Jet Propulsion Laboratory
California Institute of Technology

Validation of Flight Software on the Qualcomm Snapdragon 855 on the International Space Station

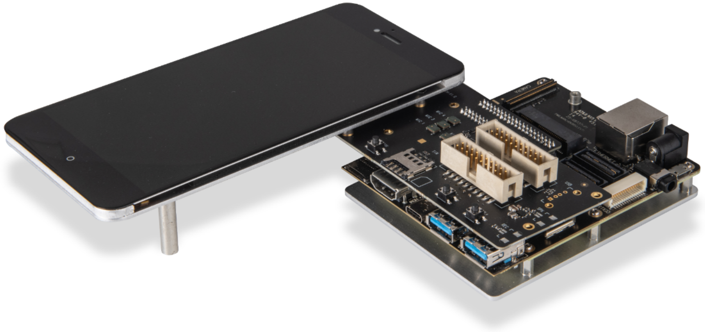
Jason Swope, Emily Dunkel, Steve Chien

This research was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Jet Propulsion Laboratory
California Institute of Technology

Background

Qualcomm Snapdragon 855 HDK



All Snapdragon images courtesy Qualcomm

- 8 core ARM system
  - 4 “silver” high efficiency cores ~ 1.80 GHz
  - 3 “gold” high performance cores ~ 2.42 GHz
  - 1 “gold prime” very high performance core ~ 2.84 GHz
- Adreno 640 Graphics Processing Unit (GPU)
- Qualcomm Hexagon 690 Digital Signal Processor (DSP)
- Neural Processing Engine
  - Directly supports Convolutional Neural Networks (CNNs) in hardware
- Running Android OS

Also Snapdragon Automotive Development Processor Units running linux (for automotive).

Rad 750

Current computing for MSL, M2020

PowerPC Heritage

Image re: MSL from CNET

JPL’s Sabertooth

- LEON 4 Based CPU
- Target 8-10x improvement in evolution from Sphinx

Image credit citation below.

W. Whittaker
Sabertooth: Integrated Avionics for Small Spacecraft Missions
2019 Space Computing Conference
https://trs.jpl.nasa.gov/bitstream/handle/2014/51550/CL%2319-4553.pdf?sequence=1&isAllowed=y

Embedded Processors

Processor	Snapdragon 855	Movidius Myriad X	Rad750	Sabertooth
Power	5W	< 1W	10+ W 5 W?	3W https://trs.jpl.nasa.gov/bitstream/handle/2014/51550/CL%2319-4553.pdf?sequence=1&isAllowed=y
MIPS	https://www.notebookcheck.net/Qualcomm-Snapdragon-855-SoC-Benchmarks-and-Specs.375436.0.html		typical 266 up to 400	1200
Cores, Clocks	8 @ 1.7-2.8 GHz	7 SHAVE @ 700MHz	1@110-200 MHz	4@ ?
RAM, NVM	16 GB	4 GB	256 MB 2GB	192 MB 8 GB
Coprocessors	GPU, DSP*, AIP* *quantized models: 8 bit fixed point	AIP** **half-precision floating point		Motor controllers

International Space Station Experiment

Hewlett Packard Enterprise Spaceborne Computing-2

Delivered to ISS turnover: Fall 2020

Delivered to the ISS onboard Cygnus NG-15: February 20, 2021.

Powered on: May 12th 2021.

Hewlett Packard Enterprise Spaceborne Computing-2 package:

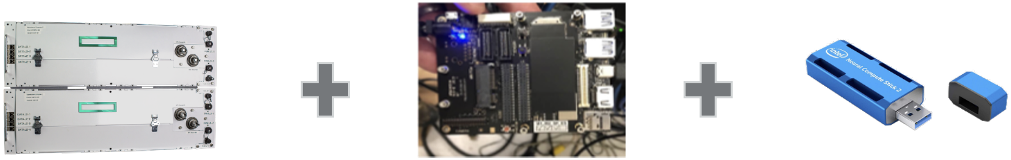
- COTS Linux workstations from HPE
- Intel Xeon 5215 Processor (10 cores)
- 4 NVIDIA Tesla GPUs
- 2 Machines aboard the ISS

2x Snapdragon 855 HDKs running Android

- Radios disabled

2x Myriad X Processors

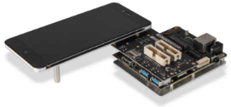
Uplinks possible periodically to load new SW



All SBC-2 images credit HPE.

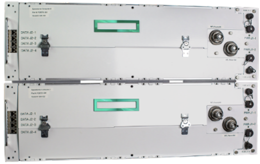
Experiment Setup

1. JPL
   Develops and Tests



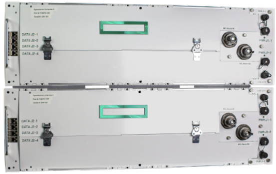
HPE:
Chippewa Falls, WI

HPE
Ground
Testbed



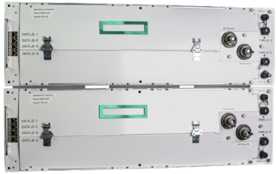
2. JPL Tests

HPE Flight
Testbed
(Ground)



3. HPE Tests

4. HPE Tests



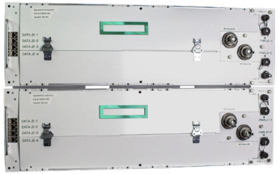
SBC-02 on ISS

5. Results → JPL

Test Harness Setup

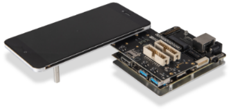
Test Harness
Computer

or



Code, Data

Results



This setup is not completely unlike an “instrument coprocessor” setup

Test Harness runs on laptop or SBC-02 linux host
Test Harness:

iterates through experiments:

delivers experiment code and data to embedded processor,

runs experiments on embedded processor
cleans up after execution, retrieving test results,
reboots if needed (timeouts)

Jet Propulsion Laboratory
California Institute of Technology

Flying Machine Learning Models

Machine Learning Models Ported and Run on ISS

- **Objective**: Deployment of machine learning models for inference on a Snapdragon on HPE's Spaceborne Computer-2 onboard the ISS to demonstrate the power of CNN hardware support

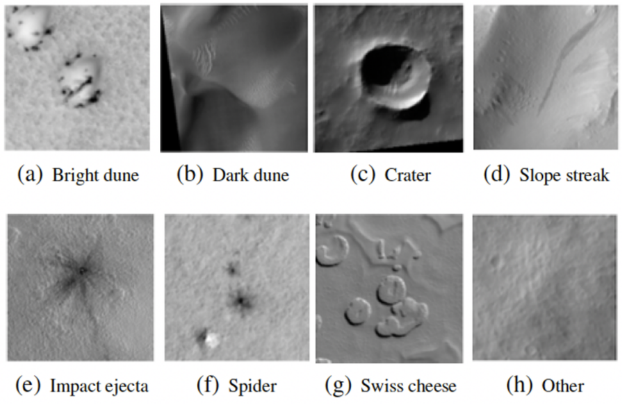
- **Mars HiRISE Classifier**
- Salience Detector
- Mars MSL Classifiers (2)
- **NavCam Image Segmentation**
- UAVSAR Flood Mapping
- SMICES Storm Classification
- **Super Resolution for Spectroscopy**
- Standard Deep Learning Benchmarks (7)

- We will show a range of the models that can be deployed

For model deployment on another COTS processor, please see: "Benchmarking Deep Learning On a Myriad X Processor Onboard the International Space Station" by Léonie Buckley (Ubotica) and Emily Dunkel (JPL)

Mars HiRISE: Deep Learning for Landmark Classification

- What: Mars surface feature classification using orbital imagery
- Applications: autonomous data collection, targeted downloads, commanding space assets, science interpretation
- Details:
  - Data: Imagery collected by the HiRISE instrument onboard the Mars Reconnaissance Orbiter (MRO)
  - Model: Transfer learning from AlexNet [Mars1, Mars2, DL]



Example of HiRISE Classes

- [Mars1] Wagstaff, Lu, Dunkel, Grimes, Zhao, Cai, Cole, Doran, Francis, Lee, Mandrake, 2021. Mars Image Content Classification: Three Years of NASA Deployment and Recent Advances. Innovative Applications of Artificial Intelligence, 33.
- [Mars2] Wagstaff, Lu, Stanboli, Grimes, Gowda, Padams. 2018. Deep Mars: CNN Classification of Mars Imagery for the PDS Imaging Atlas. The Thirteenth AAAI Conference on Innovative Applications of Artificial Intelligence. 7867-7872.
- [DL] Krizhevsky, Sutskever, Hinton. 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25, 1097-1105

