Constellations in the Cloud: Simulating and Emulating Heterogeneous On-board Processing for Distributed Measurement and Multi-Satellite Missions

Matthew French, Marco Paolieri, Vivek Venugopalan, Andrew Schmidt – USC / ISI

Tom Flatley, Alessandro Geist, Gary Crum – NASA GSFC

Ved Chirayath, Alan Li – NASA ARC

New Decade, New Challenges

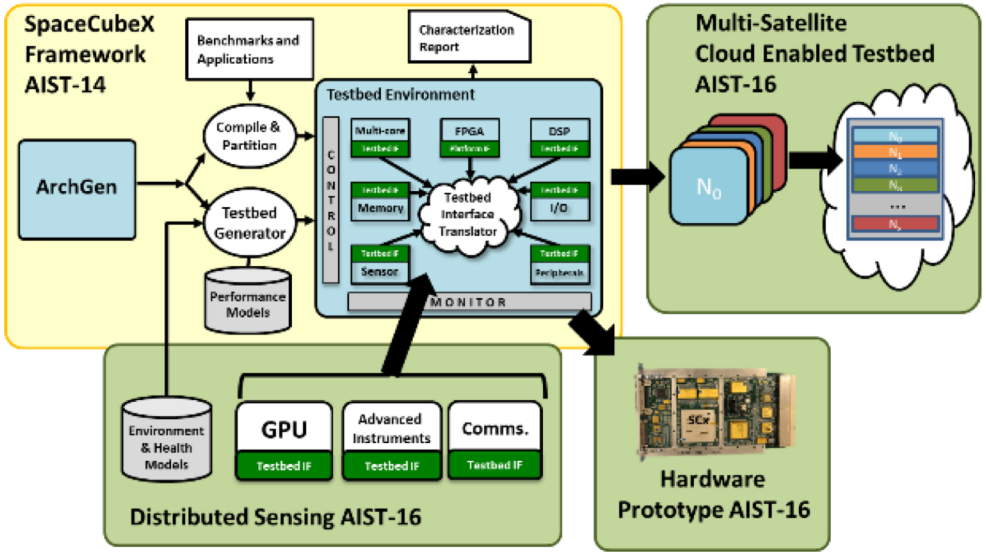
- 2017-2027 National Academy of Science Decadal Survey for Earth Science and Applications from Space (ESAS 2017) released
- Satellite constellations
  - Increased temporal sampling
- Multi-sensor and platform coordination
  - Distributed sensing
- Intelligent Sensors
  - Autonomous reaction to events
  - Self preservation

Multi-satellite, Multi-sensor Mission

- How can on-board computing support these new challenges?

SpaceCubeX2 Approach

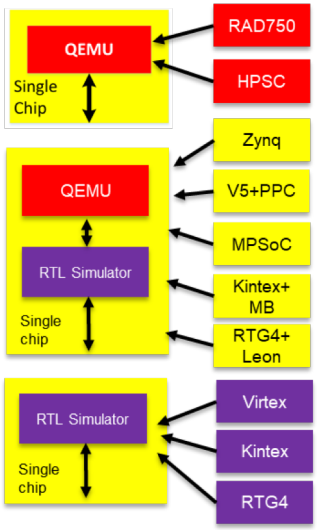
- Extend the AIST-14 SpaceCubeX On-board computing Analysis Framework to support analysis of new mission goals
- Benefits
  - Accessible, rapid prototyping of next generation satellite and multi-satellite constellations capabilities via virtual machines deployed in a cloud computing environment.
  - A proto-type heterogeneous on-board computer for experimentation of advanced autonomy and control capabilities required by intelligent instrument control and constellation management.
  - Accelerate migration of missions from UAV and airborne platforms to satellites to support distributed sensing.
  - Accurate, scalable approach to assessing Multi-Satellite mission performance.
  - Detailed analysis and initial run-time implementation of FluidCam Fluid Lensing, MiDAR, Diurnal Measurements, and Multi-Angle Measurement applications.

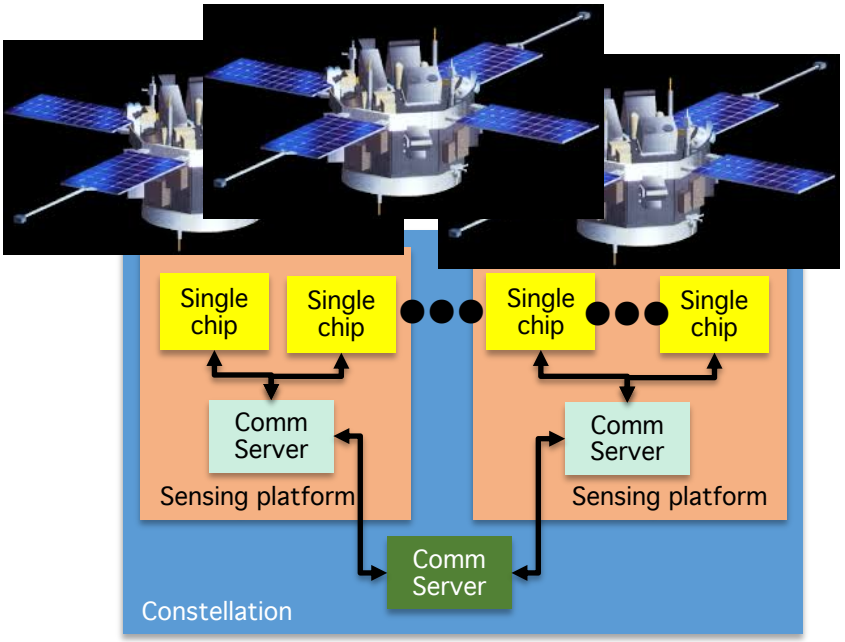


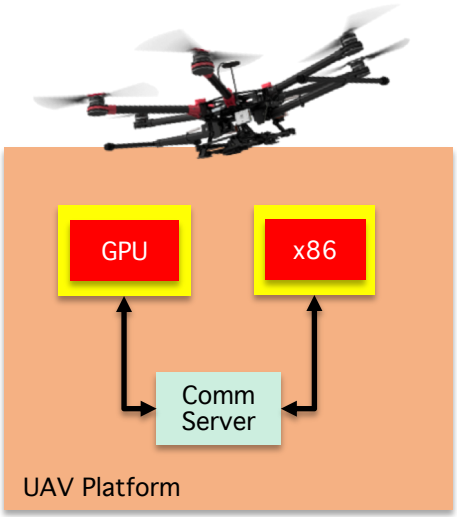
3

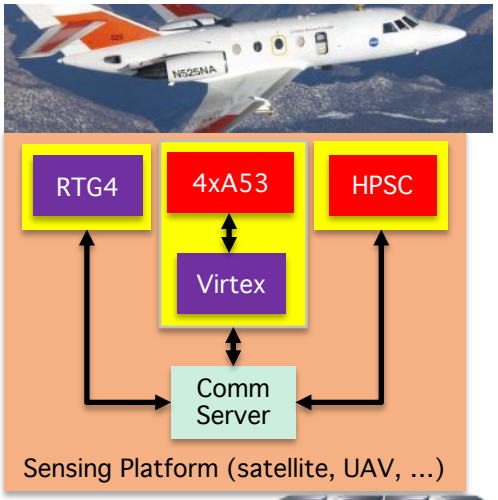
Processor Simulation Overview

- GPU co-processor emulation added to support airborne and UAV platforms
- Implement simulation framework for different platform devices
  - **QEMU** – Processor/CPU
  - **RTL** simulator – FPGA/peripherals
  - **Emulate on HW** – GPU, FPGA (limited)
- Processors Supported
  - CPU: Arm (HPSC, FPGA SoCs), PowerPC (NXP, RAD750)
  - GPU: Jetson, Tegra
  - FPGA: Xilinx Virtex, Ultrascale, and MPSoC; Microsemi RTG4





