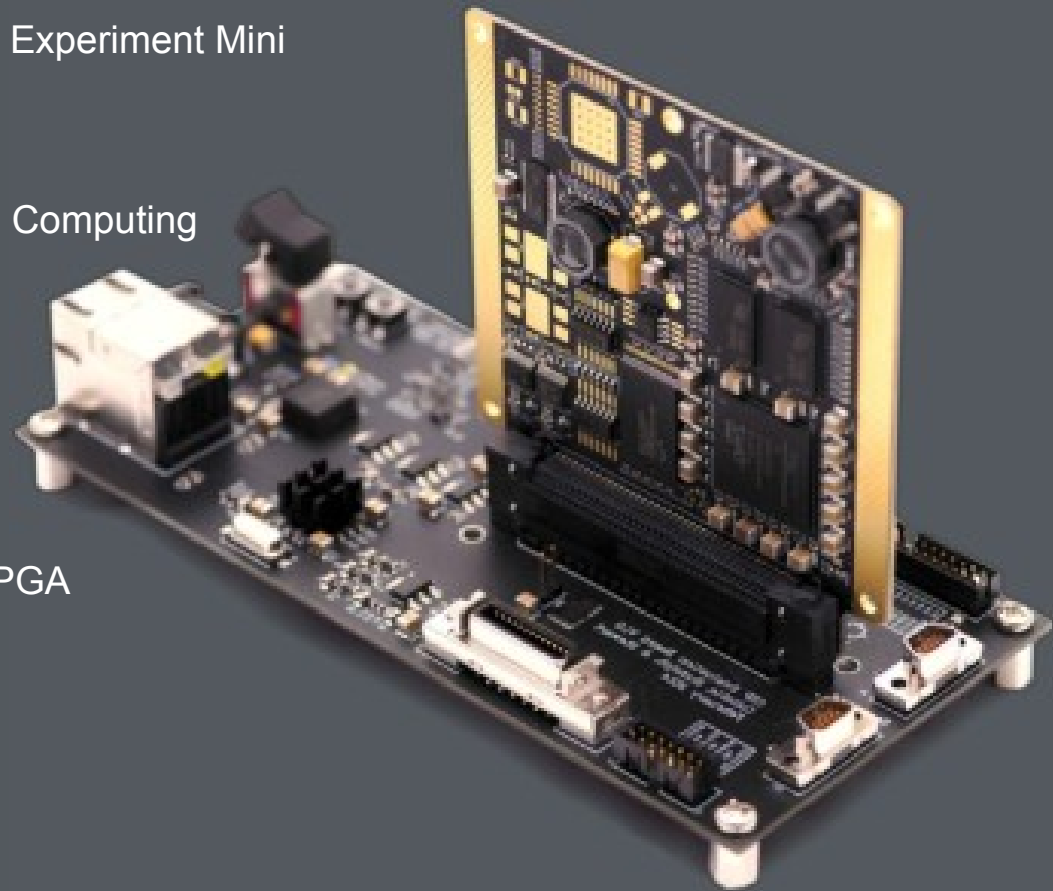
- Currently 12 Applications are available (<http://cfs.gsfc.nasa.gov/>)
- Optional, depends on mission needs.
- Easy to create
 - Sample application demonstrates messaging, events, and application loop

