Jet Propulsion Laboratory
California Institute of Technology

Next Generation Flight Computing

A Joint Investment of NASA and AFRL

Richard Doyle
Jet Propulsion Laboratory, California Institute of Technology

presentation at
2013 Workshop on Spacecraft Flight Software
Pasadena, CA

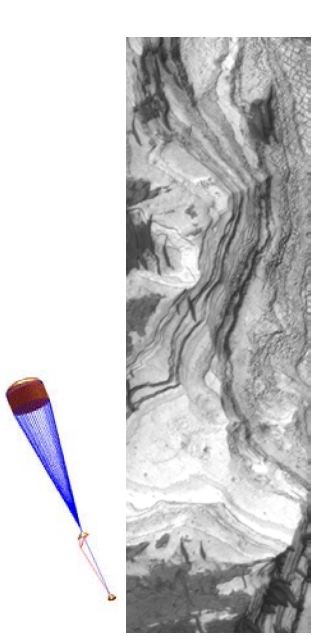
December 12, 2013

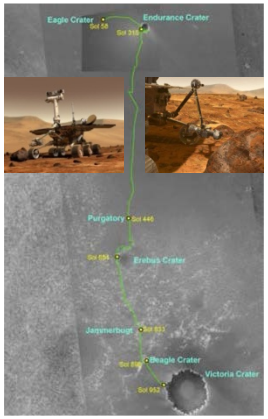
Air Force Research Laboratory:	Gabriel Mounce, Keith Avery *et al*
NASA Ames Research Center:	Michael Lowry *et al*
NASA Goddard Space Flight Center:	Wesley Powell, Michael Johnson *et al*
Jet Propulsion Laboratory:	Raphael Some, Kim Gostelow, John Lai, Larry Bergman, William Whitaker *et al*
NASA Johnson Space Center:	Montgomery Goforth *et al*

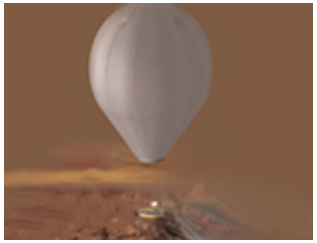
The NASA Need

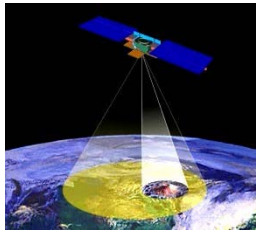
There are important mission scenarios that today cannot be accomplished robustly and cost-effectively

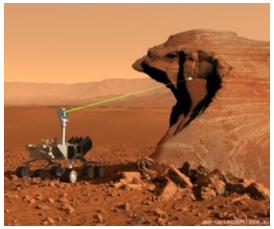
- Space-based computing has not kept up with the needs of current and future NASA missions
- Government and industry are developing high-performance space-qualifiable processors
- NASA continues to have unique requirements
  - Deep space, long duration, robotic and human missions
  - Higher performance, smaller spacecraft and lower cost
    - Onboard science data processing
    - Autonomous operations in uncertain environments
  - Extreme needs for low power and energy management, efficiency, fault tolerance and resilience