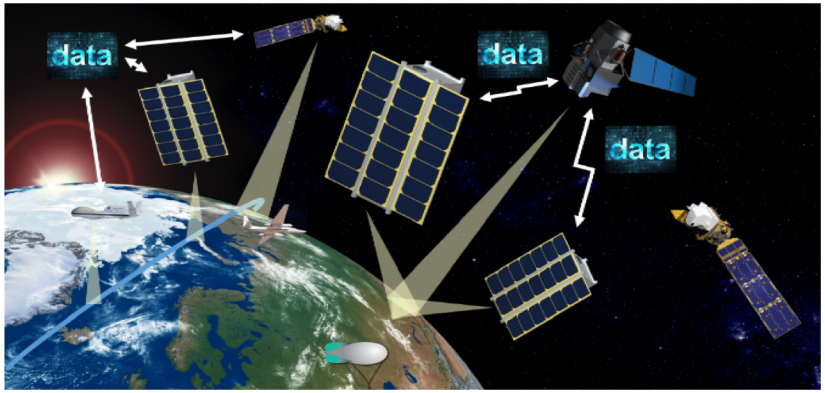




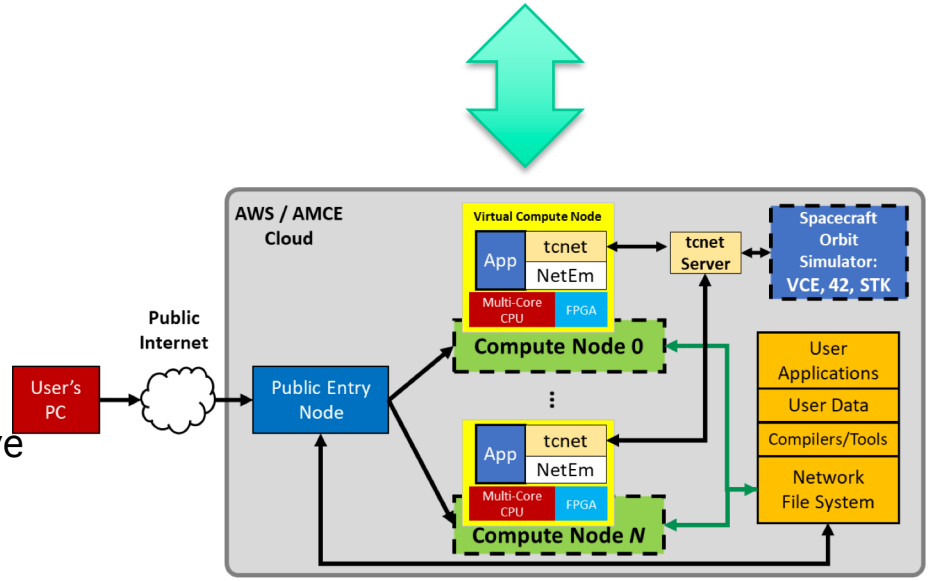
Constellation Simulation Approach

- Objective: Develop a software framework to facilitate exploration of remote sensing constellations from an on-board processing perspective.
- Simulate a scenario specified by the mission developer
  - Compute the location of each satellite, UAV, base station
  - Decide network latency/bandwidth of links
  - Receive controls and generate output of instruments
  - Simulate unexpected events, failures
- Provide a cloud environment to
  - Run the software of each satellite, UAV, base station
  - Emulate CPU models
  - Emulate networking between nodes
  - Send control signals to instruments, receive measurements
  - Collect metrics and logs for debug

Goal: Seamless transition from virtual to physical



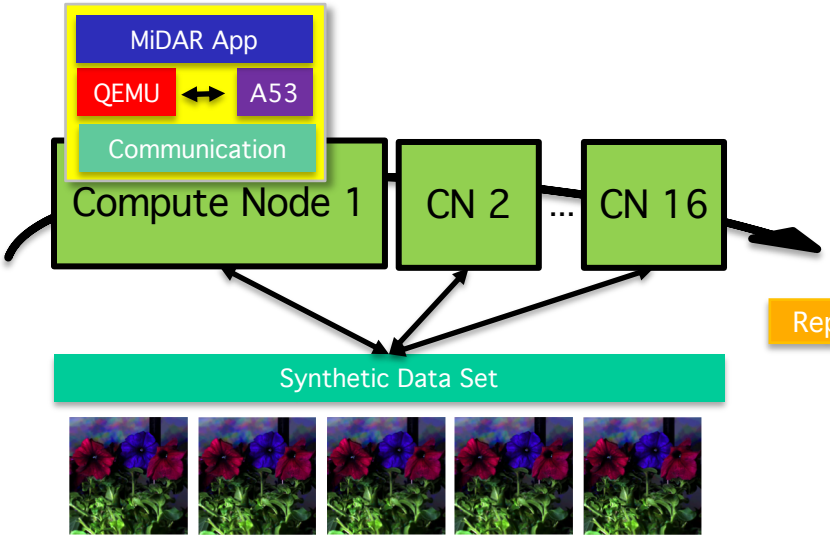
Multi-satellite, Multi-sensor Simulation



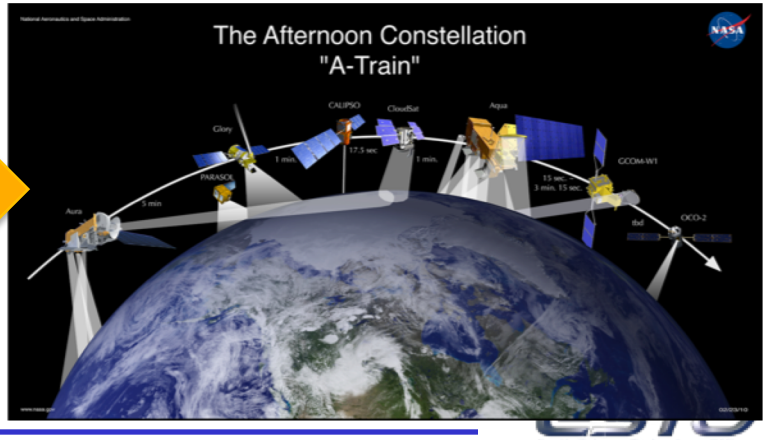
Virtual Constellation Engine

Multi-Satellite Proof of Concept

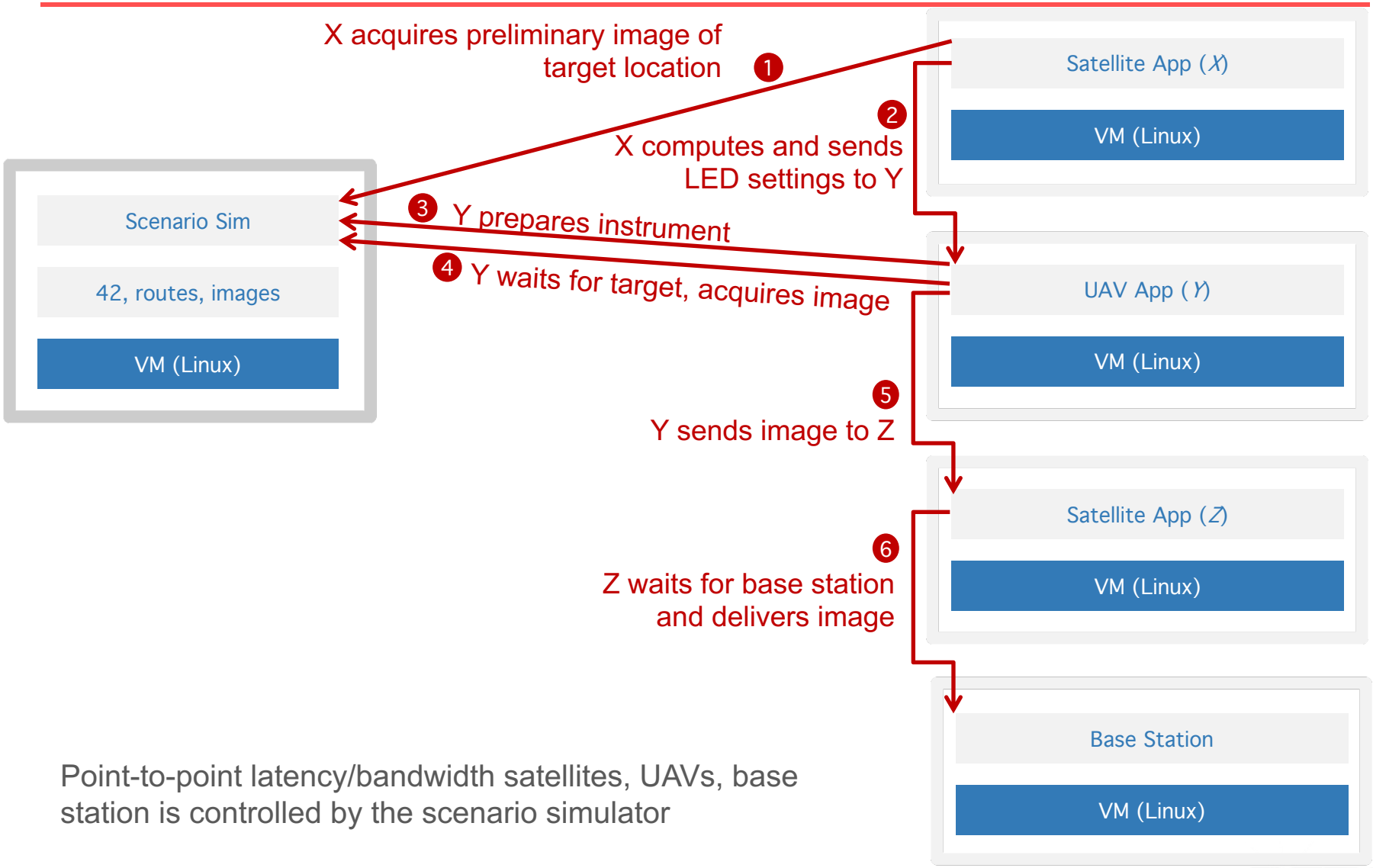
- At annual review, showed emulation of 10-node, A-Train-like constellation
  - MiDAR application operating on each node
  - Synchronized start/end, but otherwise parallel operation
  - Useful demonstration of cloud framework and scalability
  - Limitations on interactivity between platforms
- Enhanced communication architecture to enable rich exploration of collaborative platforms
  - Now modeling bandwidth, latency, and range