Mars HiRISE: Classification Discrepancy, Timing, and Power

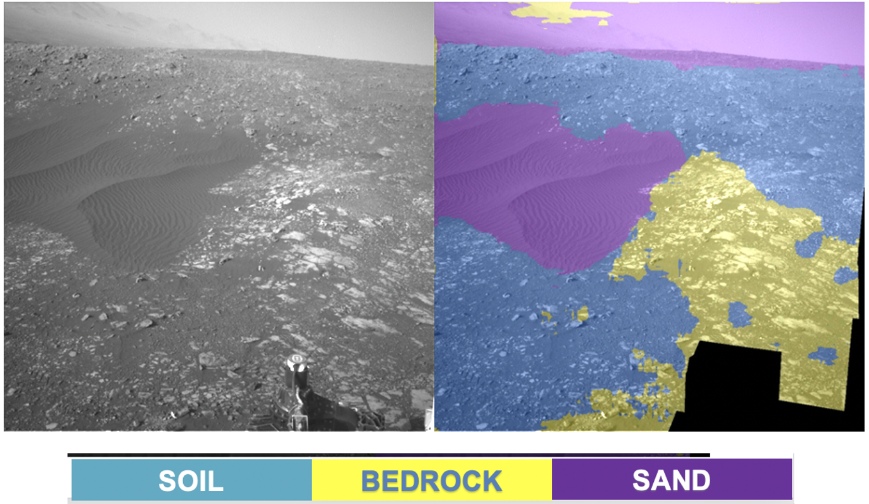
Platform	Classification Discrepancy (out of 1,793 images)	Runtime (per image, 227 x 227 grayscale)	Energy Consumption per image
Linux	0	56.9 ms	
Snapdragon CPU	0	87.8 ms x5	0.5 J x10
Snapdragon GPU	1 (0.06%)	16.3 ms x2	0.051 J x3
Snapdragon DSP/NPU	**15 (0.84%)**	**7.6 ms**	0.016 J

- Classification discrepancies are relative to a laptop run

NavCam: Deep Learning for Terrain Classification

- What: Mars Terrain hazard assessment
- Applications: onboard navigation, science interpretation, slip analysis
- Details:
  - Data: From MSL Curiosity rover Navigation Cameras (NavCams)
  - Model: DeepLabv3 architecture for image segmentation [Seg, NavCam]
  - Developed to support MSL Rover Planners, but could also be run onboard

Mars MSL NavCam Image and Label



- [Seg] Chen, Papandreou, Schroff, Adam, 2017. Rethinking Atrous Convolution for Semantic Image Segmentation, ArXiv.
- [NavCam] Deegan Atha, R. Michael Swan, Annie Didier, Zaki Hasnain, Masahiro Ono. "Multi-mission Terrain Classifier for Safe Rover Navigation and Automated Science," Submitted to IEEE Aerospace Conference, 2022.

NavCam: Classification Discrepancy and Timing

Platform	Total # missed pixels out of 322 images (513 x 513 pixels per image)	Runtime (per image)
Linux	0	1,886 ms
Snapdragon CPU	23 (0%)	6,258 ms /3
Snapdragon GPU	373,464 (0.4%)	2,233 ms /11
Snapdragon DSP (w/ run-time quantization)	**7,867,709 (9.3%)**	**192 ms**

- Were not able to pre-quantize the model so could not run on NPU
  - Using run-time quantization increases network initialization time, peak memory usage, and DLC file size. Accuracy may also be affected.

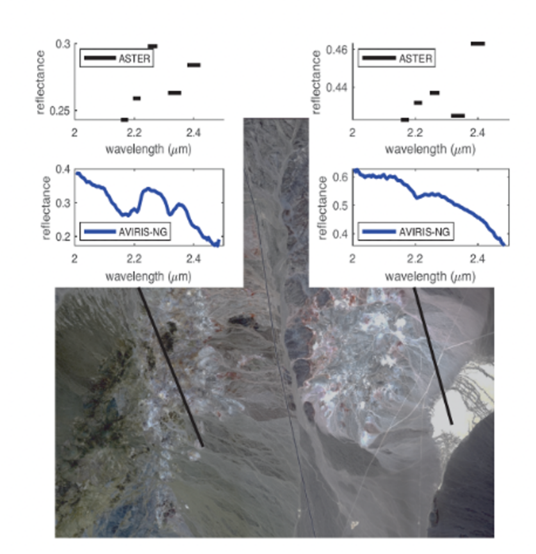
Super Resolution for Spectroscopy

- What: Model for inferring high-resolution data from low-resolution data

- Applications: improved rock and mineral identification for onboard data sub-selection

- Details:
  - Data: 85-band AVRIS-NG data from 5-band ASTER data
    - AVIRIS-NG = Airborne Visible IR Imaging Spectrometer Next Generation
  - ASTER (Advanced Spaceborne Thermal Emmision and Reflection Radiometer) data
  - Model: DCGM (Deep Gaussian Conditional Model) [SRes]

- Runtime:
  - Inference on Snapdragon DSP/AIP is twice as slow as running on the Snapdragon CPU
    - 45 vs 21 ms per input
    - This is most likely due to the small size of the model and single-pixel nature



Remote Spectroscopic Measurements of Cuprite, Nevada [SRes]

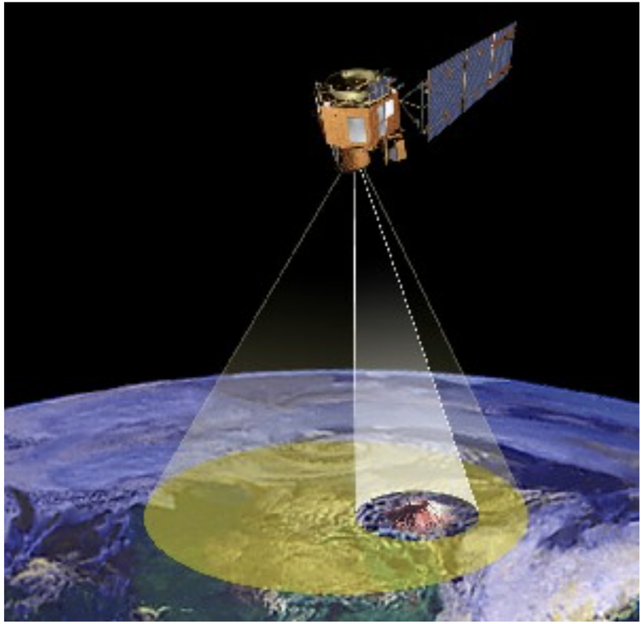
- [SRes] A. Candela, D.R. Thompson, D. Wettergreen, K. Cawse-Nicholson, S. Geier, M.L. Eastwood, R.O. Green, “Probabilistic Super Resolution for Mineral Spectroscopy”. Proc AAAI Conf. on Artificial Intelligence Vol. 34, No. 08, pp. 13241-13247, 2021.

Prior Flights: Machine Learning Inference Onboard

- Autonomous Sciencecraft Experiment (ASE on EO-1)
  - Flew SVM 2005-2017
  - Flew Random Forest, Salience, Bayesian Thresholding 2017
  - Onboard image analysis for intelligent downlink and autonomous retargeting

- Intelligent Payload Experiment (IPEX)
  - CubeSat mission
  - Flew Random Forest, Salience 2013-2014

- PhiSat-1 (2020 - 2021)
  - CubeSat mission from the European Space Agency [ESA]
  - Flew Myriad 2 (previous generation VPU), automatic discard of cloudy imagery



Earth Observing-1 (EO-1) Mission
(ai.jpl.nasa.gov)

Jet Propulsion Laboratory
California Institute of Technology

Flying Other AI

Summary

- General Algorithms
  - Fast Fourier Transform Benchmarks
  - Matrix Multiplication Benchmarks
- Instrument Processing
  - Synthetic Aperture Radar (SAR) Image Formation
  - Hyperspectral Compression
  - High-Order Wavefront Sensing
  - Match Filters
  - SMACC
  - Decision Trees
  - EL Stereo Vision
  - Astro Tipping
  - OWLS
  - Match Filters
- Planning Algorithms
  - MEXEC (Multi-Mission Executive)
  - CLASP (Compressed Large-Scale Activity Scheduling and Planning)
  - Copilot (M2020 Ground Scheduler)
  - SMICES Pointing Planning
  - Cloud Avoidance