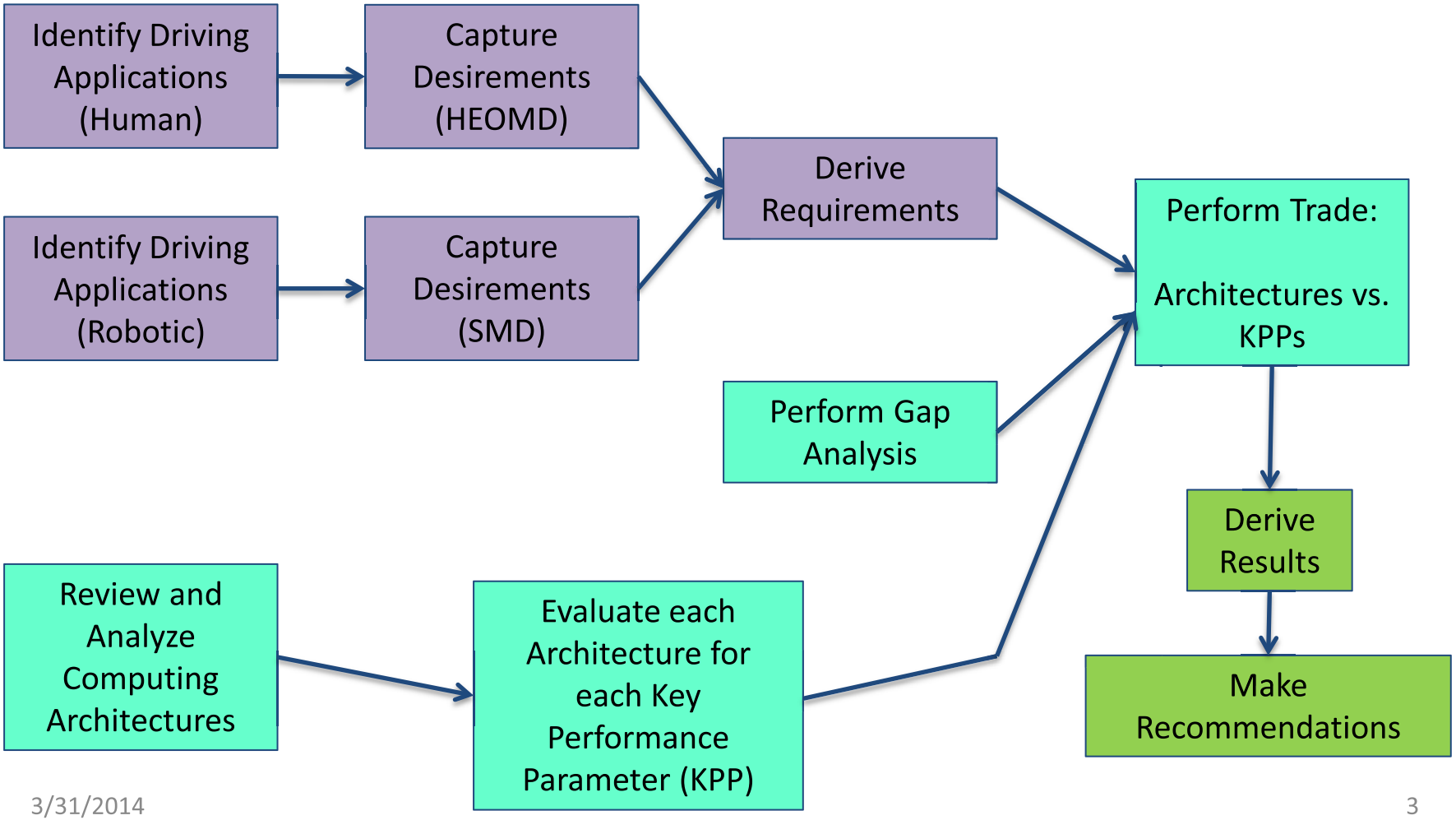






NASA Space Technology Program

Authorization of High Performance Spaceflight Computing (HPSC) Study under Game Changing Development Program (GCDP)



HPSC Formulation Phase Study Summary

General-purpose Multi-core provides optimum ROI for NASA

The Assignment	The Results
Identify relevant NASA use cases What are the paradigm-shifting NASA space-based applications that will drive next generation flight computing?	Developed 9 human spaceflight (HEOMD) and 10 science mission (SMD) use cases for future flight computing, spanning critical mission functions, high data rate instruments, and autonomy utilizing model-based reasoning techniques
Derive requirements What are the future onboard computing requirements?	100X performance increase, low power (down to 7W) with scaling, support for a range of fault tolerance, common programming languages, avoidance of additional V&V effort, interoperable with co-processors
Perform a gap analysis How/where do commercial and defense industry developments in computing fall short of NASA's unique requirements and architectural needs?	No existing or emerging spaceflight processors possess all necessary performance, power efficiency, reliability, and programmability attributes
Trade architectures against defined Key Performance Parameters (KPPs) Which computing architecture will make the most difference?	Rad-hard general-purpose multi-core best addresses the future flight computing requirements and presents the most affordable gap against the KPPs
Make a recommendation How can NASA best invest limited resources to meet the future needs of its space systems?	Competed/directed program plan for rad-hard general-purpose multi-core, with solutions for power/energy, fault tolerance and other NASA requirements, leveraging other agency and industry investments

Study specifics in following charts

NASA Applications

for High Performance Spaceflight Computing

HEOMD Use Cases

1. Cloud Services
2. Advanced Vehicle Health Management
3. Crew Knowledge Augmentation Systems
4. Improved Displays and Controls
5. Augmented Reality for Recognition and Cataloging
6. Tele-Presence
7. Autonomous & Tele-Robotic Construction
8. Automated Guidance, Navigation, and Control (GNC)
9. Human Movement Assist

SMD Use Cases

1. Extreme Terrain Landing
2. Proximity Operations / Formation Flying
3. Fast Traverse
4. New Surface Mobility Methods
5. Imaging Spectrometers
6. Radar
7. Low Latency Products for Disaster Response
8. Space Weather
9. Science Event Detection and Response
10. Immersive Environments for Science Ops / Outreach

High value and mission critical applications identified by NASA scientists and engineers

Science Mission Applications

10X+ improvement for existing applications

Enable new science and mission capabilities on future missions

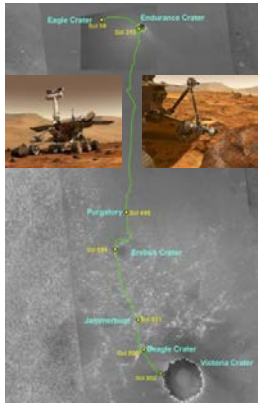
SMD Use Cases

1. **Extreme Terrain Landing**
2. Proximity Operations / Formation Flying
3. **Fast Traverse**
4. New Surface Mobility Methods
5. Imaging Spectrometers
6. Radar
7. Low Latency Products for Disaster Response
8. Space Weather
9. **Science Event Detection and Response**
10. Immersive Environments for Science Ops / Outreach