Represents



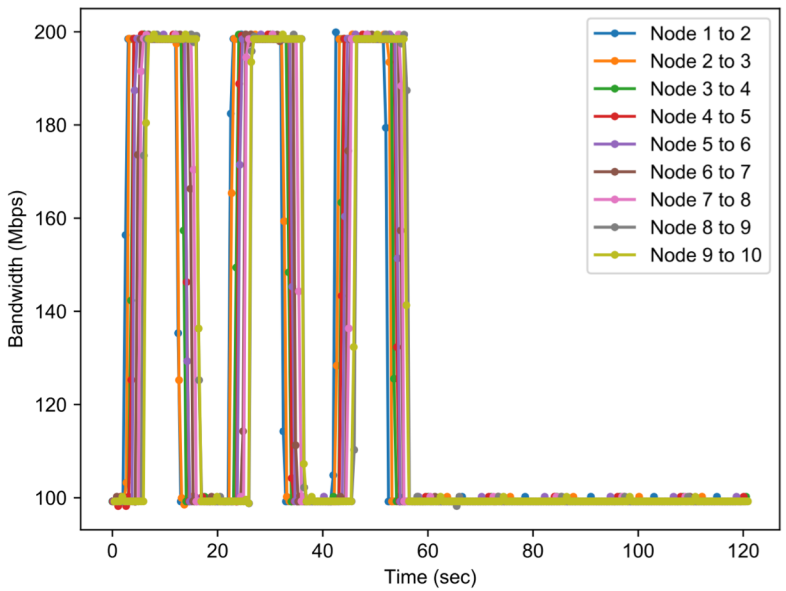
Satellite Constellation: Communications Example



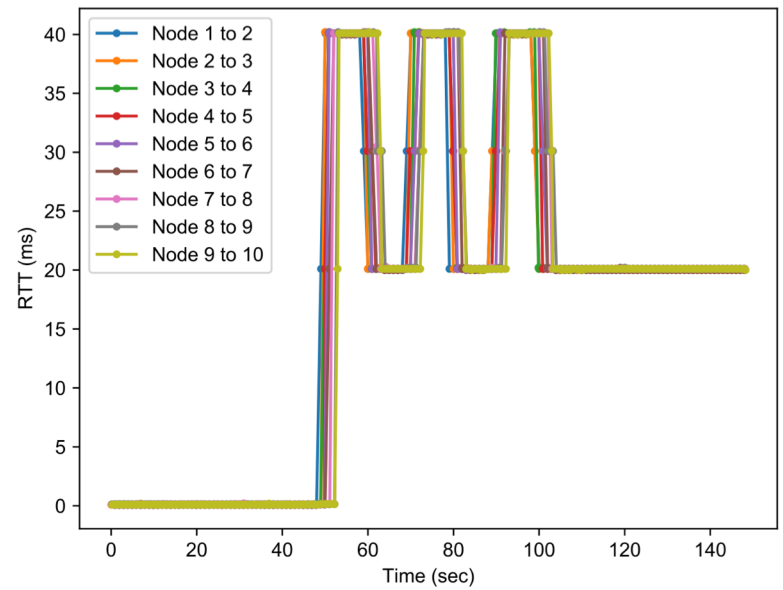
Point-to-point latency/bandwidth satellites, UAVs, base station is controlled by the scenario simulator

Communication Bandwidth and Latency Emulation

Bandwidth



Latency



Synthetic Experiments to Demonstrate Bandwidth and Latency Effects

- Running 10 VMs on AWS
- 0.7 Gbps with bursts of 10 Gbps
- Linear topology (1 → 2 → ... → 10)
- Monitoring bandwidth and RTT to next VM

Bandwidth Emulation Experiment

- Every 10 seconds, switch between 100 and 200 Mbps

Latency Emulation

- Every 10 seconds, switch between 10 and 20 ms for packets to next node

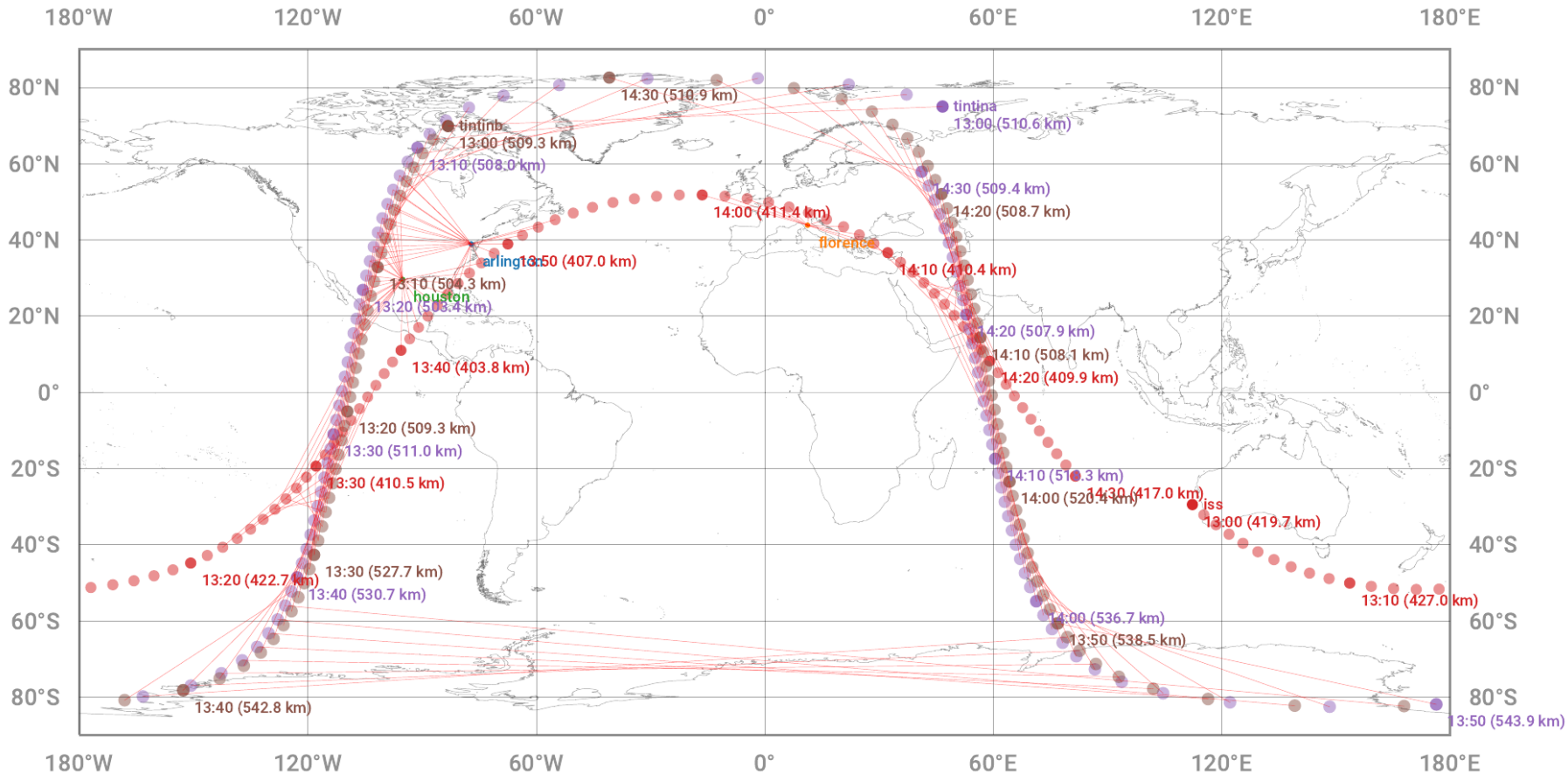
VCE Demo

vce : bash — Konsole <2>

File Edit View Bookmarks Settings Help

```
marco@tralcio:~/Projects/vce$ ls
demo  HISTORY.txt  LICENSE   MANIFEST.in  Pipfile.lock  setup.py  vce
docs  hosts        Makefile  Pipfile      README.rst    tests
marco@tralcio:~/Projects/vce$ cat demo/demo
```

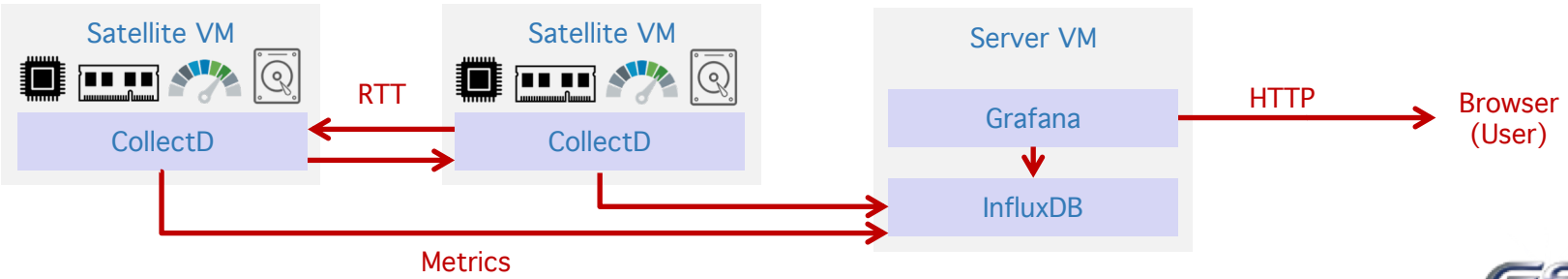
Orbits and Communication Visualization (2)