Porting to Android

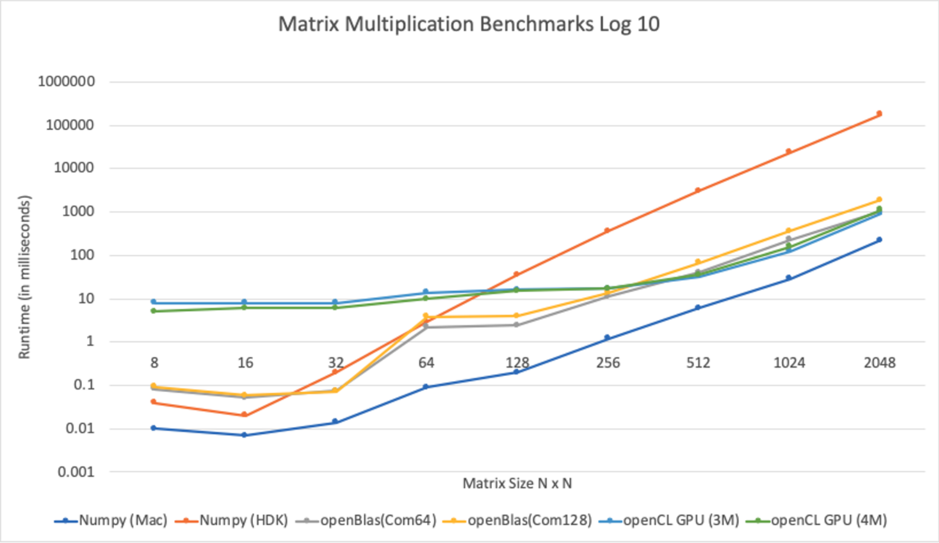
- Most applications involve cross-compiling C/C++ code for ARMv8-A architecture
- ARM only ports are fairly straightforward
  - Some single threaded, some multithreaded
- Applications are ported to GPU, DSP and/or NPU with reasonable effort
- Some applications in python, ported using the python-for-android library provided by Kivy
- Eventual goal is to benchmark all applications on all processors that make sense for that application

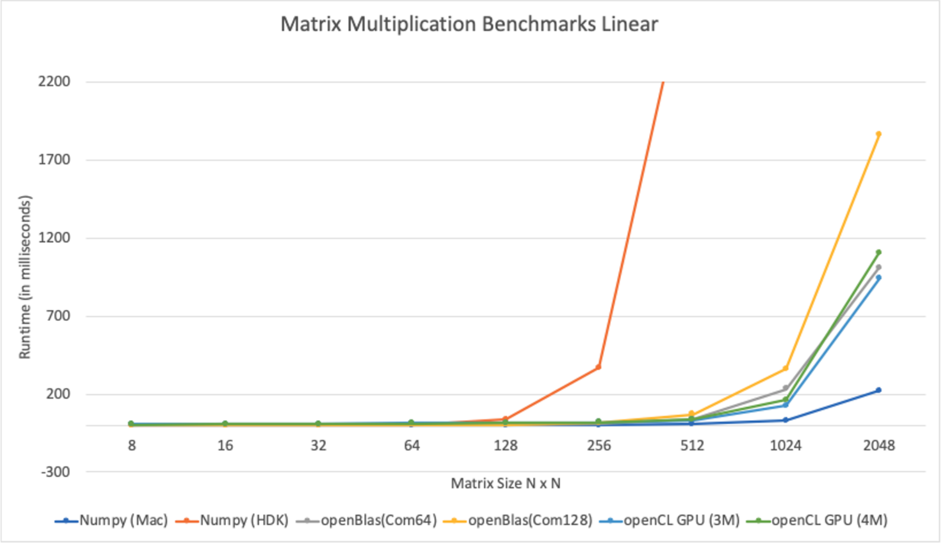
Fast Fourier Transform (FFT)

2X 5X ≈ 1.75X 2.75X

Duration **Power** **Current**	**Float64 ARM**	**Float32 ARM**	**Float32 GPU**	**UINT8 ARM (FASTCV)**	**UINT8 DSP (QHL)**	**UINT8 DSP (QCOM)**
1D FFT (1x2048)	**0.032ms** 2.5W 600mA	**0.025 ms** 2.5W 600mA	**0.025 ms** 2.1 W 525mA	**0.025 ms** 2.5W 600mA	**0.774 ms (1 thread)** 1.50W 420mA	
2D FFT (2048x2048)	**928 ms** 2.4W 550mA	**492 ms** 2.5W 600mA	**105 ms** 2.3 W 575mA	106 ms 2.5W 600mA	125.07 ms (1 thread) 2.00W 500mA 79.58 ms (2 thread) 2.47W 600mA **60.69 ms (4 thread)** 2.76W 670mA	
2D FFT (1024x1024)	**78.11 ms** 3.06W 770mA	**63.18 ms** 3.00W 766mA	**23.436 ms** 2.35W 590mA 20% GPU Utilization	**23.56 ms** 3.00W 766mA	26.33 ms (1 thread) 2.0W 500mA 17.00 ms (2 thread) 2.32W 600mA **14.14 ms (4 thread)** 2.60W 640mA	**5.128 ms (4 thread)** 2.40W 612mA

Matrix Multiplication

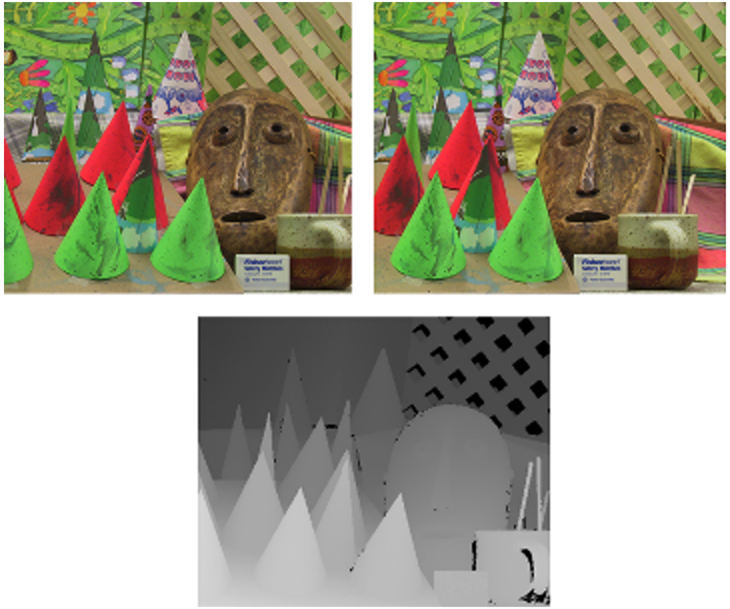




Generic benchmarks: more extensive parameter changes relating to different libraries and input data sizes.

Europa Lander Stereo Vision

- Application determines the relative range of objects in a particular image is using a set of images from a stereo vision camera
- Success scored on the measurement of the sum of absolute differences (SAD) between five patches in the two images
- Implemented on the Snapdragon ARM
- Dataset: 24 image pairs
  - Each image is natively at 5120x3840
- Images processed at three different resolutions:
  - Level 0: 5120x3840: 15.6 minutes
  - Level 1: 2560x1920: 2.1 minutes
  - Level 2: 1280x960: 19 seconds



Stereo Vision Depth Map Example

Stereo Vision

	Snapdragon CPU		Snapragon GPU	Snapdragon DSP		RAD750
	SISD	NEON		SISD	HVX	Estimate
Runtime (ms)	95.78	**39.24**	244	172.92	**42.40**	12400
Power (W)	4.55	**4.57**	2.00	1.80	**2.35**	10
Energy (J)	0.44	**0.18**	0.49	0.31	**0.10**	124

- Stereo vision runtimes over one 384x512 pixel image
- Run over the Snapdragon CPU, GPU, DSP, and a RAD750
- Further improvement possible on the Snapdragon by utilizing Qualcomm's libraries