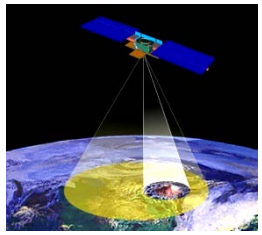
Benefits to Missions

- **Extreme Terrain Landing**

  - Enables reliable and safe landing in hazardous terrain: TRN and HDA algorithms benchmarked by Mars Program – required six (6) dedicated RAD750s



- **Fast Traverse**
  - Remove computation as a limiting factor to mobility – drive 10X faster and more safely
- **Science Event Detection and Response**
  - Increase capture rate for dynamic, transient events from ~10% to >75%, with <5% false positives, for increased and more timely science return



No longer need to size science / mission scope to flight computing capability

Human Spaceflight Mission Applications

Enable autonomous human-assist capabilities for next generation crewed vehicles and missions

HEOMD Use Cases

1. Cloud Services
2. **Advanced Vehicle Health Management**
3. Crew Knowledge Augmentation Systems
4. Improved Displays and Controls
5. Augmented Reality for Recognition and Cataloging
6. Tele-Presence
7. **Autonomous & Tele-Robotic Construction**
8. **Automated Guidance, Navigation, and Control (GNC)**
9. Human Movement Assist

Benefits to Missions

- **Vehicle Health Management**
  - Continuous monitoring/analysis of large vehicle data sets: problem detection and response, crew workload reduction, and improved vehicle maintenance during untended operations

- **Crew / Robot Interaction**
  - Robots respond to high-level instructions from crew or ground personnel while maintaining safe operations and interactions with the crew

- **Automated GNC**
  - Move compute-intensive GNC applications onboard for faster, safer docking; close proximity operations; collision avoidance; automated precision landing within an affordable power/propulsion budget