AWS Data and Metrics Collection

Monitoring of constellation VMs

- Each node runs **`collectd`** to gather application metrics and forward them to the server
- The server collects and stores data as timeseries (**`influxdb`**)
  - CPU, memory, disk utilization
  - Network utilization (TX, RX)
  - RTT to every node
  - State of each instrument
- Visualization tools (**`grafana`**) for real-time or offline analysis of the collected metrics

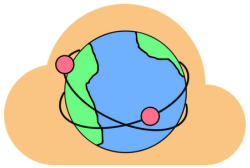


VCE Open Source Release

VCE Cloud Framework

Home Getting Started FAQ GitHub

USC University of Southern California



Virtual Constellation Engine

A cloud framework to emulate line-of-sight, latency, and bandwidth of satellite applications in AWS virtual machines. Performance metrics (CPU, memory, disk, network) are collected from the constellation and recorded in a time series database to analyze performance bottlenecks.

Motivation

The goal of VCE is to assist in prototyping, development and evaluation of distributed applications running on satellite constellations, airborne systems, and ground stations.

Example applications include Earth imaging, marine biology, and ecology studies based on spectral reflectance instruments and infrared imagers and sounders.

These missions require autonomous coordination across multiple platforms to capture dynamic events, to combine measurements from different instruments, to transmit preliminary results with low latency, and to perform intelligent instrument utilization for increased lifespan.

Approach

Each base station or satellite in the constellation runs as a separate AWS virtual machine.

At startup, a VCE agent is installed to control network parameters such as link state (up/down), latency and bandwidth: a VCE server (running on a separate VM) provides these parameters to the VCE agents based on the simulated position of each satellite.

In addition, the user can schedule network faults or distruptions, to test the performance of the distributed application under failure.

Metrics such as CPU, memory, disk and network usage are collected from each virtual machine over time, and stored in a time series database for real-time or offline analysis.

News

- Jun. 4, 2019: Version 0.1 of our cloud framework is released as open-source on GitHub.
- Aug. 1, 2019: Our cloud framework is presented at IGARSS 2019 in Yokohama, Japan.

This work is part of the SpaceCubeX project supported by the AIST Program of NASA Earth Science Technology Office (ESTO) under grant 80NSSC17K0286.

VCE Cloud Framework

Available at: https://isi-rcg.github.io/vce/

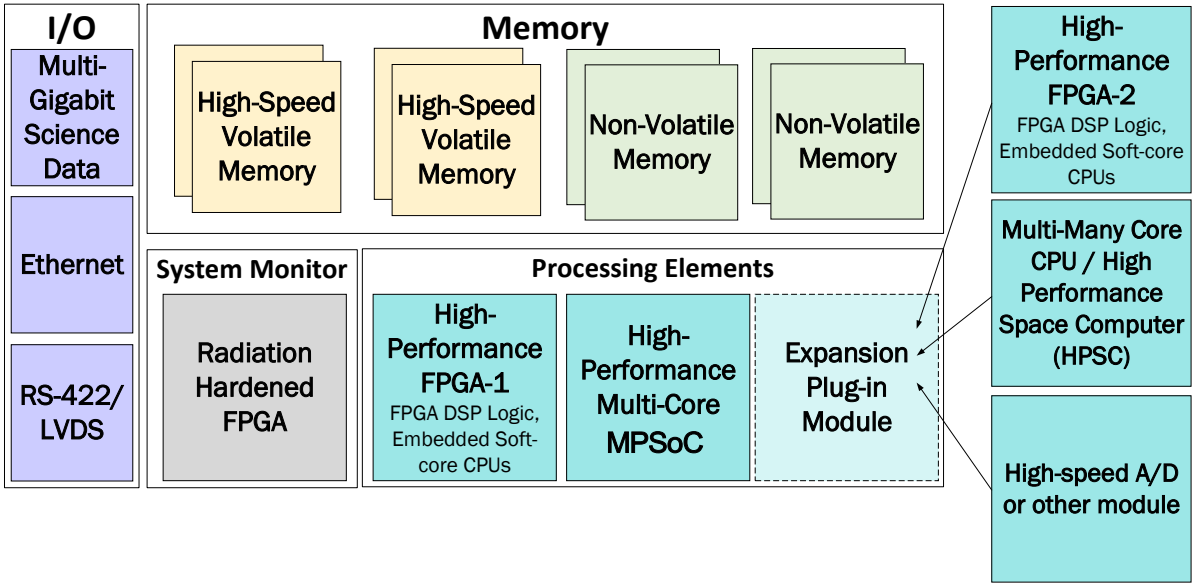
AWS account required

SpaceCube v3.0 On-board Computer

Overview

- Next-Generation SpaceCube Design
- 3U SpaceVPXLite Form-Factor
- Ultimate goal of using High-Performance Space Computer (HPSC) paired with the high-performance FPGA
  - HPSC will not be ready in time for the prototype design
  - Special FMC+ Expansion Card Slot

SpaceCube v3.0 Architecture



High-Level Specifications

1x Xilinx Kintex UltraScale

- 2x 2GB DDR3 SDRAM (x72 wide)
- 1x 16GB NAND Flash
- External Interfaces
  - 24x Multi-Gigabit Transceivers
  - 82x LVDS pairs or 164x 1.8V single-ended I/O
  - 30x 3.3V single-ended I/O,
  - 8x RS-422/LVDS Rx/Tx pairs
- Debug Interfaces
  - 2x RS-422 UART / JTAG

1x Xilinx Zynq MPSoC

- Quad-core Arm Cortex-A53 processor (1.3GHz)
- Dual Arm R5 processor (533MHz)
- 1x 2GB DDR3 SDRAM (x72 wide)
- 1x 16GB NAND Flash
- External Interfaces
  - I2C/CAN/GigE/SPIO/GPIO/SPW
  - 16x Multi-Gigabit Transceivers
- Debug Interfaces:
  - 10/100/1000 Ethernet (non-flight)
  - 2x RS-422 UART / JTAG

Rad-Hard Monitor FPGA

- Internal SpaceWire router between Xilinx FPGAs
- 1x 16GB NAND Flash
- Scrubbing/configuration of Kintex FPGA
- Power sequencing
- External Interfaces:
  - SpaceWire
- 2x 8-channel housekeeping A/D with current monitoring

MiDAR

IRAD FY18 – MiDAR Instrument kW-class Transmitter and Receiver Development

Principal Investigator: Dr. Ved Chirayath, ved.c@nasa.gov, 650-604-6278

Goals & Objectives

Capability Need/Knowledge Gap:

Bypassing the limits of passive instrumentation and the need for ambient downwelling solar radiation, MiDAR advances state-of-the-art active remote sensing through narrowband structure illumination from UAV platforms.

Objectives:

(1) Build high power MiDAR array and driver (2) Collimate multispectral beam (3) Precise control of array (4) Calibration

Technology Status Quo

Start TRL (12/1/17): 2

End TRL (12/1/19): 6

Technology State-of-the-Art:

Most prototypes at this stage have been relatively low power, stationary, and unable to achieve the levels of irradiance required for remote sensing applications at larger distances, and hence have been predominantly been purposed for the task of object detection, relighting and close-up monitoring.

Approach & Deliverables

Technical Approach:

(1) COTS LEDs for main MiDAR array, covering 4 UV, 4 RGBB, and 2 NIR bands, with an input power of 1-5kW and an output irradiance of 1-2 W/m^2/nm. Driver circuit is a switch-mode DC-DC buck PWM capable circuit for control.

(2) Parabolic reflecting optic to within a 30° half-cone angle with 3D printed LED base acting as heat sink, allowing for electrical connection throughput

(3) Control through MOSFET gates via microprocessor, PWM to control amplitude modulation

(4) Calibration through spectroradiometers, accounting for temperature, intensity distribution pattern, and physical degradation.

Deliverables: *kW-class MiDAR array, with collimating optics and associated electronics. Integration with existing sUAS fleet, including flight demonstration. Instrument and calibration specifications*

Conceptual view of MiDAR transmitter unit (left) and associated radiation distribution pattern (right)

Potential Next Steps & Applications

Next Steps:

Airborne and spaceborne Earth Science mission possibilities through EV, ESTO InVEST. Follow-on proposals to ACT and IIP focusing on high altitude/space demonstrations. Ongoing airborne field campaign capability with applications to life-detection missions for astrobiology through PSTAR.

Customers & Applications:

NASA ESTO, NASA Earth and Space Science, USGS, IUCN, EPA, US Army. Applications in nocturnal remote sensing, long-distance optical comms, BRDF characterization, mineral ID, UV imaging, and 3D SfM.