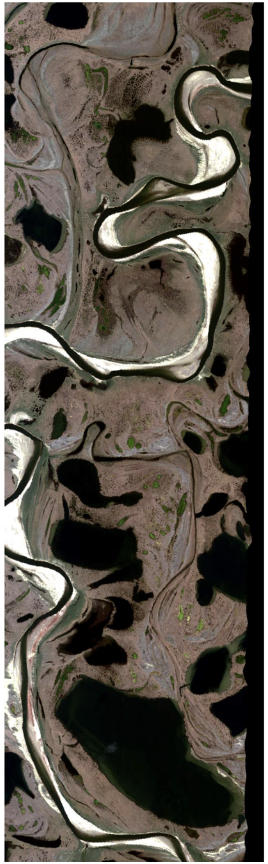
Decision Trees - Thermal Anomaly Detection

- Decision tree logic to detect high thermal emissions (lava, wildfires) using radiance band ratios
  - (Human expert constructed trees Davies et al. 2006 RSE)
- Classifies into hot and extremely hot pixels utilizing multiple radiance bands
- Run over Aviris-NG data
  - Image size 4500x390x425
- Implemented on the ARM and GPU of the Snapdragon
  - ARM runtime: 21s (cryosphere and lava)
  - GPU runtime: 13s (cryosphere and lava)

Skysat image of the Fagradalsfjall volcano in Iceland

Decision Trees Cryosphere Detection

- Decision tree based on cryosphere detection
  (human expert constructed tree Doggett et al. 2006 RSE)
- Classifies clouds, water, snow, ice, and land within an image
- Run using AVIRIS-NG data over Alaska
  - Image size 4500x390x425
- Implemented on the ARM and GPU of the Snapdragon
  - ARM runtime: 21s (cryosphere and lava)
  - GPU runtime: 13s (cryosphere and lava)



AVIRIS-NG
image of Alaska

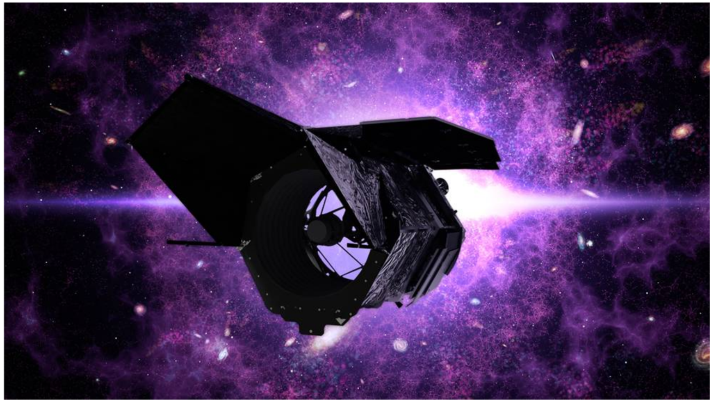
Random Decision Forest - Thermal Anomaly

- Random Forest to detect thermal anomalies in images using radiance band ratios
- Classifies into the binary classes of thermal anomaly and not a thermal anomaly
- Run over Planet Skysat data (4 bands: Red, Green, Blue, NIR)
- Image labels were generated from radiance band ratios established by Ashley Davies
- Training Dataset: 13 images
  - Image size: ~ 4x10,000x12,500, pixel size = 50cm
  - 11 of Fagradalsfjall
  - 2 of Kilauea
- Test Dataset:
  - 4,000,000 pixels from a Fagradalsfjall image
- RDF: number of trees: 1,300 trees (100 trees trained on each training image), max depth: 10, weights equalized by class
- Runs single threaded on the CPU
- Runtime: 10.5 minutes

Skysat image of the Kilauea volcano in Hawaii

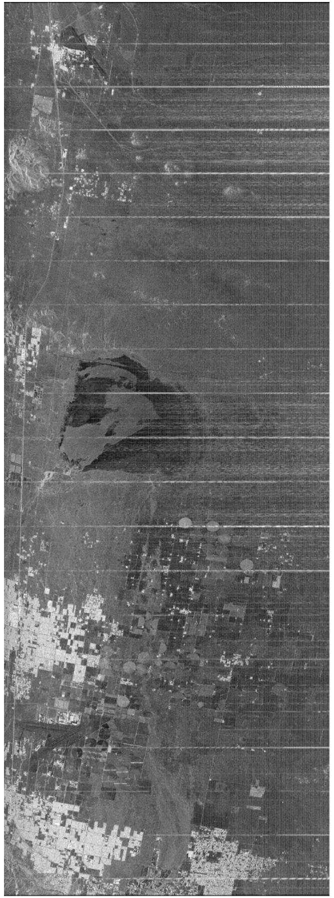
High Order WaveFront Sensing (HOWFS)

- Python implementation of HOWFS for Roman Space Telescope Coronagraph Instrument
  - Onboard processing would facilitate the mission
  - Faster processor is necessary for onboard processing
- Single threaded non-optimized port - Currently a little slow
  - Original double precision code took 2.2 hours
  - Moving to single precision took 1.8 hours
- Further Work
  - GPU
  - Multi-thread
  - Move from Python (allow easier use of specialized hardware)



Synthetic Aperture Radar (SAR) Image Formation

- Image Processing pipeline from Uninhabited Aerial Vehicle SAR (UAVSAR) instrument
- A pipeline of 3 ARM applications, 2 GPU applications
    - Mainly a row-wise and column-wise 2D FFT with filters applied
- Goal of <240 Seconds (~ real time)
- Image Size: 27916x26880
- Currently takes 217 seconds
    - Could possibly be further improved, as GPU usage is only at about 60%

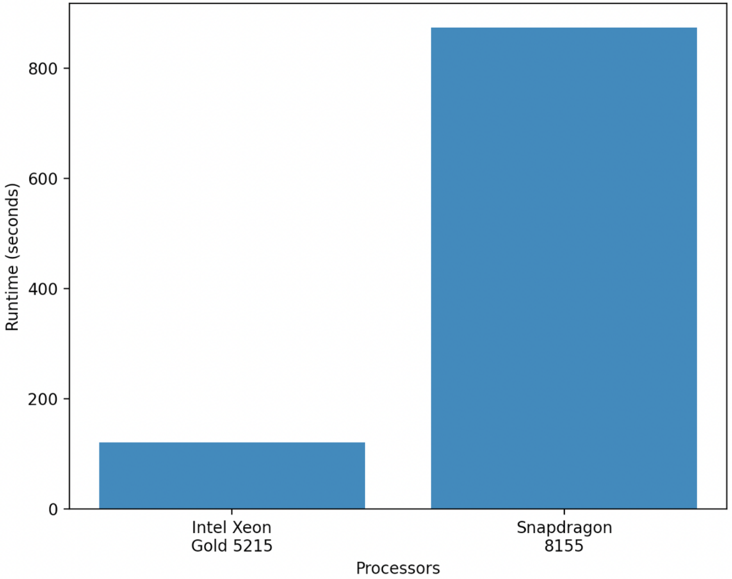


The Rosamond Calibration Array (RCRA), located near the south beach of Rosamond Dry Lake Bed in California.

Match Filters

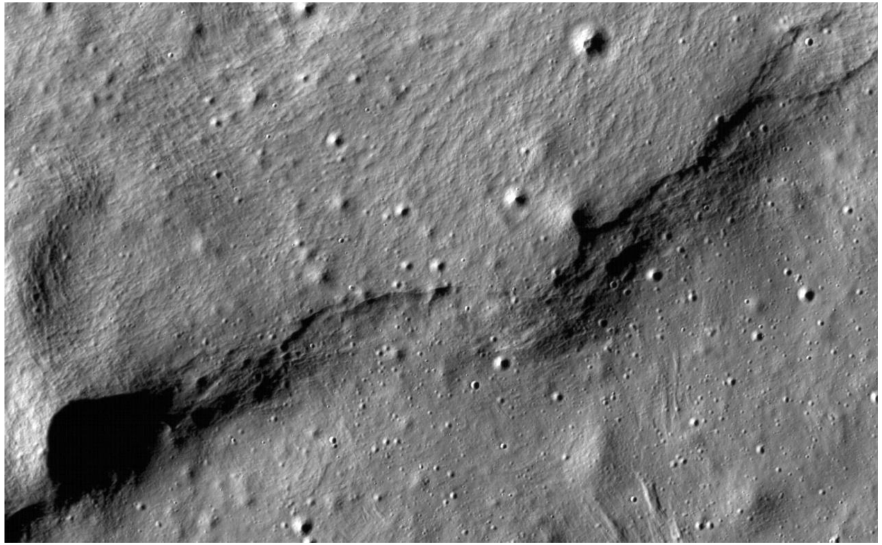
- Running on images from the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS)
  - Images of Cuprite site in 2014
- Currently running only Kaolinite, Calcite and Alunite detection
  - Plan to expand to ~20 minerals
- Runs on ARM only, single threaded
- Much of the runtime is I/O, so multithreading has modest effect
- Image Size: 670x2512x425
- Runtime through 8 images on 1 mineral on the graph
  - ~850s on Snapdragon ARM

https://www.jpl.nasa.gov/missions/airborne-visible-infrared-imaging-spectrometer-aviris