No longer need to size science / mission scope to flight computing capability

NASA Flight Computing High-Level Drivers

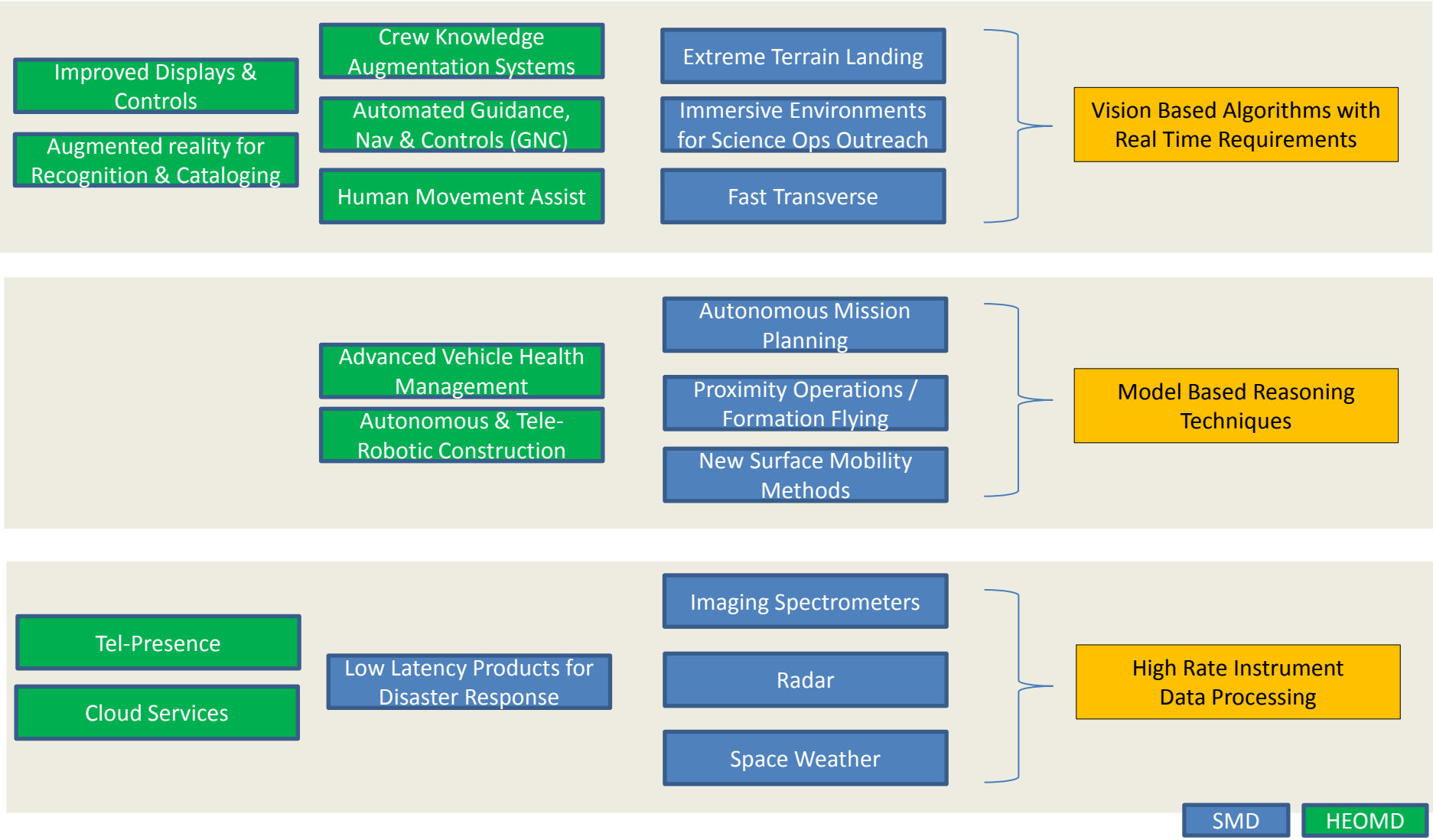
as derived from the NASA use cases

Computation Category	Mission Need	Objective of Computation	Flight Architecture Attribute	Processor Type and Requirements
Vision-based Algorithms with Real-Time Requirements	• Terrain Relative Navigation (TRN) • Hazard Avoidance • Entry, Descent & Landing (EDL) • Pinpoint Landing	• Conduct safe proximity operations around primitive bodies • Land safely and accurately • Achieve robust results within available timeframe as input to control decisions	• Severe fault tolerance and real-time requirements • Fail-operational • High peak power needs	**• Hard real time / mission critical** **• Continuous digital signal processing (DSP) + sequential control processing (fault protection)** **• High I/O rate** **• Irregular memory use** **• General-purpose (GP) processor (10's – 100's GFLOPS) + high I/O rate, augmented by co-processor(s)**
Model-Based Reasoning Techniques for Autonomy	• Mission planning, scheduling & resource management • Fault management in uncertain environments	• Contingency planning to mitigate execution failures • Detect, diagnose and recover from faults	• High computational complexity • Graceful degradation • Memory usage (data movement) impacts energy management	**• Soft real time / critical** **• Heuristic search, data base operations, Bayesian inference** **• Extreme intensive & irregular memory use (multi-GB/s)** **• > 1GOPS GP processor arrays with low latency interconnect**
High Rate Instrument Data Processing	High resolution sensors, e.g., SAR, Hyper-spectral	• Downlink images and products rather than raw data • Opportunistic science	• Distributed, dedicated processors at sensors • Less stringent fault tolerance	**• Soft real time** **• DSP/Vector processing with 10-100's GOPS (high data flow)** **• GP array (10-100's GFLOPS) required for feature ID / triage**

Future NASA use cases require dramatic improvement over RAD750

Mapping of Use Cases

to the Computational Driver Categories



Deriving Requirements

Requirements Template

- Space Environment(s)
  - Radiation environment at the time of application; e.g., geosynchronous (GEO), low-Earth orbit (LEO), deep space?
- Spacecraft Power Environment(s) / Constraint(s)
  - Available power for avionics and computing, e.g., small spacecraft or rover with limited power availability (6 Watts processor power, 10-15 Watts computer power), medium sized spacecraft (7-12 Watts processor power, 15-30 Watts computer power), or large spacecraft with large power budget (>12-25 Watts processor power, >30-100 Watts computer power)?
- Criticality/Fault Tolerance
  - Is this application life or mission critical, must it operate through faults, can it tolerate errors if detected?
- Real-Time
  - Does the application have a hard real-time deadline; if so, what is the required timing?
- Type(s) of Processing
  - Algorithm kernel(s). Is it primarily e.g., digital signal processing (DSP), data base query, is it parallelizable, is it amenable to a data flow approach?
- Memory Access Pattern
  - What is the primary data movement pattern, e.g., does it fetch data from memory once and then flow through a processing chain, or does it access data in a continuous random access pattern, or does it access sequential data in a continuous and regular pattern?
- Duty Cycle
  - What is the pattern of execution, e.g., is the application called continuously over a long period of time, is it called once and operate for only a short duration, is the application execution profile spiky and/or unpredictable?
- Data Rate
  - What are the I/O and memory access data rates?