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

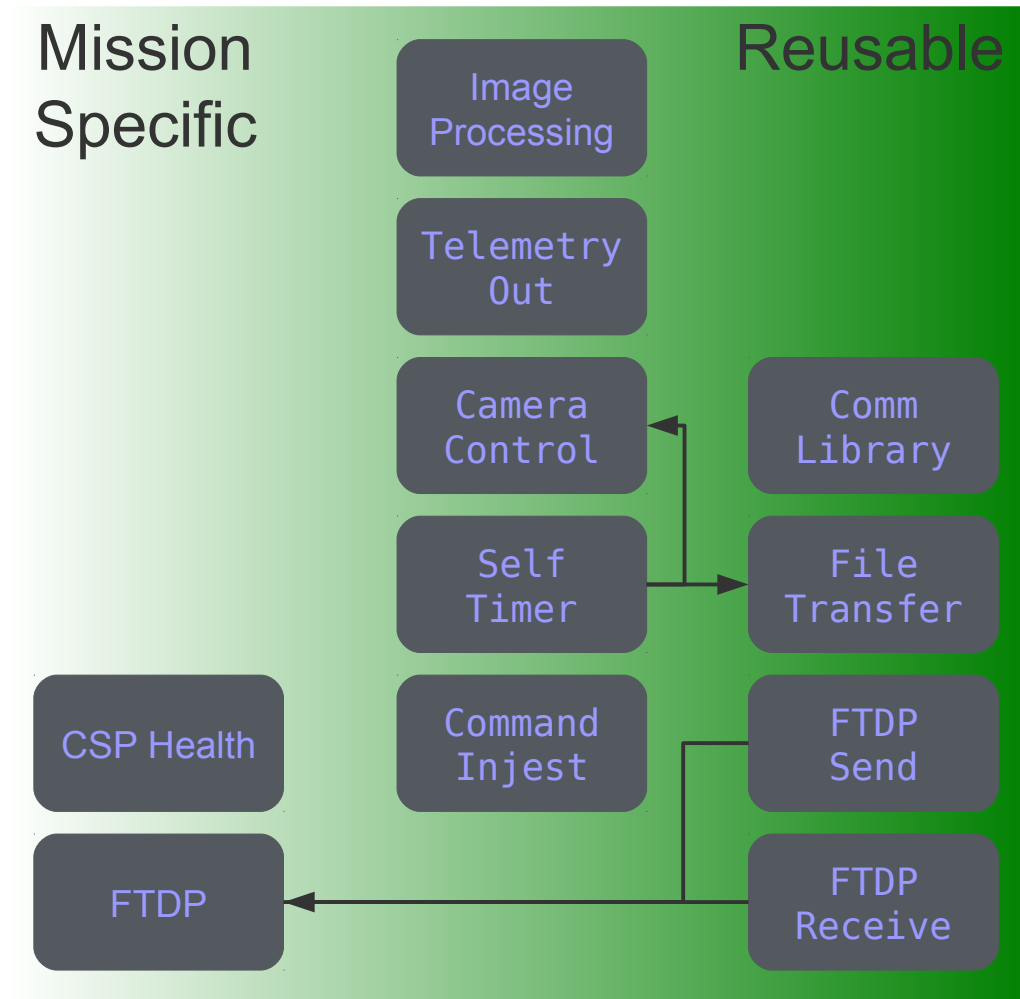
CHREC Space Processor

- 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)
- Two CSPv1 in tandem
 - Xilinx Zynq 7020
 - Arm Dual Core Cortex A9 and Artix-7 FPGA
- Runs cFS!
- Launch 2016