Lunar Match Filters

- Running on a dataset of modified M3 imaging spectrometer images of the Karpinsky feature of the moon
- Aims to demonstrate the detection of volatile water molecules on the lunar surface
    - Currently running on different forms of water (OH, molecular H_2O, and H_2O ice)
- Runs on ARM only, single threaded
- Performance mirrors Match Filters
- Image size 304x1000x301
- Runtime 108.4s



https://photojournal.jpl.nasa.gov/catalog/PIA23236

Hyperspectral Compression

- Benchmarked on test images for Earth Surface Mineral Dust Source Investigation (EMIT)
- 64 lines, 640 samples per line, 481 spectral bands
- ARM only port, GPU port in progress
- MSamples/sec = lines * samples per line * bands / runtime
- EMIT Target is 23.1 MSamples/sec (near real time)

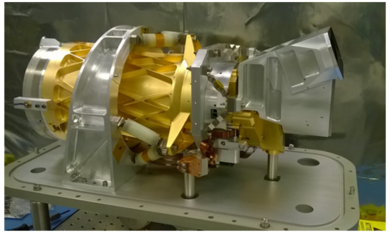
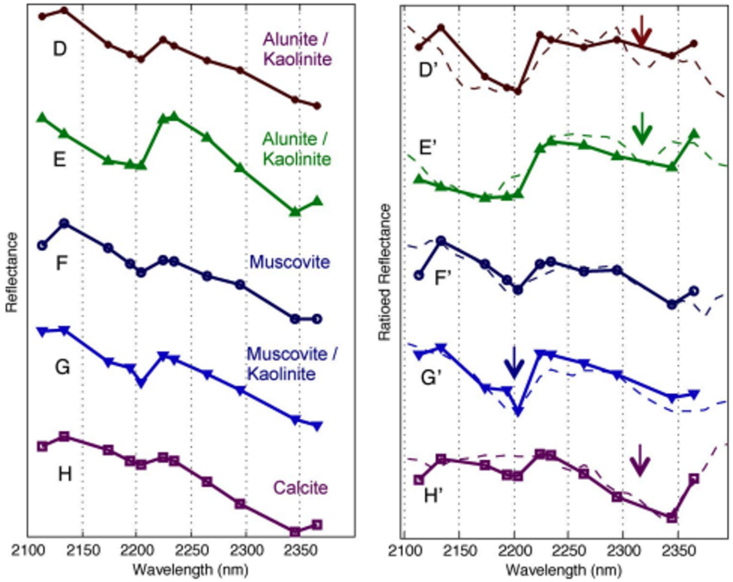


Image from https://earth.jpl.nasa.gov/emit/instrument/overview/

	CPU	GPU	DSP	Virtex-5	GTX 580
CCSDS Standard	123.0.B-2	123.0.B-1	123.0.B-2	123.0.B-1	123.0.B-1
Compression	Lossless				
Runtime (ns)	14.12	6.5	184.5	25	16.18
Sample Rate (MSamples/sec)	70.82	153.85	5.42	40.00	61.80
Power (W)	6.1	3.5	1.9	2	>100

SMACC

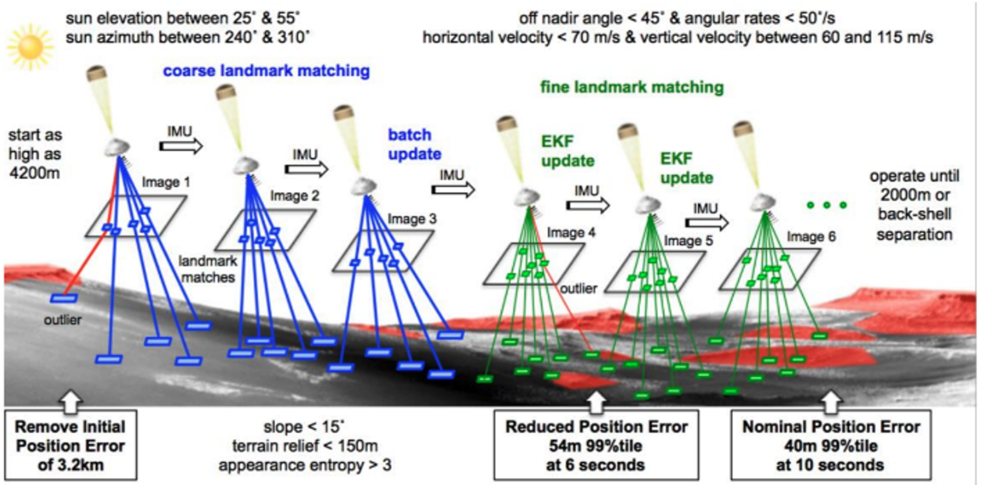
- Sequential Maximum Angle Convex Cone (SMACC) spectral endmember extraction
- Snapdragon implementation run with AVIRIS-NG Data
- Extracts the top 5 endmembers from the data
- Can be used to extract radiance values that match with minerals and other materials
- Image Size: 638x679x425
- Runtime: 16.9 seconds
- Based on D. Thompson et al. 2012 TGARS



Example endmember reflectance values compared to mineral reflectance base values

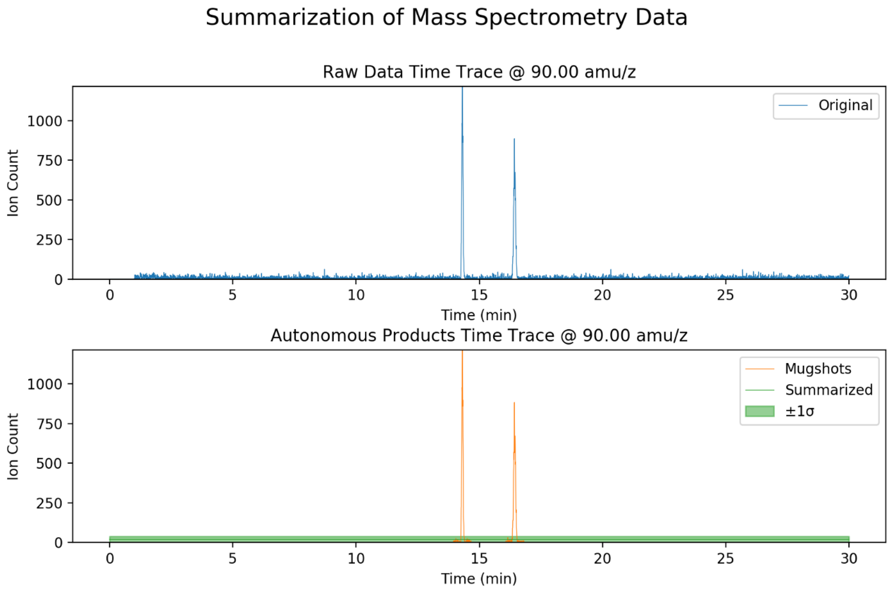
Landing Vision System/Astrotipping

- Landing Vision System currently implemented as a hybrid FPGA + Processor solution
- Problem divided into COARSE and FINE modes
- COARSE
  - Run on 1024x1024 image
  - Image Warping
  - Match to template (FFT)
  - Runtime: 2.46s
- FINE
  - Run on 1024x1024 image
  - Normalized Cross Correlation
  - Runtime: 2s
- Implemented on the Snapdragon ARM



Ocean Worlds Life Surveyor (OWLS) autonomous algorithms

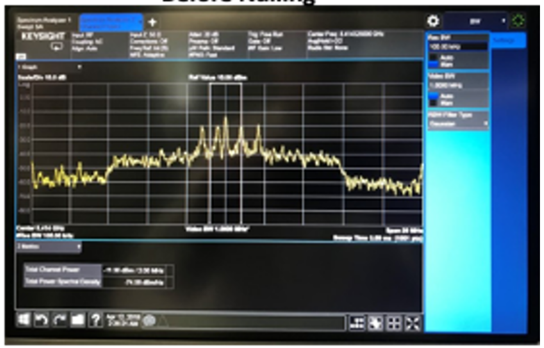
- Project aimed at autonomously detecting signatures of life in water at the molecular and cellular scale
- Algorithms process data from different instrument bands from instruments aboard the OWLS instrument
  - They process and prioritize autonomous data science products (ADSPs) to send the most scientifically viable subsets of information back
- Implemented on the ARM of the Snapdragon
  - Multithreaded version is in progress
- Dataset: 24 images, 1024x1024
- Runtime: 22.5s