Eigen-Apps Summary

~60 application variants/derivatives reduced to 10 representative sets of requirements

App to Eigen-App Mapping	DSP	GP	P	Mission Critical	LP
Throughput = 1-10 GOPS					
Autonomous Mission Planning		X	X	X	X
Disaster Response	X	X			X
Hyspiri	X	X	X		
Throughput = 10-50 GOPS					
Fast Traverse	X	X	X	X	X
Extreme Terrain Landing	X	X	X	X	X
Adept		X	X		
Optimum Observation	X	X	X		X
Space Weather	X		X		X
Robotic Servicing	X	X	X	X	
Cloud Service	X	X	X		
Advanced ISHM		X	X		X
Autonomous and Telerobotic Construction		X	X	X	X
Througput = 50-100s GOPS					
Hyperspectral Imaging	X	X	X		X
RADAR Science	X	X	X		
RADAR EDL	X		X	X	X
Automated GN&C	X	X	X	X	
Human Movement Assist	X	X	X		X
Crew Knowledge Augmentation		X	X		
Improved Displays and Controls		X	X	X	X
Augmented Reality		X	X		X
Telepresence		X	X		X

- Requirements that represent groups of key cross cutting applications
- Derived by selecting low power applications from full applications set and grouping by throughput, processing type, mission criticality

Eigen-App	Throughput	DSP	GP	P	LP	MC
1	1-10 GOPS	X	X	X	X	
2	1-10 GOPS		X	X	X	X
3	10-50 GOPS	X	X	X	X	X
4	10-50 GOPS	X	X	X	X	
5	10-50 GOPS		X	X	X	X
6	10-50 GOPS		X	X	X	
7	50-100 GOPS	X	X	X	X	X
8	50-100 GOPS	X	X	X	X	
9	50-100 GOPS		X	X	X	X
10	50-100 GOPS		X	X	X	

KEY

- DSP – Digital Signal Processing
- GP – General Purpose Processing
- P – Parallelizable
- Mission Critical – Requires Additional Fault Tolerance
- LP – Max Power Available for Processor Chip <6W

Computing Architectures

Candidates evaluated under the HPSC task

- **General-purpose multi-core**
  - Rad-hardened
  - COTS
- **DSP multi-core**
  - Rad-hardened
  - COTS
- **Reconfigurable computing (e.g., FPGAs)**
  - Rad-hardened
  - COTS
- **Graphics processing units (GPUs)**
  - Rad-hardened
  - COTS
- Also: **Hybrid architectures utilizing co-processors**
  - General Purpose Multicore + Reconfigurable Computing
  - General Purpose Multicore + GPU
  - General Purpose Multicore + DSP Multicore

Key Performance Parameters

Application-referenced KPPs

- Computational performance
- Radiation and fault tolerance
- Power and energy management
- Software verfication and validation

Architecture-referenced KPPs

- Software verification and validation (this is the single cross-over KPP)
- Programmability and flight software applicability
- Interoperability
- Extensibility and evolveability

Additional KPPs

- Non-recurring cost
- Recurring cost
- Cross-cutting applicability across the NASA mission set

Other Constraints

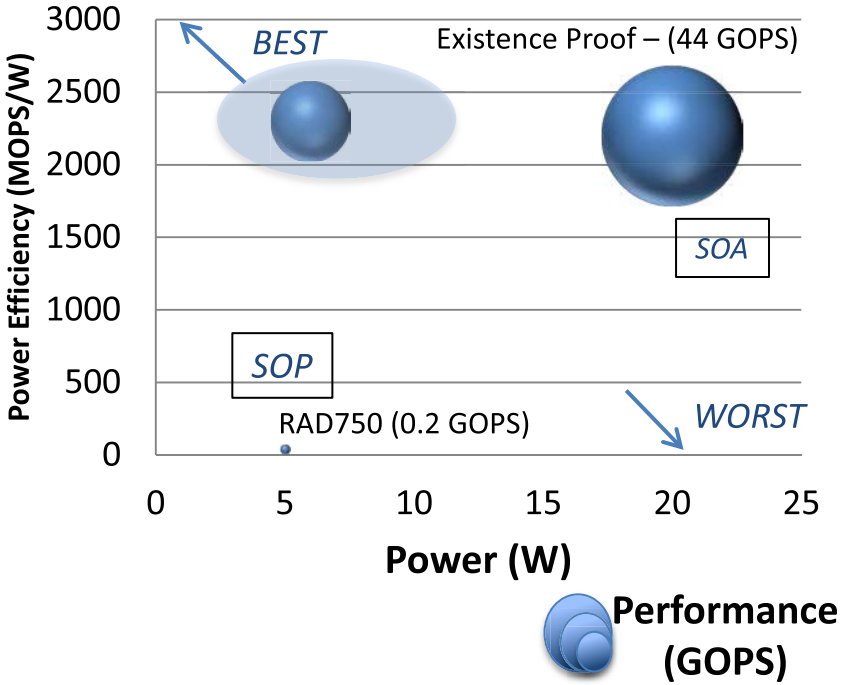
- TRL 5-6 in <=3 years within available resources

Study Recommendation

Rad-hard General Purpose Multi-core

Rad-hard General Purpose Multicore

- **Best overall fit to application requirements** – provides both general purpose and some DSP capability as well as interoperability with co-processors (DSP, reconfigurable)
- **Conducive to Power Scaling at core-level**
  - **Power Dissipation** issues to address fit within available investment resource envelope
- **Conducive to thread-based Fault Tolerance**
  - Fault detection/correction/isolation
  - Ability to segregate failed cores from the pool of available cores in support of graceful degradation
- **Scalable to 10x increase in the number of cores**
  - Combined with power scaling, allows the increased cores to consume power only as thread-load dictates



