Journey to the Core: Core Flight Executive at Lockheed Martin Space Systems Company

Matt Dean
Michael Phillips
Ryan Slabaugh

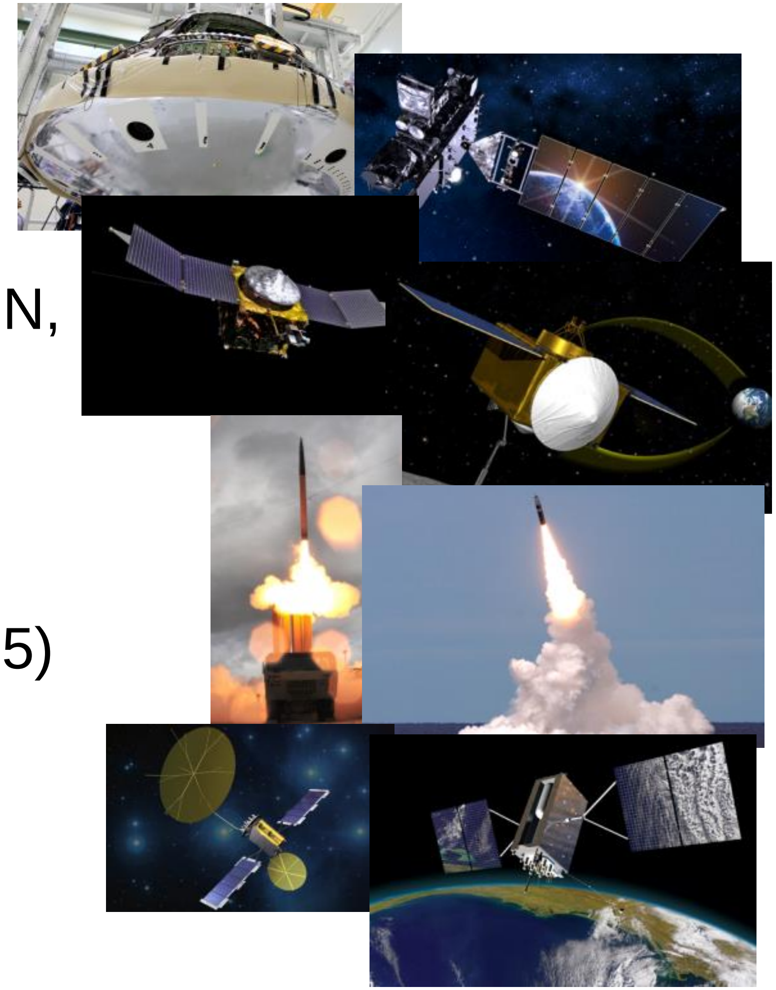
December 2014

Agenda

- Core Flight Executive (cFE) uses at LM
- Performance studies for cFE
- Software Development Environment

LMSSC Background

- Civil Space
  - Human Space Flight: Orion
  - Weather and Climate: GOES-R
  - Exploration: MRO, GRAIL, MAVEN, JUNO, InSIGHT, OSIRIS-REx
- Strategic Missile Defense
  - Missile Defense: THAAD
  - Fleet Ballistic Missile (Trident II D5)
- Military Space
  - MUOS, GPSIII, SBIRS
- Advanced Technology Center



$8.0B in sales for SSC in 2013

IRAD Study Projects

- 2012 RTI DDS Middleware feasibility studies
- 2013 FSW Design Pattern discovery studies (presented @ ‘13 WS)
- 2014 Project to explore migrating LMSSC from stove-piped, program-specific FSW development to enterprise-wide solution
  - Focus has been on Executive layers
  - LMSSC monitored cFE progress over last few years and began investigating in earnest in early 2014
  - Visited with GSFC and adopters of cFE (ARC and JHU-APL) for lessons learned
  - Adopted for use in real-time simulators, demonstration spacecraft FSW, and payload FSW in mid-2014
- 2015 Build on cFE and cFS to meet LMSSC needs

IRAD 2014: Middleware Performance Comparison

- **The task:**
  - Compare performance between real-time middleware capable of performing publish-subscribe tactic
  - Use HWIL testbeds and simulators to get relative performance metrics to inform decisions later
- **The players:**
  - cFE
    - Implemented by GSFC over decade of program sharing and re-use
    - Contains 5 most similar components used in spacecraft FSW, one of which is Software Bus publish-subscribe capability
    - Strengths: flight heritage, man-rated certification, wide adoption within NASA for a variety of purposes, operating system abstraction for easier porting
  - RTI DDS
    - RTI DDS is an implementation of OpenDSS standard from OMG
    - Successfully used in several places within LM (not LMSSC)
    - Publish-subscribe tactic viewed as the enabling capability to reduce integration duration, complexity, and cost
    - Strengths: implemented quality of service settings, record and playback tools,
- **The metrics:**
  - Mainly interested in characterizing delay introduced by sophisticated IPC methods