Resources & TA

Roadmap TA Alignment:

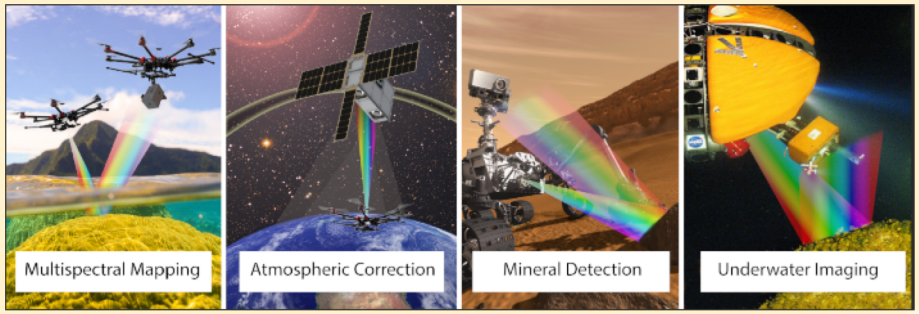
8.1.2, 8.1.3 Electronic and Optical Components in Technology Roadmap 8

Total FTE: 0.4

Total Procurement: ~$110k/yr

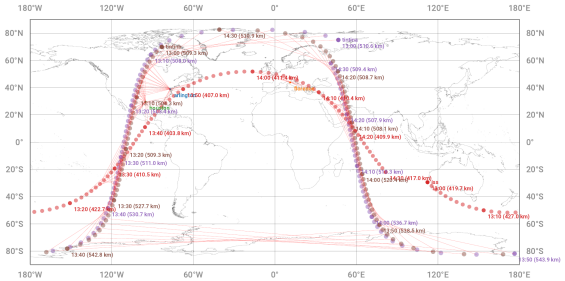
Total Contributions: $1.8M

MiDAR is the first active multispectral video instrument with remote sensing and life detection capabilities



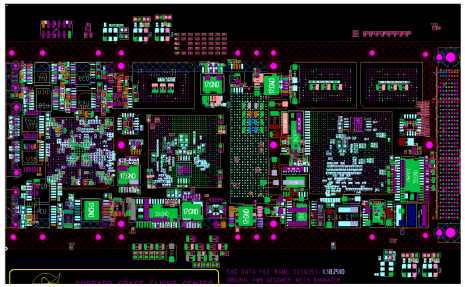
Summary

- Virtual Constellation Engine
  - Major enhancements to communications modeling, orbit simulation, and system level metrics
  - Now capable of positing and simulating arbitrary constellations.
  - Open source release. Seeking test cases and users.
- Application Mapping to Heterogeneous Computing
  - Python to hardware mapping tools published at IEEE FCCM, open sourced, and demonstrated at several conferences.
- Hardware Updates
  - PCB completed schematic design, review, and pre-layout signal integrity check
- MiDAR Application Mapping
  - Code refactored to support CPU, GPU, and FPGA profiling and implementation. Updating algorithm for real-time operations.
- Future Research and Development
  - Integration with OSSEs
  - Exploration of AI Edge Computing
  - SpaceCube3 Hardware Deployment



VCE Multi-satellite Orbit Visualization

VCE Multi-satellite Metrics Visualization



SpaceCube3 Hardware Layout

Questions?

Communications Demo: Scenario Specification

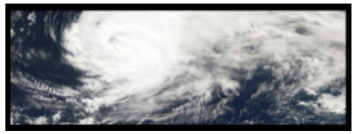
Simulation parameters controlled through a configuration file

- General parameters
  - Initial simulated time
  - Duration
  - Frequency to recompute satellite positions
  - Frequency to collect metrics from running VMs
- Constellation Satellites
  - Orbital Elements, specified as two-line element set (TLE)
  - Virtual machine type (`c5.4xlarge`) and image (user-provided)
  - Hostname for communication with other satellites
  - Onboard instruments (each has state variables)
- Ground Stations
  - Virtual machine type, image, hostname
  - Latitude, longitude, altitude

```
simulation:
  zone: us-east-1d                        # AWS zone where
instances are started
  orbits_start: '2019-02-15 13:00:00'     # UTC, start time of
orbits simulation
  orbits_step: 1.0                        # in minutes, frequency
of orbits recalculation
  duration: 90.0                          # in minutes (cloud
instances stopped afterward)
satellites:
- hostname: iss
    type: t2.micro
    ami: ami-02592fa20805dbe0b
    tle1: '1 25544U 98067A   19040.02956382  .00000286  00000-0
11897-4 0  9995'
    tle2: '2 25544  51.6414 278.5526 0005375   9.1011 131.7761
15.53240808155340’
- hostname: tintina
    type: t2.medium
    ami: ami-02592fa20805dbe0b
    tle1: '1 43216U 18020B   19038.95616218  .00000830  00000-0
41860-4 0  9990'
    tle2: '2 43216  97.4542  48.4380 0013970 130.6064 229.6388
15.19819550 53208'
    midar:
      state: on
stations:
- hostname: arlington
  type: t2.micro
  ami: ami-02592fa20805dbe0b
  lat: 38.882737
  lon: -77.105993
  alt: 78.0
```

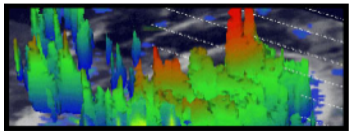
NASA Earth Science Mission Computing Challenges ~2010

- New Instruments required to produce essential data to help scientists answer critical 21st century questions
  - Global climate change, air quality, ocean health, ecosystem dynamics, etc…
- Missions specifying instruments with significantly increased:
  - Temporal, spatial, and frequency resolutions → to global, continuous observations
  - Current/near-term data at rates >10^8 to 10^{11} bits/second
- On-board processing ~100-1,000x than previous missions (compression, storage, downlink)
- Adding new capabilities such as low-latency data products for extreme event warnings

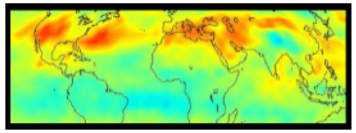


ACE

GEO-CAPE

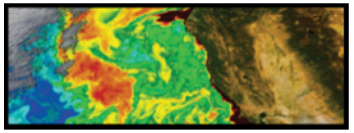


3D-Winds

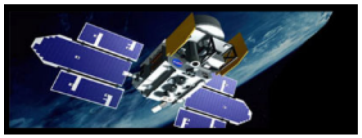


ASCENDS

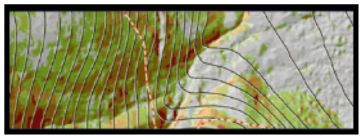
HyspIRI



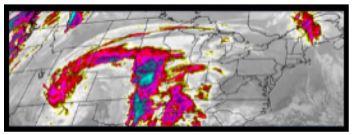
PACE



TSIS-1



LIST

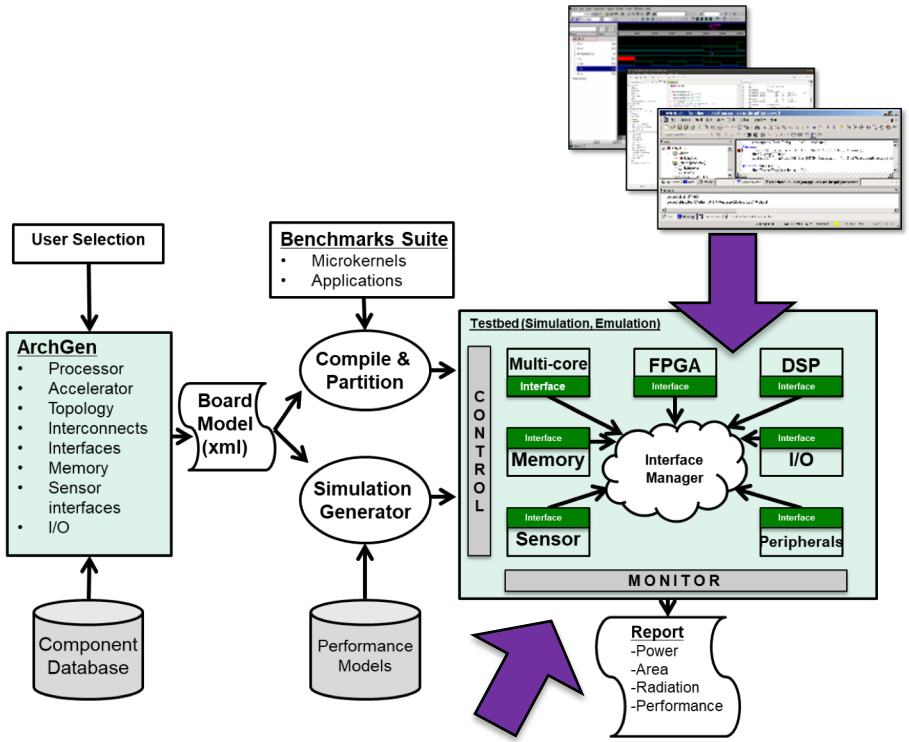


PATH

Hybrid computing is a key cross-cutting technology directly applicable to missions recommended in the Decadal Survey