Computational Performance, Efficiency, and Scalability of Multi-core redefines general-purpose computing for space systems

HPSC Formulation Task Recommendations

Investment Focus and Approach

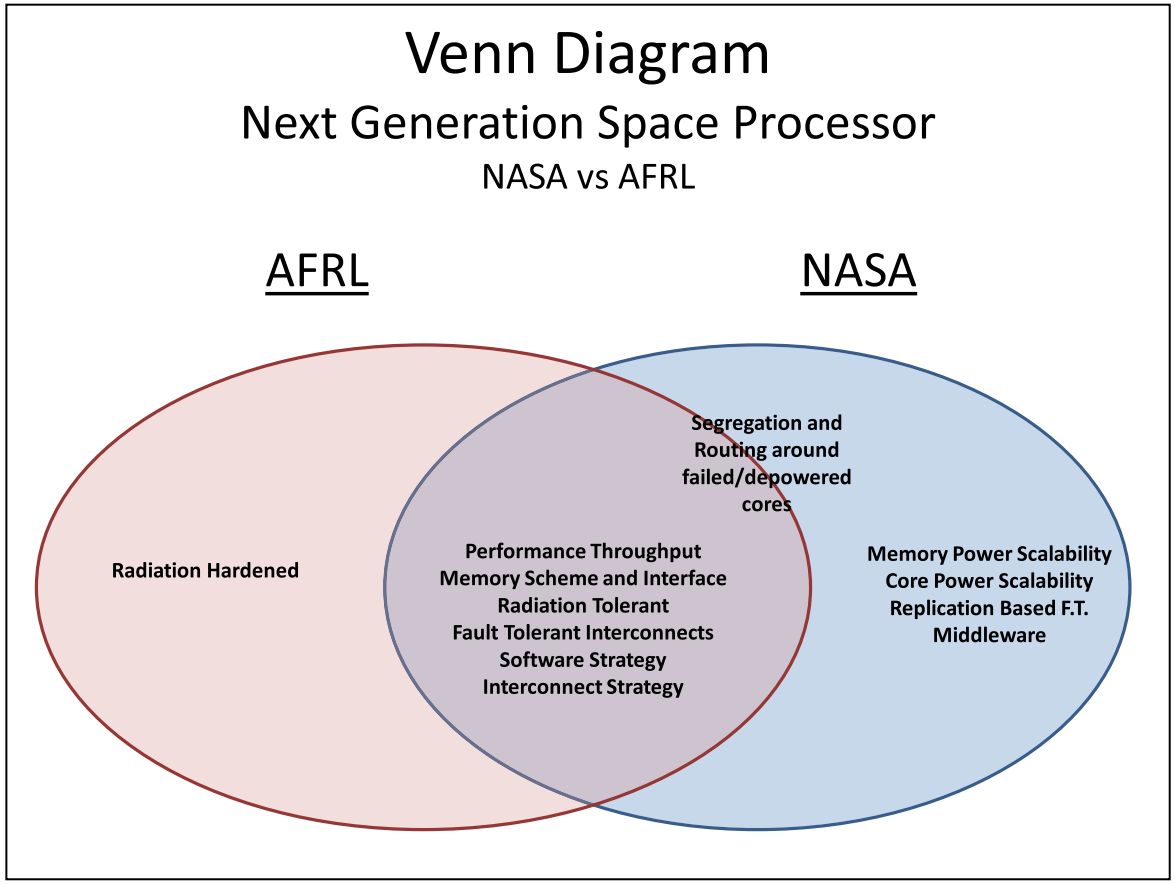
- Focus on **Rad-hard General-purpose Multi-core**
  - Leverage government and industry investments
- Issue a BAA for hardware architecture designs in FY13
  - Solicit flight computing system concepts
  - Prepare NASA requirements and benchmarks for early evaluation of architectures
  - A competitive initial phase, seeking innovative solutions and early risk retirement
- Product of the investment
  - Multi-core hardware chip with bundled real-time operating system (RTOS) and FSW development environment, integrated on an evaluation board
- Include a directed software investment
  - Middleware elements for allocating/managing cores for varying operational objectives, working closely with the FSW community, driven by knowledge of the NASA applications

Challenge the community to develop an innovative, extremely high performance, low power, flexible, rad-hard GP multi-core processor within available budget and schedule

Enter AFRL!