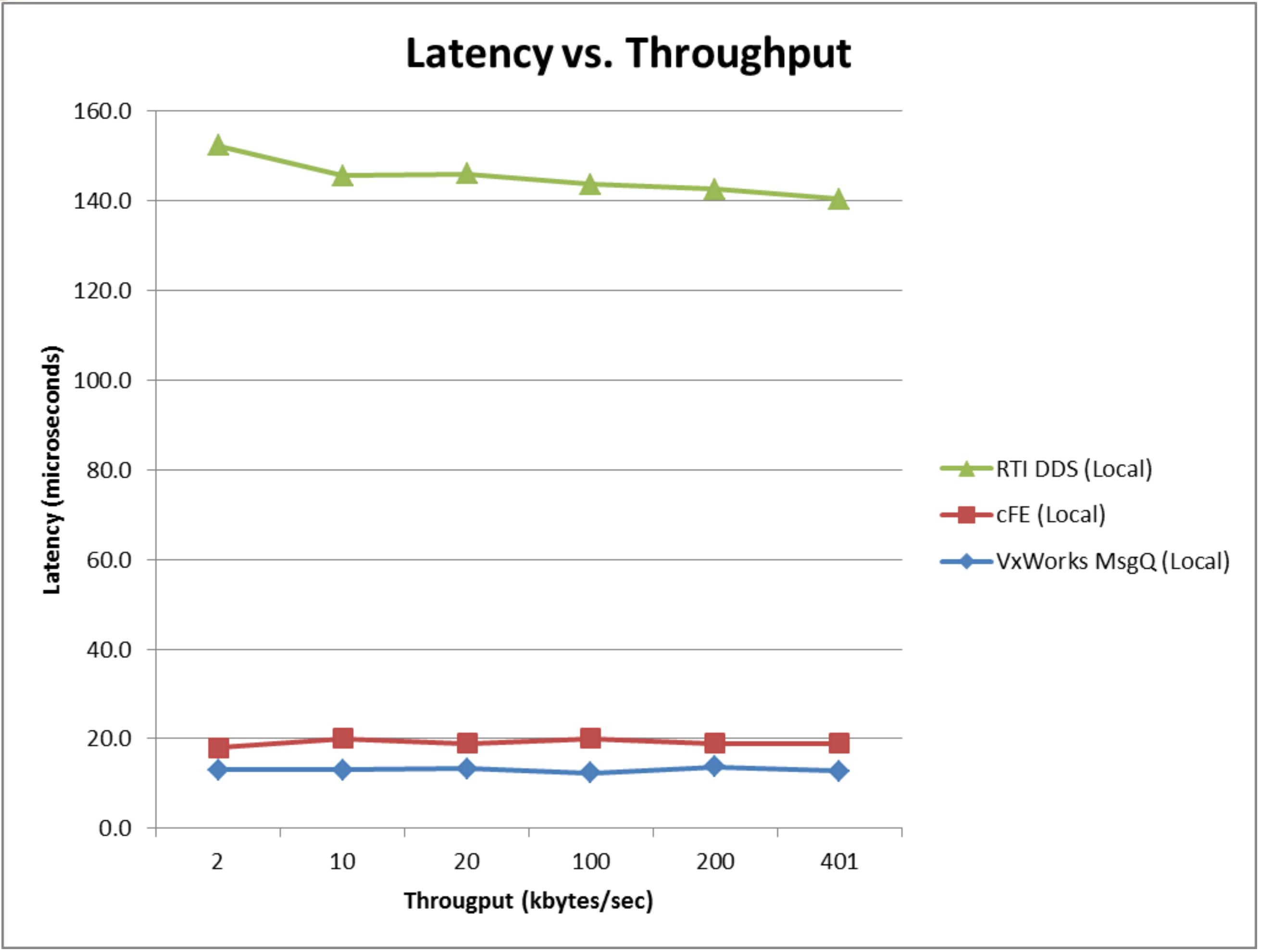
Concerns in LM that publish-subscribe will introduce unacceptable latencies

IRAD 2014: Middleware Performance Comparison

- **The test environment**
  - MVME6100 single-board computers in a VME backplane
    - Plentiful, PowerPC-based processors
    - Similar to BAE RAD750 which is widely used on LMSSC spacecraft programs
  - WindRiver VxWorks 6.6 RTOS
    - Used standard VxW message queues to compare to the other two
  - Vanguard VME analyzer for timing
    - Instrumented test code around middleware transport calls
    - Trap on VME address writes
    - Vanguard claims 7ns timing resolution
    - Experience has shown reliable resolution to hundreds of nanoseconds resolution
  - Test applications that transport messages of various lengths
  - For the moment, test only covers transport within a single SBC; not across processor nodes.
  - cFS SW Bus Network app still not released so prevented comparison across processor nodes
- **The expectations**
  - cFE performance will be dependent on number of subscribed users since bulk of delay is on sender
  - Less insight to RTI DSS implementation, but anticipating RTI's solution to be slower overall.

Test env focused on software running in a single single-board computer

IRAD 2014: Middleware Performance Results



Results for multi-single board case included for comparison

IRAD 2014: Middleware Performance Conclusions

- **cFE wins battle with RTI…**
  - Most comparable to local VxW message queues since same mechanism
  - RTI DDS still 7X slower within single board
  - Latency performance between the implementations mostly insensitive to data rates sampled
  - Need to continue to stress cFE with additional test cases examining more subscribers and longer messages
- **…but obviously not the answer for every use**
  - Ease of integration trades latency performance
- **Test environment gives relative measures expected on flight processors**
  - Additional benchmarking tests are needed on more flight-like processors to gain more confidence

Judicious use of publish-subscribe will ensure success for LM programs

IRAD 2014: SW Dev Environment

- **Continuing to adopt more and more FOSS tools**
  - Google test for unit test
  - Gcov for coverage analysis
  - Eclipse for IDE
  - Cppcheck for code standards and other static code analysis
  - Jenkins for continuous integration
  - Mercurial and Git for SW CM
- **Integrated tool suite using Eclipse is an improvement**
- **Conclusions**
  - Tool chain, process, and techniques are major aspect of being affordable
  - FOSS adoption taking off in LM compared to previous years

FOSS adoption continuing to accelerate as FOSS products mature