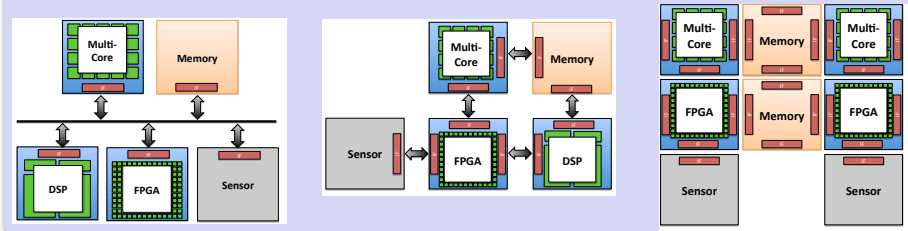
AIST-14: Onboard Computing Analysis Framework

Simulation Platforms

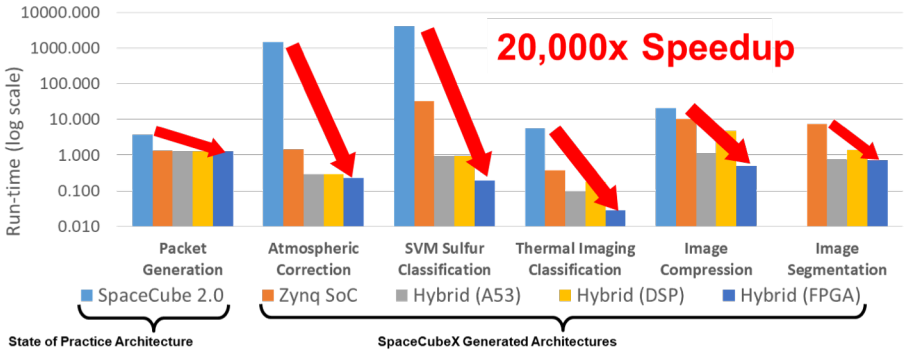


Emulation Developer Board Platforms

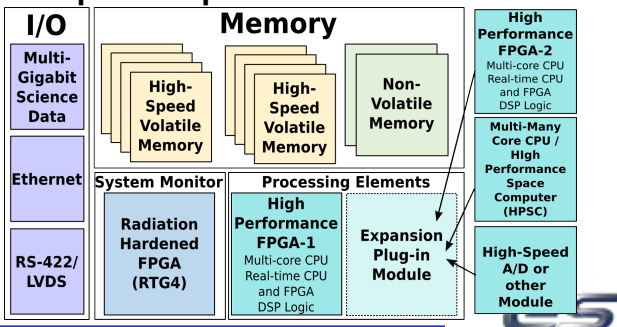
Board Topology Candidates



Performance Analysis for SpaceCubeX Architectures



Proposed SpaceCube 3.0 Architecture



End to end tools which enable rapid, accurate exploration of on-board

Publications, Demos, and Educational Outreach

- Paper submissions
  - Andrew G. Schmidt, Vivek Venugopalan, Marco Paolieri, Matthew French, "Constellations in the Cloud: Virtualizing Remote Sensing Systems," submitted to 2019 IEEE International Geoscience and Remote Sensing Symposium.
  - Marco Paolieri, Andrew G. Schmidt, Vivek Venugopalan, Matthew French, "Rapid Prototyping and Verification of Remote Sensing Constellations," submitted to 33rd Annual Conference on Small Satellites
- Demonstrations
  - Hot & sPyC: Enabling Software Developers to Target Heterogeneous On-board Computing Accelerators, 2018 AGU Fall Meeting NASA Booth Demo
  - Hot & Spicy: Improving Productivity with Python and HLS for FPGAs, 2018 IEEE FCCM Demo night
  - Various industry and government: Northrup Grumman, UTRC, Raytheon, Mahlet Consulting, DoD and IC agencies
- Published Hot & sPyC at top FPGA conference
  - Sam Skalicky, Joshua Monson, Andrew Schmidt, and Matthew French: **Hot & Spicy: Improving Productivity with Python and HLS for FPGAs**, 26th IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM 2018), April 2018
  - Open source version of Hot & sPyC released
    - Available at: https://github.com/ISI-RCG/spicy

Virtual Constellation Engine Objects

Constellation
- Number, type of nodes (In-situ, UAV, Airborne, Sat, Ground station)
- Interconnect topology

Platform
- Type (In-situ, UAV, Sat)
- Position, Orbit
- Instrument
- On-board processing
- On-board storage
- Communications type

Instrument
- Type
- Data rate, format
- Field of view
- Control
- Health

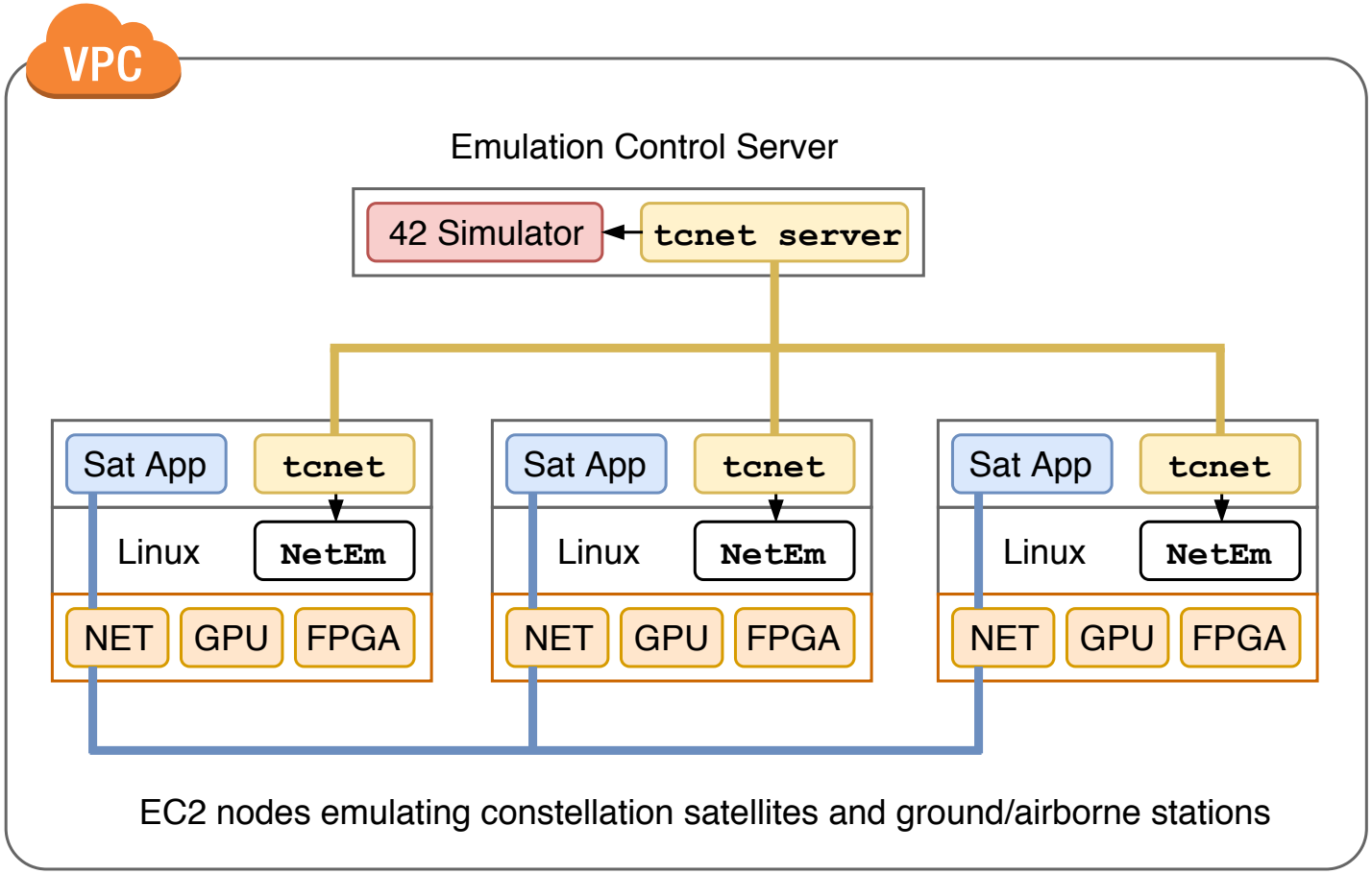
On-board Processing (ArchGen)
- Type
- Board Topology
- Onboard memory