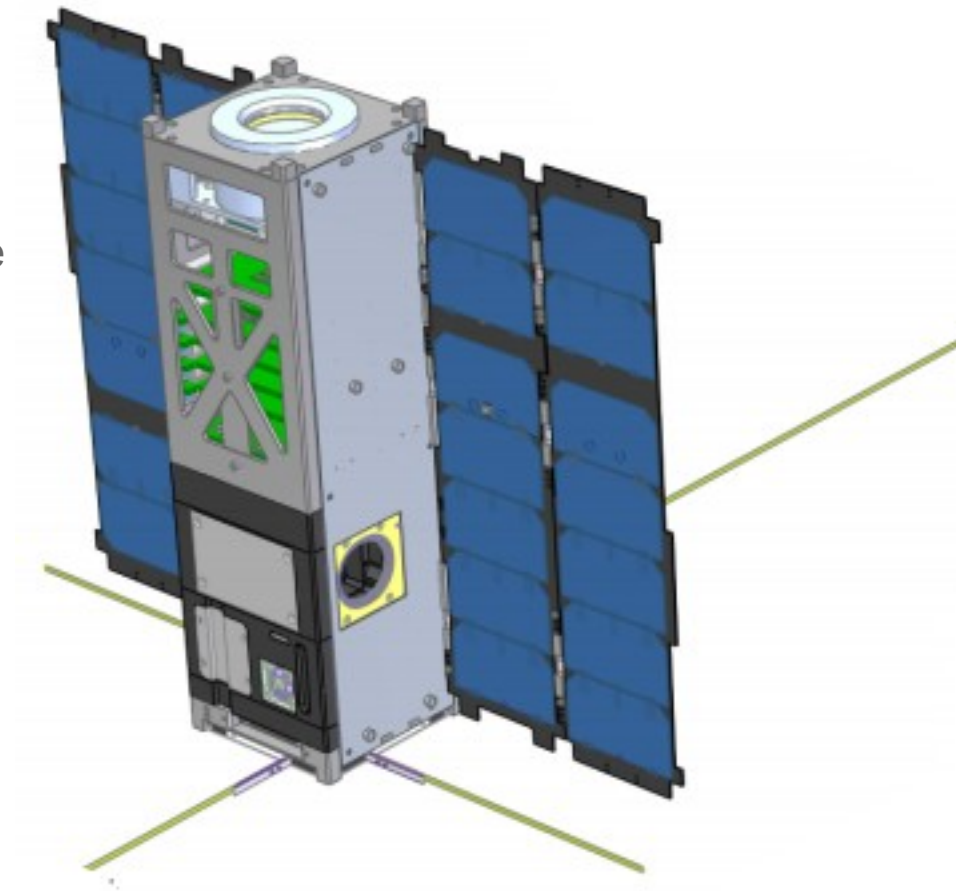
cFS on the CHREC Space Processor

- 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



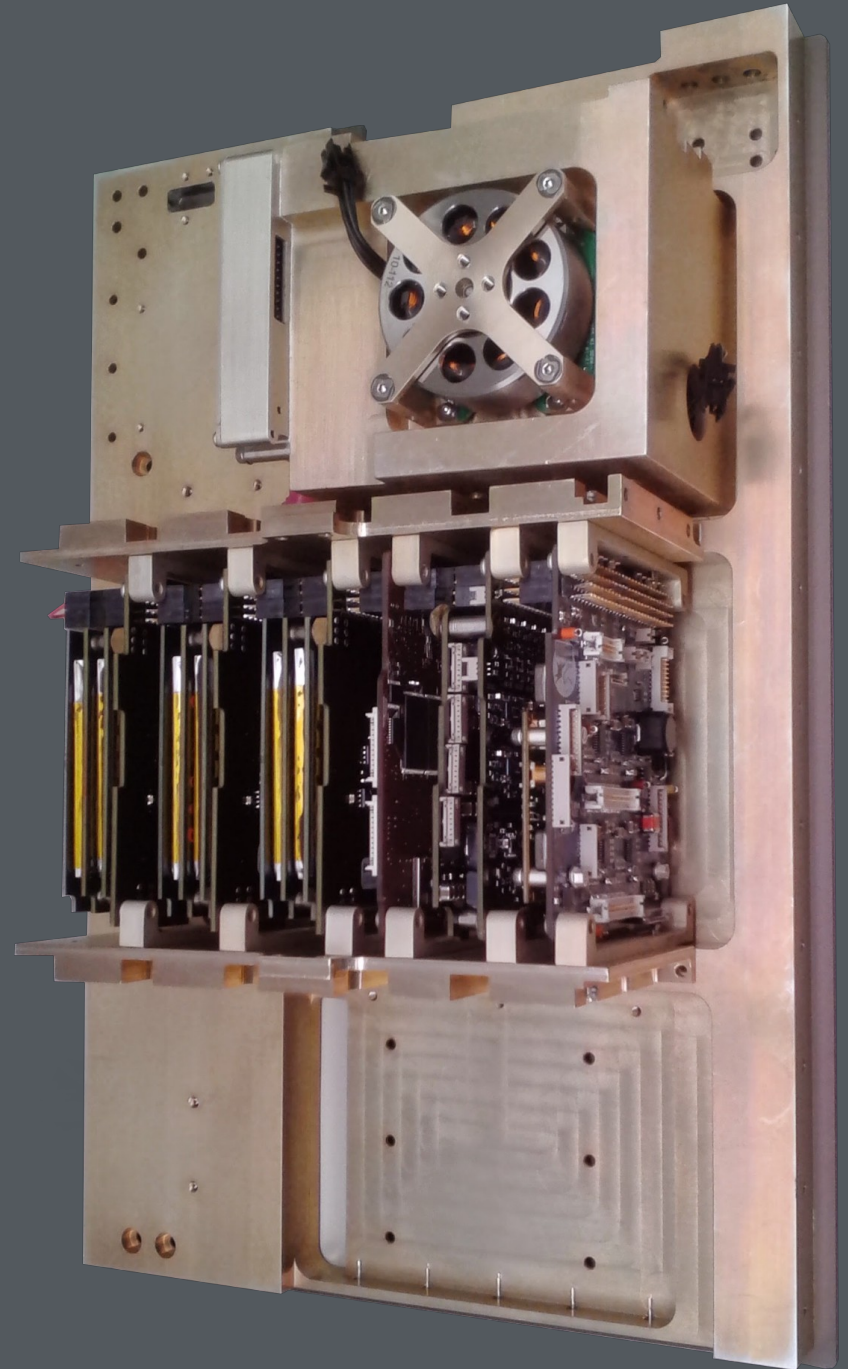
CeREs

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