Future Flight Computing

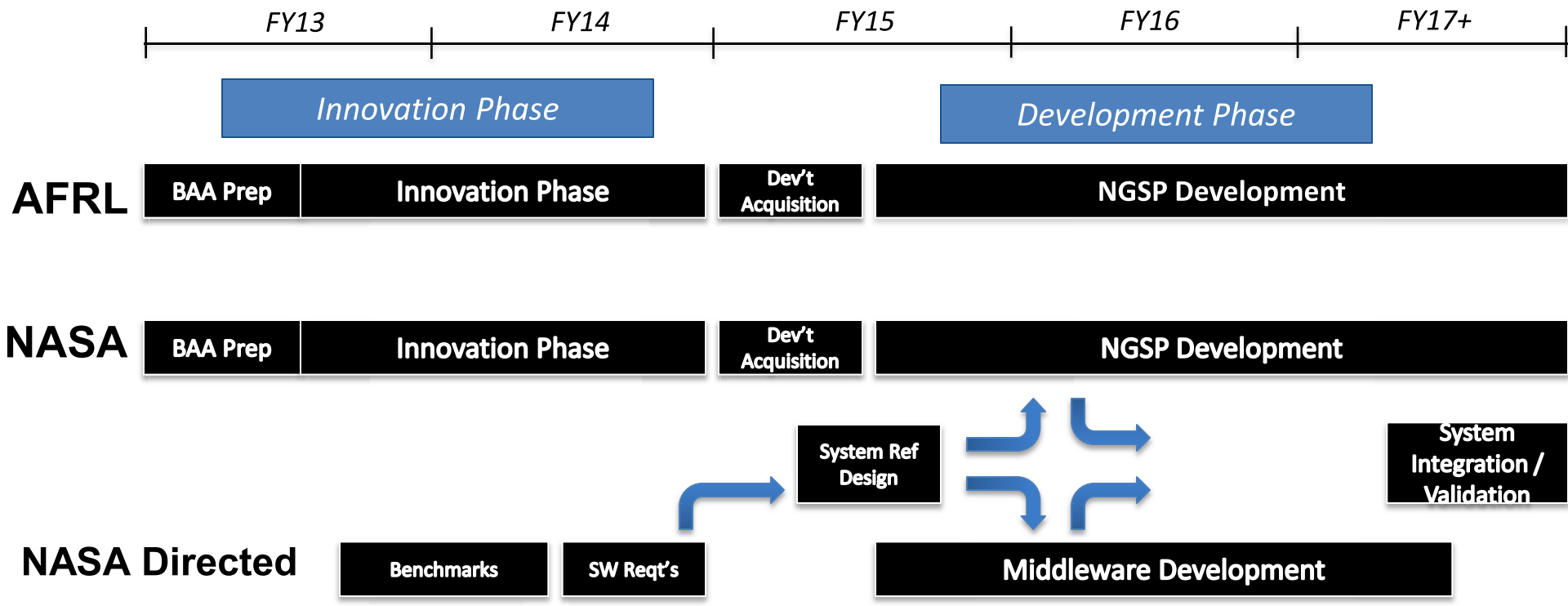
AFRL vs. NASA Requirements



Sufficient common ground to justify a joint AFRL-NASA program

Joint NASA-AFRL Program Concept

Next Generation Space Processor (NGSP)



Initial joint program plan was approved and funding allocated for Innovation Phase
BAA issued and benchmark development initiated

NGSP Analysis Program

Innovation Phase – Top Level Objectives Summary

- A minimum of 24 processor cores and 24GFLOPS to support both highly parallel applications and to provide a high degree of granularity for power management, fault tolerance and program unit distribution.
- A minimum of 10GFLOPS, eight 10Gb/s I/O and four DDR 3 Memory Ports at 7 Watts
- Dynamically power scalable at core level granularity by powering and depowering cores in real time without disrupting system operation, with very low idle power load (<<1W)
- Based on commercially available hardware and software IP (processing cores, external I/O and memory interfaces, software stack and development environment)
- Able to reset individual cores or a cluster of cores, as determined by smallest unit of granularity
- Radiation hard to at least 1 Mrad TID, Latch up Immune to an LET of at least 90, with a hardware-uncorrected SEE rate of not greater than 0.01 event/day in Adams 90% worst case GEO environment
- Interoperable with other high performance computing architectures, e.g., FPGAs

*****Proposers encouraged to offer alternatives *****

AFRL BAA-RVKV-2013-02
Proposal Due Date 5/29/13

NGSP Analysis Program

Innovation Phase – Top Level Summary of Desired Deliverables

- Identification of process and/or RHBD library to be used, along with test data to substantiate claims of radiation hardness
- Simulation/model results for a set of NASA-defined application benchmarks
- Simulation/model results of fault response and power management
  - Demonstrated ability to operate through faults and restore correct operation
  - Demonstrated ability to dynamically manage power under software control
  - Measure latencies and overhead, validate boundaries and mechanisms
- Management/Development plan that makes a credible case that TRL 5-6 can be achieved within available cost and schedule constraints
  - List of IP to be used, and agreements in place to acquire if selected
  - Detailed work breakdown structure provided to at least 3 levels
  - Complete device development schedule (including margin)
  - Detailed development cost estimate (including reserves)
  - Risks identified and risk management approach provided