On-board Mass Storage
- Type
- Depth, width
- Latency

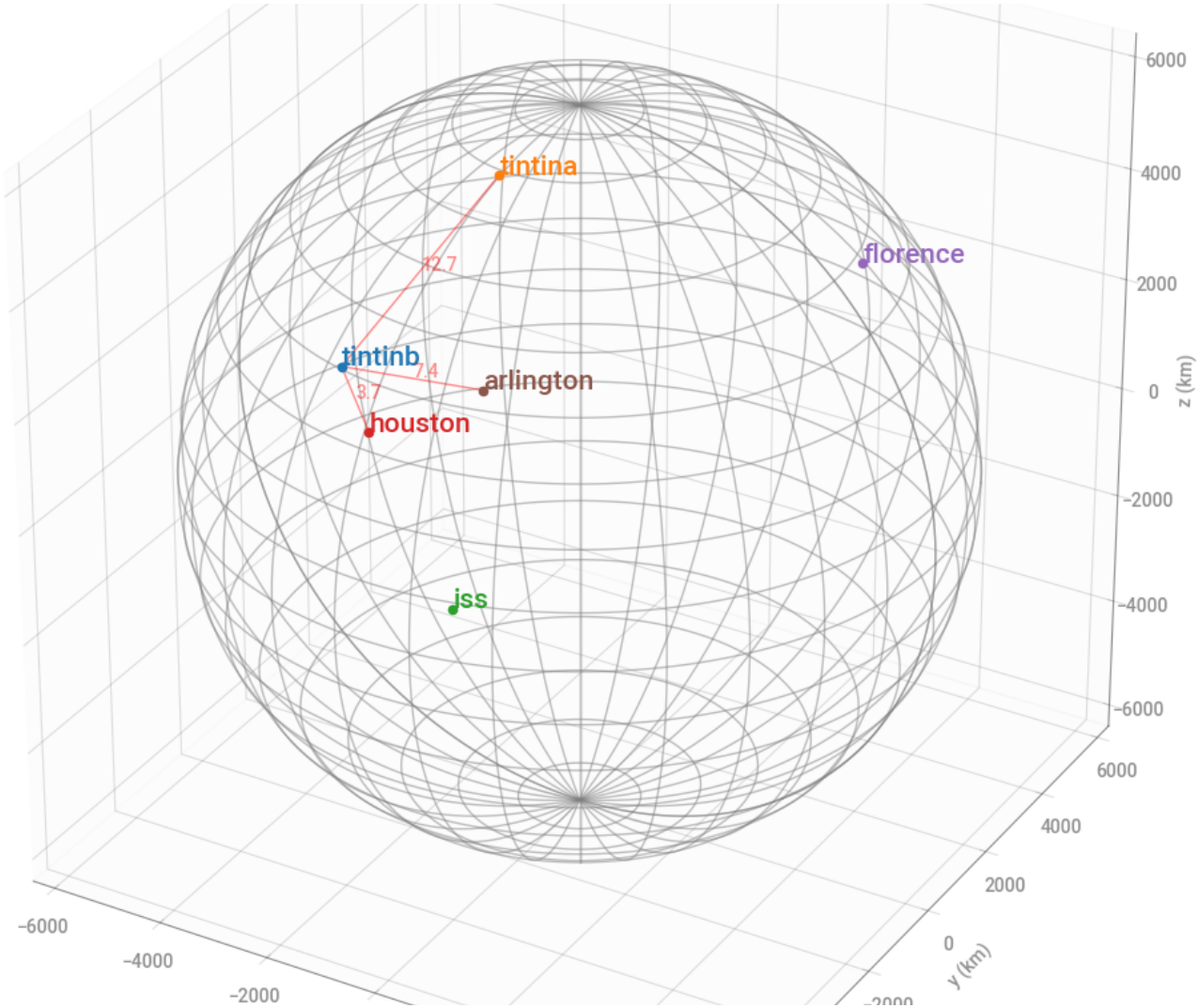
Communications
- Type
- Frequency
- Bandwidth
- Antenna
- Pointing accuracy
- Radiated Power

Objects enable common representation for all nodes in mission

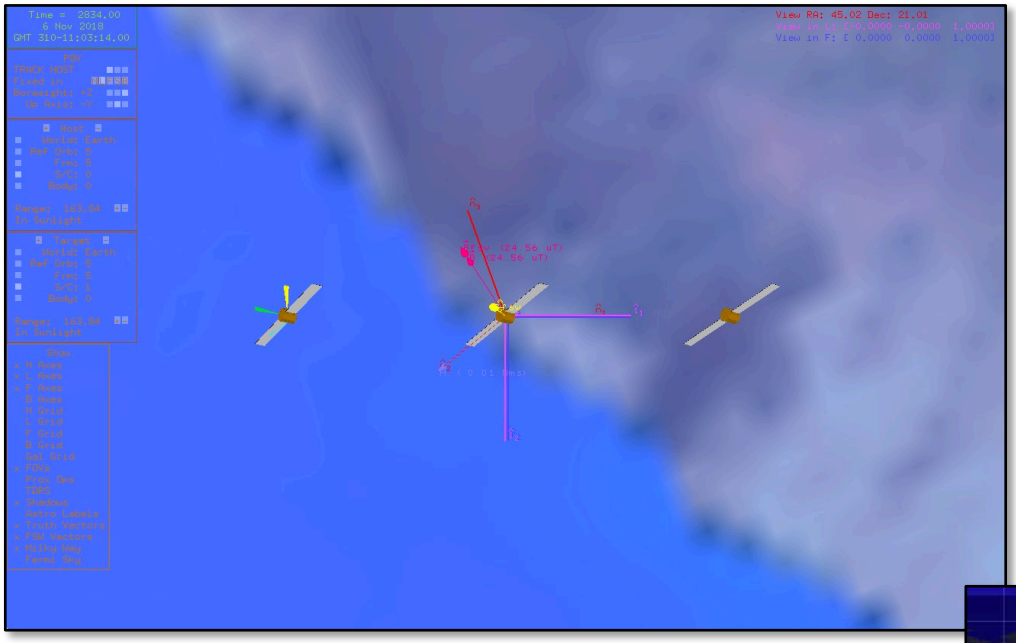
System Architecture of Cloud Environment



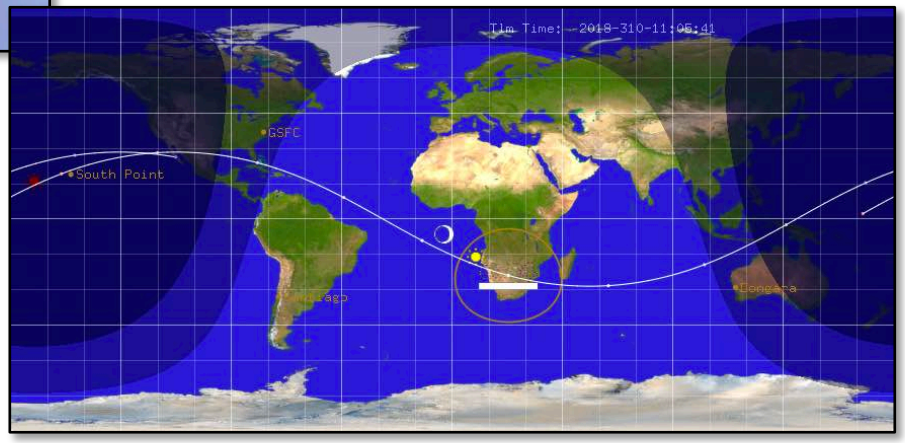
Orbits and Communication Visualization (1)



Spacecraft Orbit Simulator Compatibility



```
SC 0
POSITION  5.595633e+06 -4.654951e+05 3.270739e+06
VELOCITY  -3.943655e+03 -9.658923e+02 6.786713e+03
ANGVEL  -2.150729e-03 -8.730559e-03 3.047642e-03
QBN  4.978506e-01 2.715380e-01 6.578030e-02 8.210267e-01
SUNVEC  2.046207e-01 7.946831e-01 -5.714973e-01
MAGVEC  -4.000001e-05 3.950385e-06 -2.229278e-05
ANGMOM  -2.150729e-01 -1.746112e+00 9.142926e-01
[EOF]
TIME  2014-172-00:00:09.000000000
SC 0
POSITION  5.591685e+06 -4.664607e+05 3.277523e+06
VELOCITY  -3.951781e+03 -9.652154e+02 6.781957e+03
ANGVEL  -2.166512e-03 -8.826776e-03 3.082492e-03
QBN  4.976636e-01 2.670972e-01 6.514566e-02 8.226457e-01
SUNVEC  2.020739e-01 7.952902e-01 -5.715589e-01
MAGVEC  -4.017491e-05 4.007970e-06 -2.203250e-05
ANGMOM  -2.166512e-01 -1.765355e+00 9.247477e-01
[EOF]
TIME  2014-172-00:00:10.000000000
SC 0
POSITION  5.587729e+06 -4.674255e+05 3.284303e+06
VELOCITY  -3.959900e+03 -9.645371e+02 6.777191e+03
ANGVEL  -2.175600e-03 -8.892831e-03 3.107244e-03
QBN  4.974667e-01 2.626091e-01 6.450325e-02 8.242588e-01
SUNVEC  1.994947e-01 7.959086e-01 -5.716042e-01
MAGVEC  -4.035051e-05 4.067855e-06 -2.176584e-05
ANGMOM  -2.175600e-01 -1.778566e+00 9.321733e-01
[EOF]
```