Dellingr

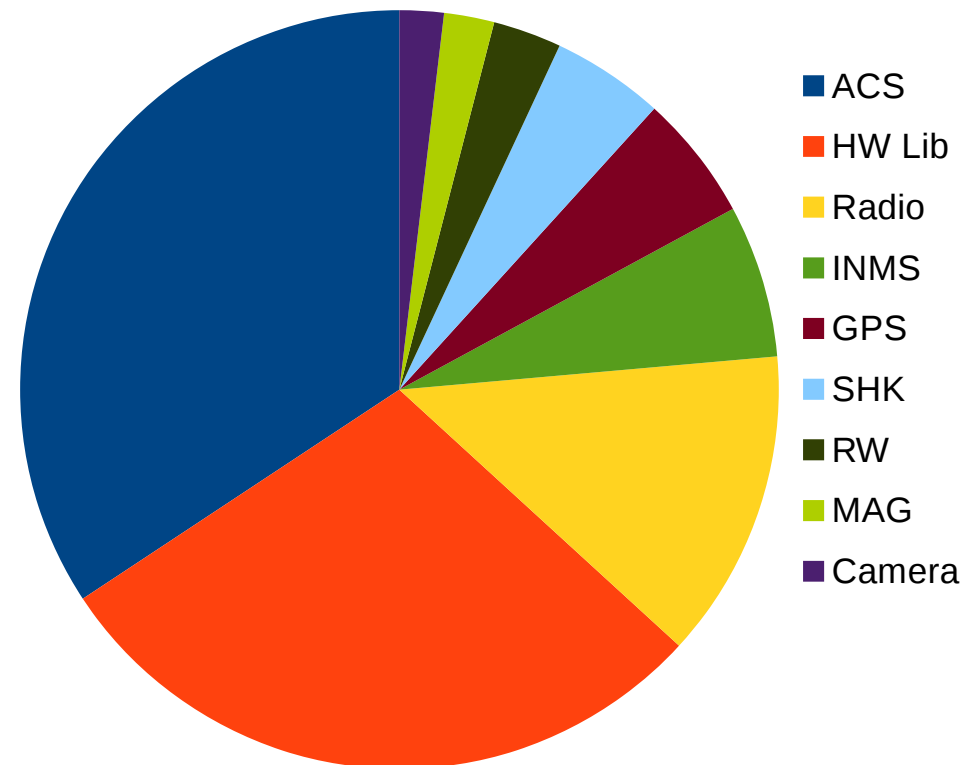
- Hardware:
 - ARM7 processor (40 Mhz 2Mb RAM)
 - Reaction Wheels
 - Magnetorquers
 - Sensors (FSS)
- Science
 - INMS
 - Magnetometer
 - Thermal Louvre