AFRL BAA-RVKV-2013-02
Proposal Due Date 5/29/13

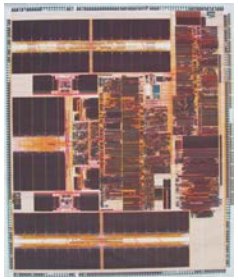
Use of Benchmarks

- Provide benchmarks that capture realistic performance estimates and evaluate machine performance under off-nominal conditions
  - Include both application-based (e.g., algorithm kernels) and architecture-based (e.g., energy management, fault tolerance) benchmarks
  - Provide ability to stress to failure
  - Based on real space applications, but generalize to encompass classes of applications
  - Capture at a general level – avoid ITAR and proprietary/competition-sensitive issues
  - Provide in a form that can be used in conjunction with simulations, models, machines
  - Allow users to implement for best performance on target architecture

- Approach is to develop a library of *component specifications* – including algorithms, non-algorithmic sections of code, data structures, data access patterns and flows, stressor "knobs", required measurements
  - Starting with complex high performance applications, decompose into modular benchmark components
  - Generalize these components to be non-application specific (to the extent possible)
  - Concatenate component specs to build application benchmark specification
  - Add application level stressor knobs and measurements, external "observer" requirements, fault tolerance and power estimation configurations
  - Specify results reporting format

Benchmarking Rationale

- Without benchmarking, there is considerable risk that a computer will not live up to its expected performance
  - Prospective users typically assess performance based on processor clock rate, word width, and instruction set
  - However, design features can introduce bottlenecks that can limit performance for real-world applications to a fraction of the expected performance
  - Often these bottlenecks are difficult to identify via design reviews
  - Benchmarks provide the most reliable means to assess real-world performance
- Case in point – RAD750
  - Original processor had serious performance limitations due to memory performance
  - Modifications were necessary in the implementation of the L2 cache and DMA



- The risk may be greater
  - The RAD750 implemented fixes at the board level, whereas HPSC fixes would likely need to be done in silicon
  - For multi-core processors, because application performance can be sensitive to architecture (i.e., topology, interconnect methodology, memory implementation), it is essential that the optimal architecture be established up front for the broad array of space-based applications

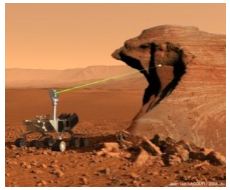
AFRL / NASA BAA

Status and FY13-14 Plans

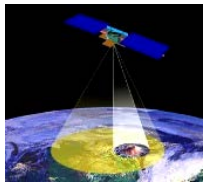
- **April 2013**: AFRL/NASA BAA posted
  - Next-Generation Space Processor Analysis Program BAA-RVKV-2013-02
- **May 2013:** Proposals due
- **September 2013**: Contract awards
  - BAE (Manassas, VA)
  - Boeing (Seattle, WA)
  - Honeywell (Clearwater, FL)
- **November 2013:** Program Kickoff Meeting
- **February 2014:** Technical Interchange Meeting
  - Contractor derived USAF requirements documents delivered
  - Finalized joint AFRL/NASA requirements document published
- **June 2014**: Final benchmarks for NASA applications delivered to contractors
- **November 2014**: Innovation Phase concludes
  - Draft final reports delivered to AFRL/NASA
  - Software requirements available
  - Evaluation and selection of architecture(s) for implementation

Transition Plan for Mission Use

First-user Infusion Pathways



- **Mars Exploration** – Future Mars missions (2020 and beyond)
  - ***Landing Vision System*** – TRN and HDA algorithms for pinpoint landing, for site access and sample return
  - ***Fast Traverse*** – Remove computation as a limiting factor for surface mobility, drive safely 10X+ faster
  - ***Rover Surface Science*** – Perform science operations continuously during traverse: "Walk and chew gum"
  - ***In-Situ Resource Utilization*** – Planning and execution of prospecting and processing operations
  - *Discussions underway with the **NASA Space Technology Program** and the **Mars Exploration Program** towards a Mars 2020 technology payload concept to demonstrate capabilities supported by HPSC*



- **Earth Science** – Future Earth-observing missions will carry high data rate instruments (hyper-spectral, radar …), HyspIRI study

- **Human Spaceflight** – MPCV/Orion: Time/space and memory partitioning will be an important human rating requirement – multi-core can provide natural fault containment

- **US Air Force** – Sensor Payload Processing: Future systems will generate large amounts of data (hyper-spectral, hyper-temporal, radar …)

Summary

Flight Computing for the Future

- Future NASA mission scenarios call out for significantly ***improved flight computing*** capability
  - Several NASA OCT Roadmaps and the NRC report identify ***improved flight computing*** as a foundational technology
- AFRL independently identified the need for ***improved flight computing***
- ***Improved flight computing*** means enhanced computational performance, energy efficiency, and fault tolerance
  - Like power and propulsion, flight computing is a ***core flight capability***; a technology advance will be a capability ***multiplier*** impacting the return from all future missions

It is time to move beyond the 1990's technology of the RAD750 and *Redefine the role of computing in space systems*