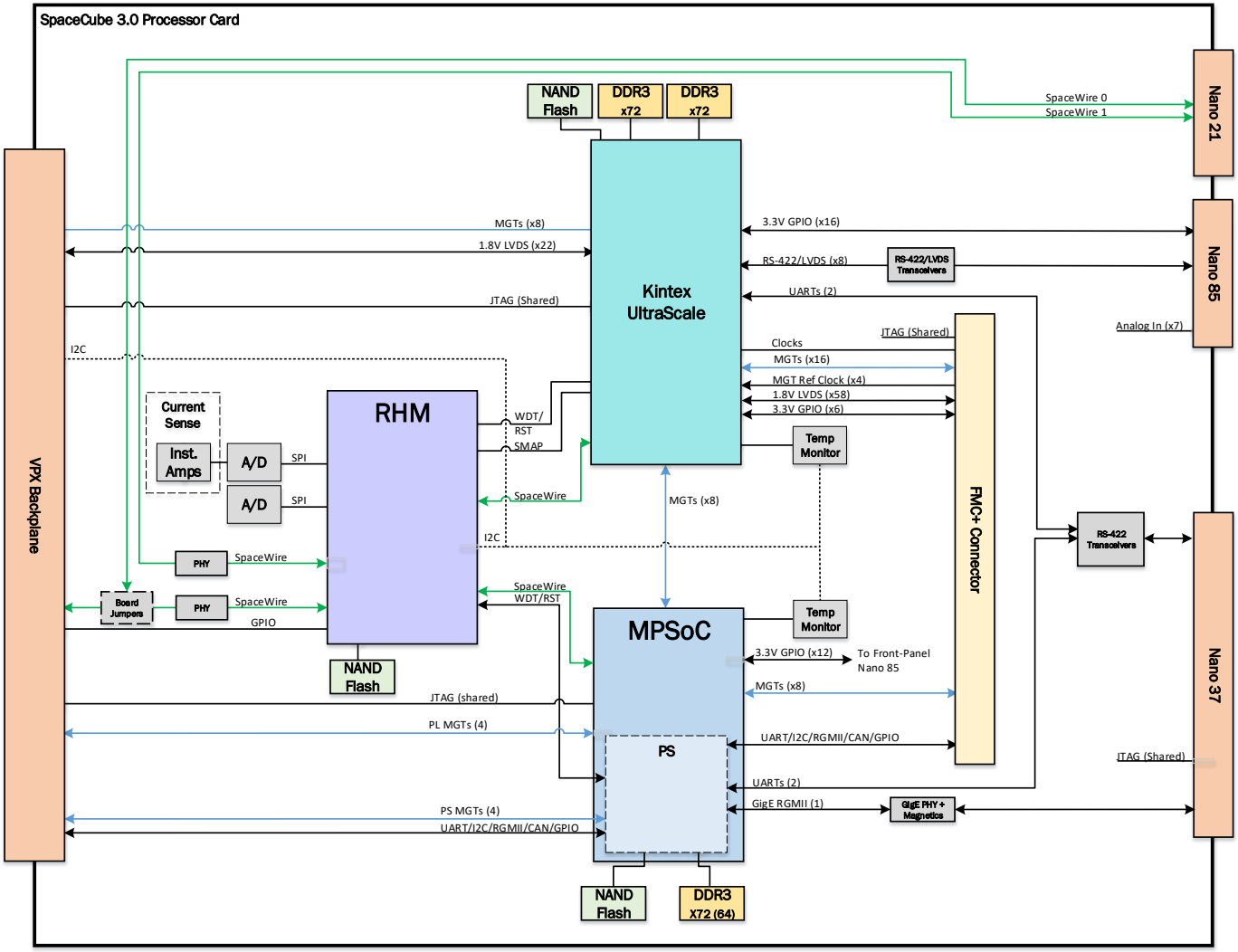
GSFC's 42: General Purpose Multi-body, Multi-Spacecraft Simulation

- Currently can be used to generate inputs for VCE
- Additional development required to integrate VCE within 42

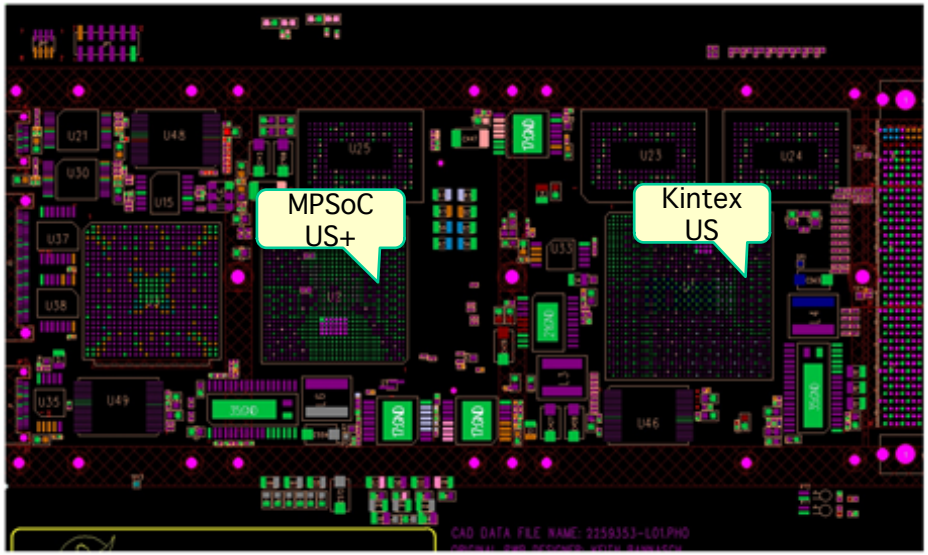
System Tool Kit (STK):

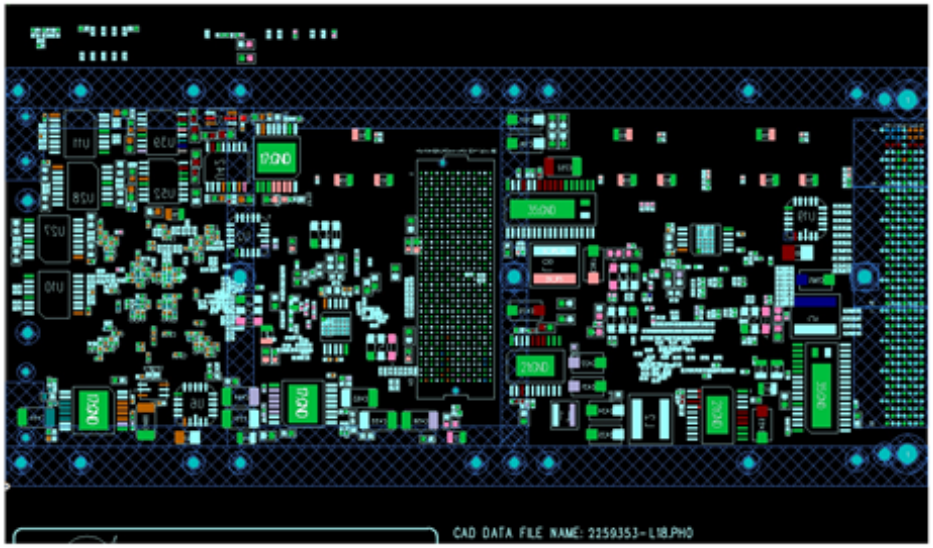
- High fidelity, industry standard. Paid licensing. Suitable for high TRL modeling
- VCE extensible to interface with STK if needed

Prototype Block Diagram



PCB Layout Parts Placement in Progress



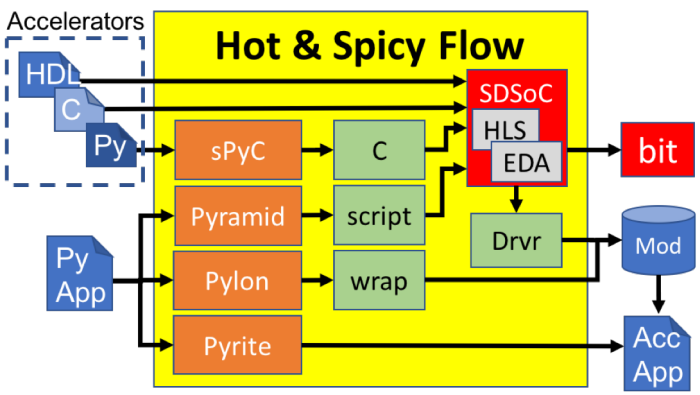


Hot & Spicy – Accelerating Mapping Python Applications to FPGAs

Motivation: ARC implementation of MiDAR in Python

Goal: Accelerate important Python functions by leveraging existing EDA tools (HLS)

- Developed Framework to accelerate Python Apps
  - Open-source release at https://github.com/ISI-RCG/spicy
- Targets SoC+FPGA systems where App can be Python
- Cross-compiles Python function to C, accelerates with HLS
- Automates system generation, drivers, integration



Canny Edge Detection from Earth Science Benchmark

39,137x performance gain over original Python
6x gain over optimized OpenCV

Reduces application mapping time from Months to Hours, enabling rapid exploration of optimization paths