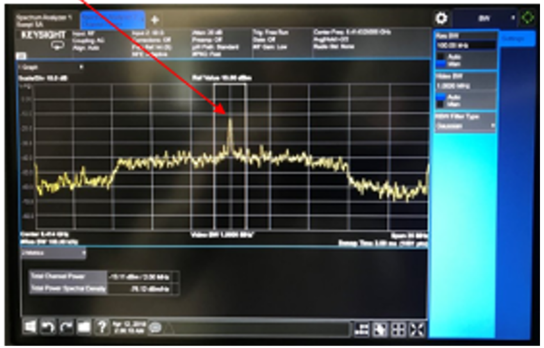
ESPRIT

- ESPRIT is a method to detect signal interference
  - Algorithm requires eigen-decomposition of 20x20 to 100x100 matrix.
- The compute capability required for ESPRIT was too prohibitive with the 50MHz SPARC (LEONv3) softcore available on the modem FPGA.
- Snapdragon can detect this interference and configure nulling filters present on the modem FPGA (or in software for software defined radios).
- Implemented in Python
- Runs single threaded on the CPU
- Runtime: 20 minutes

Before Nulling



Carrier

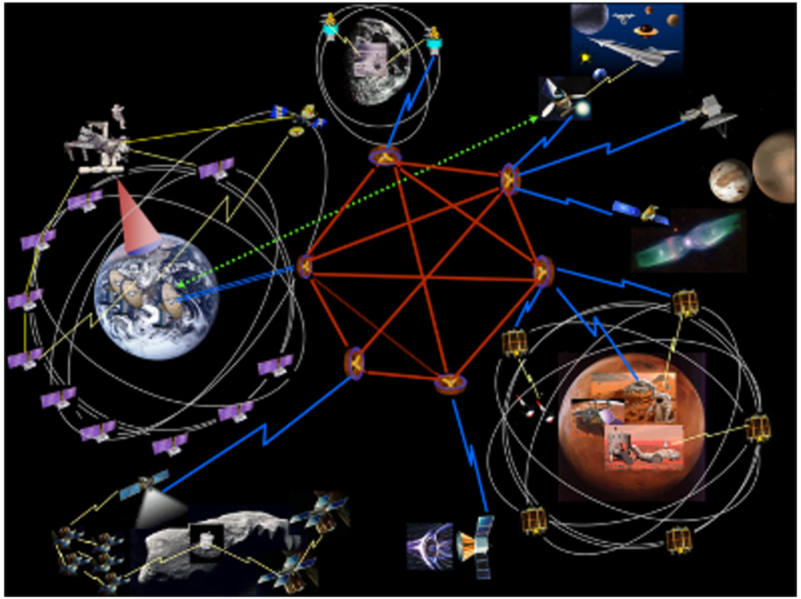


ESPRIT Runtimes

Config. ID	LEON3 No FPU, No HW MUL/DIV	LEON3 with FPU, no HW MUL/DIV	LEON 3 with FPU and HW MUL/DIV	Snapdragon 8155 (Single Threaded)	Snapdragon 8155 (Multi Threaded)
A	150 s	74 s	15 s	1.320 s	0.477 s
B	60 s	23 s	6 s	1.049 s	0.441 s
C	36 s	14 s	4 s	0.988 s	0.425 s
D	24 s	9 s	2 s	0.959 s	0.418 s
E	**5 min**	**?**	**?**	**2.841 s**	**0.587 s**

- Comparison of Snapdragon performance to LEON 3 processors
- Dataset comprised of recordings of tone interference on the uplink of an Iris radio

Delay Tolerant Networking (DTN)



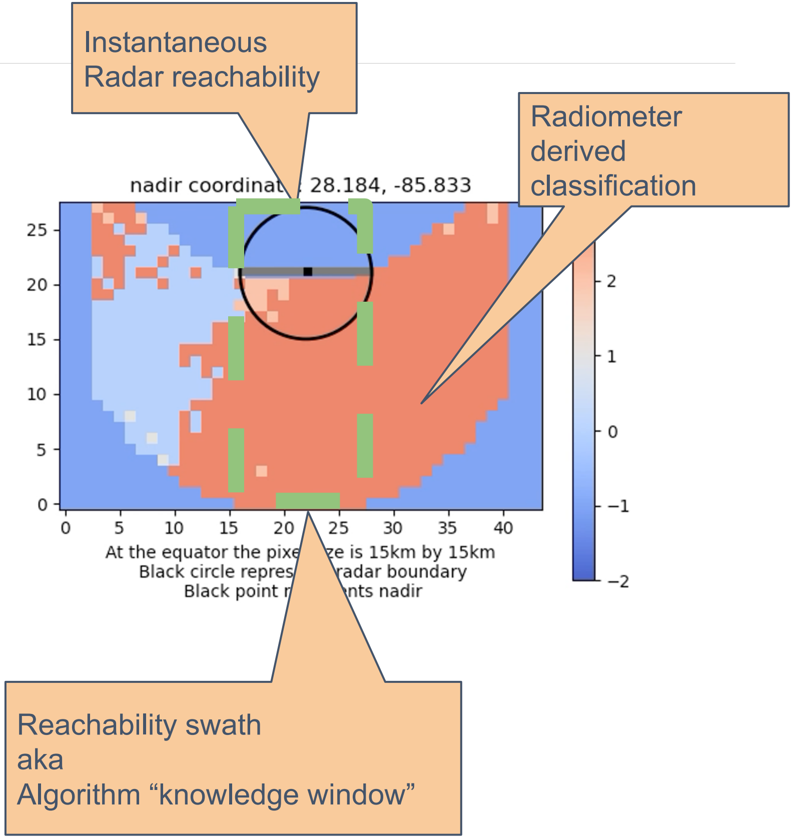
- The additional memory space and compute power of the Snapdragon makes it possible to implement the delay tolerant networking (DTN) algorithm onboard
    - Algorithm works by traversing a graph to setup a relay network, and to find the next node to forward data to
    - Uses Dijkstra’s algorithm to identify the best nodes to forward data
- Implemented in Python
- Runs single threaded on the CPU
- Runtime: 35 minutes

Planning/Scheduling FSW on Conventional Flight CPU's

- AI-based Planners have flown onboard spacecraft as early as 1999
- Remote Agent Experiment onboard Deep Space 1 flew for 48 hours *(Jonsson et al. 2000)*
  - Demonstrated closed-loop, goal-based commanding and batch replanning onboard a RAD 6000 PPC
- The Autonomous Sciencecraft Experiment flew CASPER on Earth Observing 1 for over 12 years (2003 - 2017) *(Chien et al. 2005)*
  - Encountered limited observability, limited memory, and limited CPU constraints onboard a Mongoose V (enhanced version of RAD 3000; no FPU)
  - “Continuous replanning” in response to state changes and goal adds/deletes
- Intelligent Payload EXperiment Cubestat (IPEX) in 2013 flew CASPER *(Chien et al. 2016)*
  - Used an 200 MHz Atmel ARM9 (no FPU) running Linux
- ASTERIA was a 6U Cubesat running Linux on a Cortex 160 *(Troesch et al. 2020)*
  - Ran several MEXEC scenarios, encountered similar RAM and CPU limitations

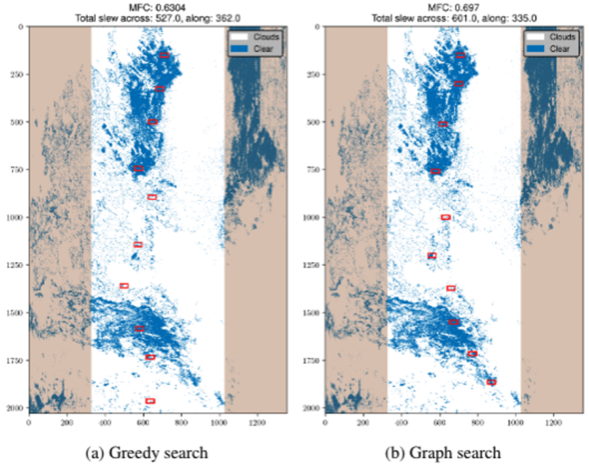
SMICES Pointing Planning

- SMICES Point Planning is the Smart Ice Cloud Sensing (SMICES) storm targeting planner
- Running on the Caribbean region of a Global Weather Research and Forecasting (GWRF) model's simulated data
- Uses a radiometer to identify cloud types in the orbit path and schedules a radar for further imaging
- Scientists define the value of each cloud type
- Generates an observation list based on the constraints of the radar and available clouds
- Run over 15,232 timesteps (1 timestep = 2 seconds, ~8.5 hours simulation time)
- Runtime of 53.6 seconds on Snapdragon ARM