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



Performance

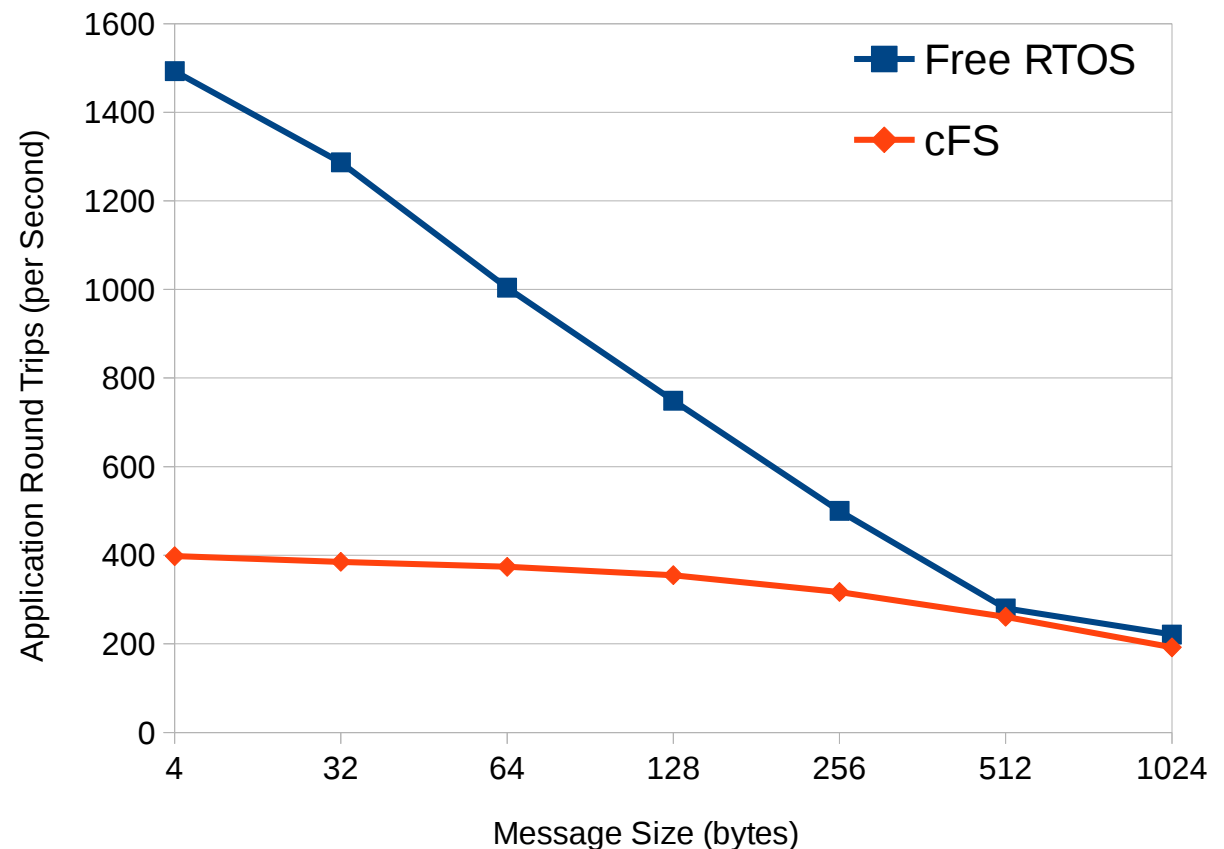
- cFS imposes some performance costs
- Compared build with just FreeRTOS vs cFS
- Code available: <https://github.com/jcmarsh/cpek>

	FreeRTOS	CFS
Dhrystone (per second)	11300.7	10576.4
Whetstone (kWIPs)	865.7	852.1
Hardware ping (per second)	757	621

Application Communication Costs

- cFS supports publish / subscribe message passing through the software bus.
- Adds functionality to FreeRTOS queues, increases overhead.
- Chart shows round trip messages passed between two applications.