Cloud Avoidance

- Algorithms developed to avoid imaging clouds while in orbit given a lookahead and instrument view
- Run over 50 images with dimensions 1354x2030
- Four algorithms developed
  - Greedy search
    - Median Runtime Per Image: 49.7 ms
  - Graph search
    - Three Implementations
      - Adaptive Grids
        - Median runtime per image: 199.2 ms
      - Mixed Grids
        - Median Runtime Per image: 13.7 ms
      - Fixed Grids
        - Median runtime per image: 9.7 ms
- Implemented in Rust for the Snapdragon ARM



Visualizations of the two cloud avoidance algorithms

MEXEC

- Separately threaded Planner and Executive
- Takes “Task Network” as input
    - Set of state timelines, task templates, and tasks
- Generates conflict free plans and monitors task execution, responding to deviations or exogenous events

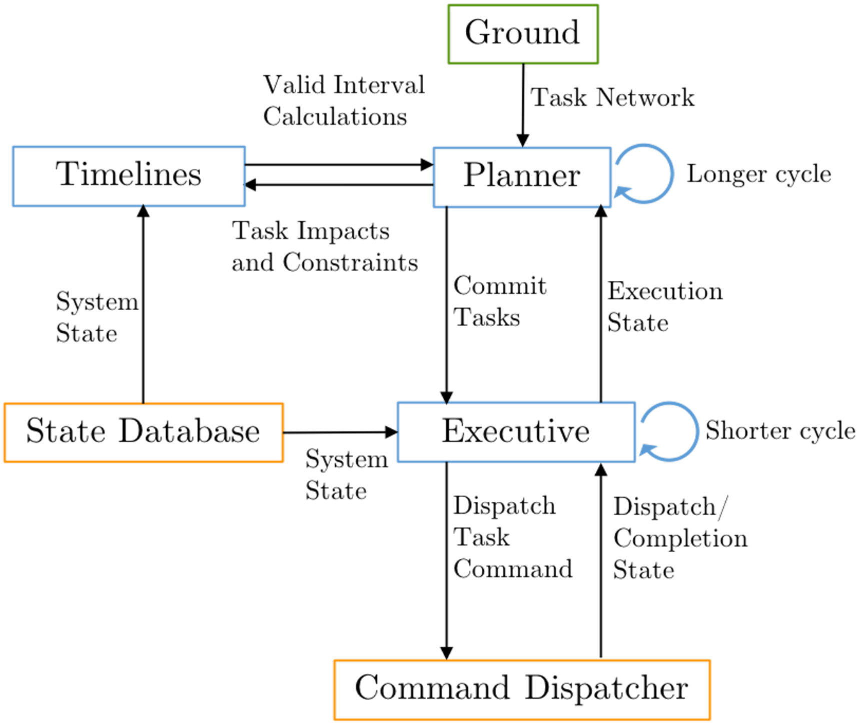
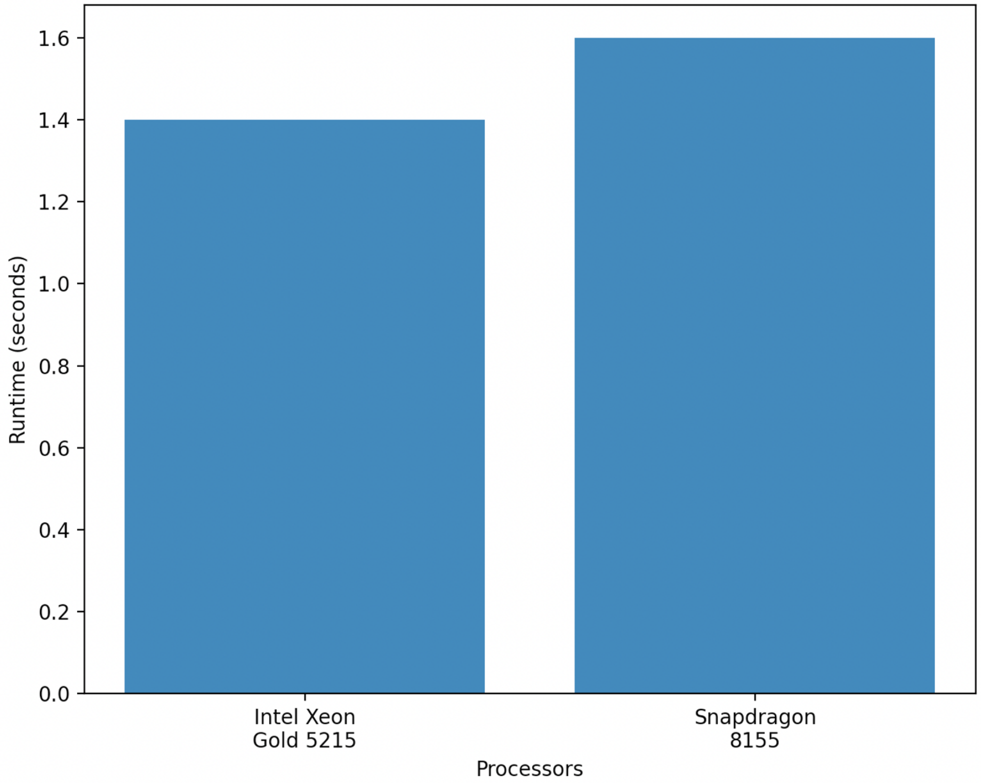


Image from https://ai.jpl.nasa.gov/public/projects/mexec/

MEXEC Benchmark Scenario

- MEXEC consists of multiple components, but the most computational demanding is the planner, so that is used for benchmarking purposes.
- MEXEC also runs continuously on a cycle, for benchmark purposes we only time the first plan generation.
- As a benchmark, we use the Europa Lander Prototype test scenario *(Wang et al. 2020)*
- Multi-day schedule, exercises hierarchical planning, valid interval search, constraint satisfaction, etc.
- Running as a single threaded ARM application

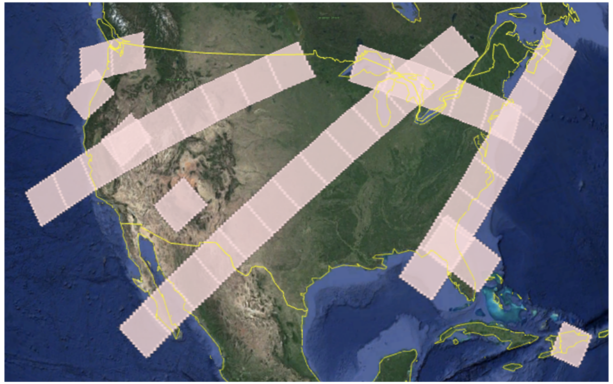
MEXEC Results



Sabertooth (scaled to final clock speed)	Snapdragon 8155 ARM
92s	**1.6s**

CLASP

- CLASP is the Compressed Large-scale Activity Scheduler and Planner
- CLASP has been used for NISAR, ECOSTRESS, EMIT, OCO3, and other missions.
- Spacecraft, instrument, and orbiting body models define the scenario
- Science Campaigns define the scientific goals
- CLASP generates an observation schedule based on the scenario constraints