FreeRTOS vs cFS Message Passing

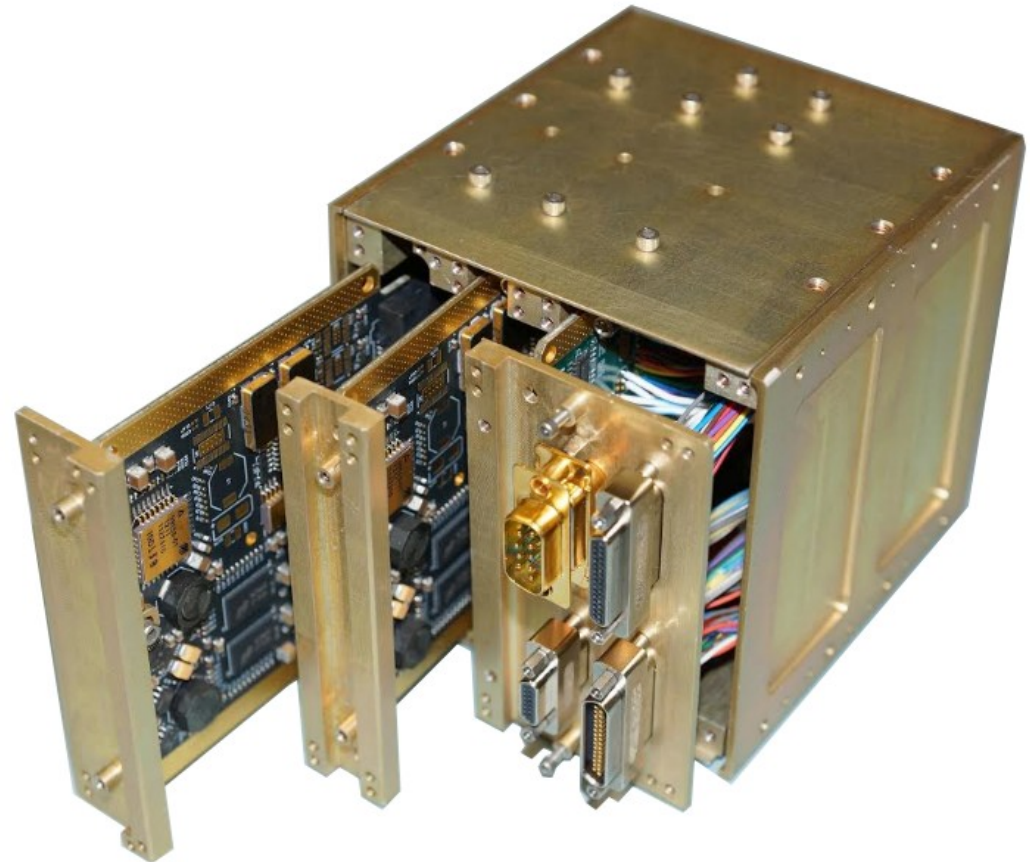


Future Work

- 42 Simulator integration: <http://fortytwospacecraftsimulation.sourceforge.net/>
- 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
- Open Source (<http://cfs.gsfc.nasa.gov/>)
- Already being used on NASA CubeSats



References

- J. Wilmot, “Implications of responsive space on the flight software architecture,” in Proceedings of Responsive Space Conference. AIAA, 2006.
- D. Ganesan, M. Lindvall, C. Ackermann, D. McComas, and M. Bartholomew, “Verifying architectural design rules of the flight software product line,” in Proceedings of the 13th International Software Product Line Conference. Carnegie Mellon University, 2009, pp. 161–170.
- Rudolph, D., Wilson, C., Stewart, J., Gauvin, P., George, A., Lam, H., Crum, G., Wirthlin, M., Wilson, A., Stoddard, A. “CSP: A Multifaceted Hybrid Architecture for Space Computing” Proceedings of the AIAA/USU Conference on Small Satellites, Advanced Technologies, SSC14-III-3. <http://digitalcommons.usu.edu/smallsat/2014/AdvTechI/3/>
- McComas, D. “NASA/GSFC’s Flight Software Core Flight System” Workshop on Spacecraft Flight Software, San Antonio, TX, 2012
- Fesq, L., Dvorak, D., “NASA’s Software Architecture Review Board’s (SARB) Findings from the Review of GSFC’s 'core Flight Executive/Core Flight Software' (cFE/CFS)” Workshop on Spacecraft Flight Software, San Antonio, TX, 2012
- Kanekal, S., “CeREs: a Compact Radiation bElt Explorer”, Proceedings of the AIAA/USU Conference on Small Satellites, Poster Session, SSC14. <http://digitalcommons.usu.edu/smallsat/2014/Poster/16/>