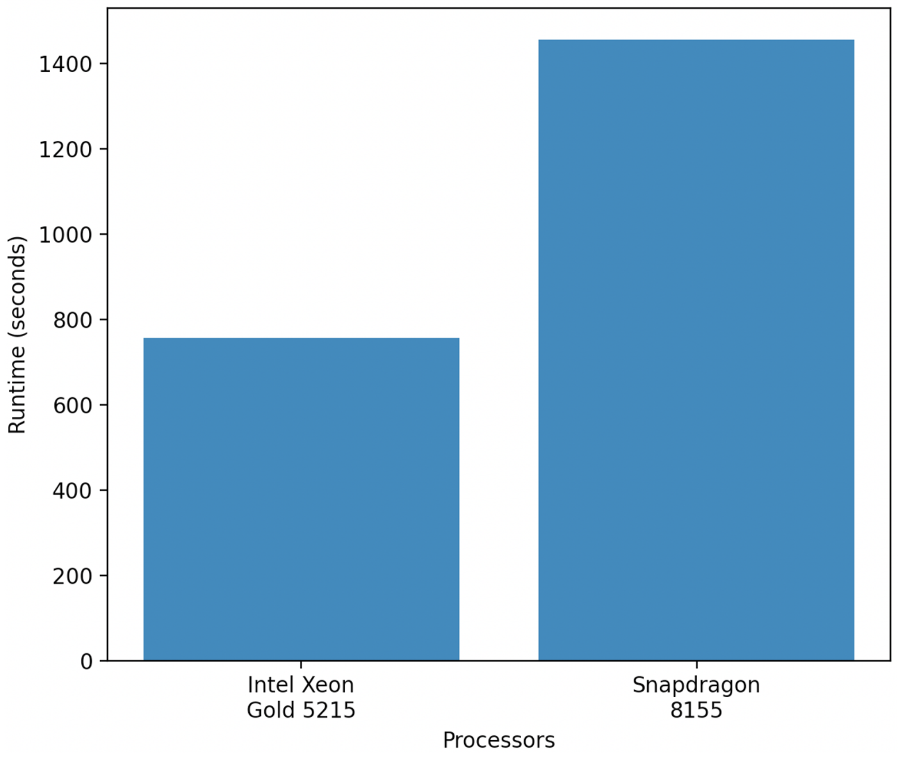


ECOSTRESS schedule portion of the Contiguous United States

CLASP Benchmark Scenario

- For our benchmark, we generate 2 years of 2 week schedules using ECOSTRESS data from 2018-2020
- We generate a single 2 week schedule to collect our timing metric faster
- Currently CLASP is single threaded on the Snapdragon ARM
    - CLASP performs a large amount geometric reasoning to compute feasible observations (target visibility) which could benefit greatly from GPU acceleration (in progress). Visibility computations are independent in overflight time so can be parallelized in that dimension.

CLASP Results



Copilot

- M2020 ground scheduler; currently in use for M2020 operations for scheduling wake/sleep and preheats
- Uses the same scheduling algorithms as the M2020 onboard scheduler
- Challenges include wake/sleep constraints, preheat constraints, variability in execution, and complex operations handover handling.

Plan for sol type “medium drive with post drive imaging”

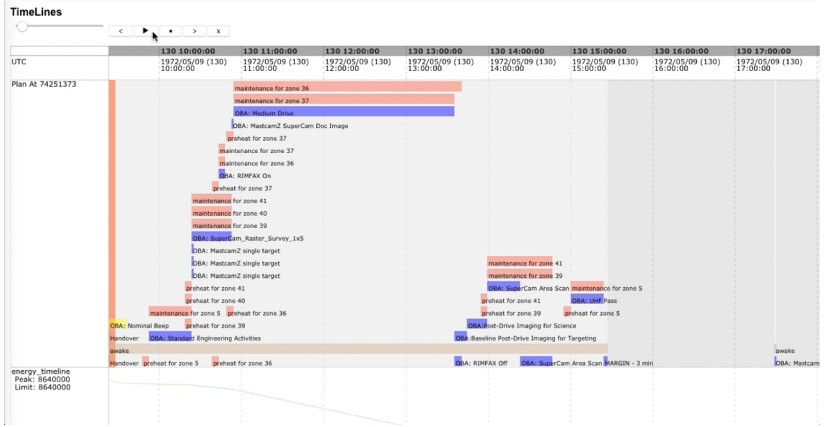
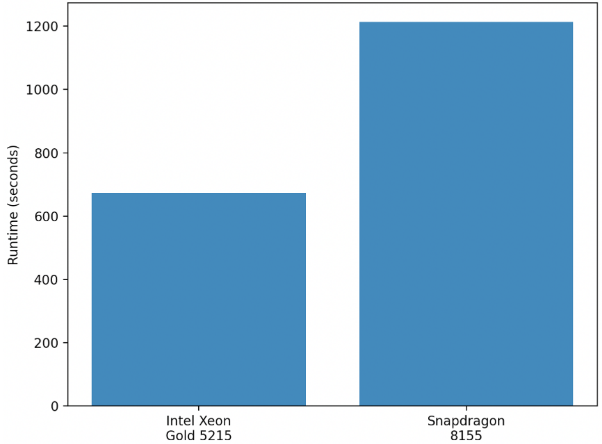


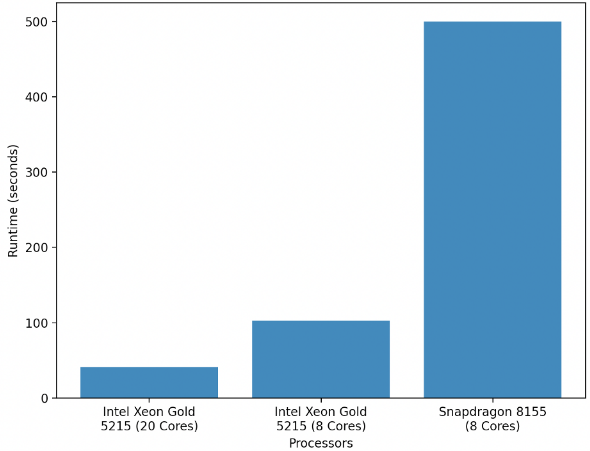
Image from https://ai.jpl.nasa.gov/public/projects/m2020-scheduler/

Copilot Benchmark Scenario

- For this benchmark, we are running with ~800 x 1 martian day (or sol) planning problems that are generated by random variation of 7 base plans or “sol types”
    - Vary execution durations, incoming/outgoing energy state, and alternative action options
- Copilot has a single threaded and multithreaded version
    - Large problem already split into 800+ small problems, so easy to parallelize
- Benchmarked against SBC2 using 1 core, 8 cores (to match 855) and 20 cores

Copilot Results





- Runtime of processors on the Copilot benchmark problem
  - Top: Serial
  - Bottom: Parallelized
- On the Intel, all 10 cores 2.5GHz
  On the Snapdragon, 8 cores range from 1.8-2.8 GHz
- Generates 800 variants, so Snapdragon takes <1s per generated plan in parallelized results

Future Work

- Measure additional computing performance metrics
  - Power/Energy Consumption
  - RAM Footprint
- Benchmark on additional processors:
  - **LEON4 Sabertooth**, **RAD/PPC 750**, NVIDIA Jetson Nano, and more
- Parallelize Applications across multiple cores
- Make greater use of hardware acceleration
  - GPU, DSP, NPU as applicable
- Port more applications - see next slide

Future Planned Applications

- Normalized Difference Index Science Product Generation
- Volcanic plume detection
- TIR (ECOSTRESS) Data Processing Pipeline
- VSWIR (EMIT) Science Data Processing Pipeline
- Sensorweb Tasking
- Surface Wa980—ter Extent (SWE)
- and more!

Conclusion

- Ported machine learning, instrument processing, scheduling, and benchmark applications
- Benchmarked above on problems on
  - **Qualcomm Snapdragon 855**
  - **Intel Movidius**
  - Intel Linux Benchmark
  - Sabertooth
  - Rad 750
  - NVIDIA Jetson Nano

in progress!

- Working towards measuring runtime performance, power/energy, RAM against other flight processors
- Work intended to facilitate future flight of these capabilities to enable future single and networked autonomous spacecraft missions.

Team

- Jet Propulsion Laboratory, California Institute of Technology
  - Faiz Mirza
  - Jason Swope
  - Dr. Emily Dunkel
  - Evan Davis
  - Dr. Zaid Towfic
  - Dr. Damon Russell
  - Dr. Joseph Sauvageau
  - Dr. Douglas Sheldon
  - Dr. Steve Chien
  - Dr. Dennis Ogbe
  - Lauren West
- Ubotica
  - Juan Romero-Cañas
  - Dr. José Luis Espinosa-Aranda
  - Léonie Buckley
  - Elena Hervas-Martin
  - Fintan Buckley
- Hewlett Packard Enterprise
  - Dr. Mark Fernandez
  - Carrie Knox

We would also like to thank many of the applications providers who also supported the testing of their applications.

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov