



Fault Mitigation Schemes for Future Spaceflight Multicore Processors

Rafi Some, Kim P. Gostelow, John Lai, Leonard Reder, Jim Alexander, Brad Clement

Jet Propulsion Laboratory, California Institute of Technology

June 20, 2012



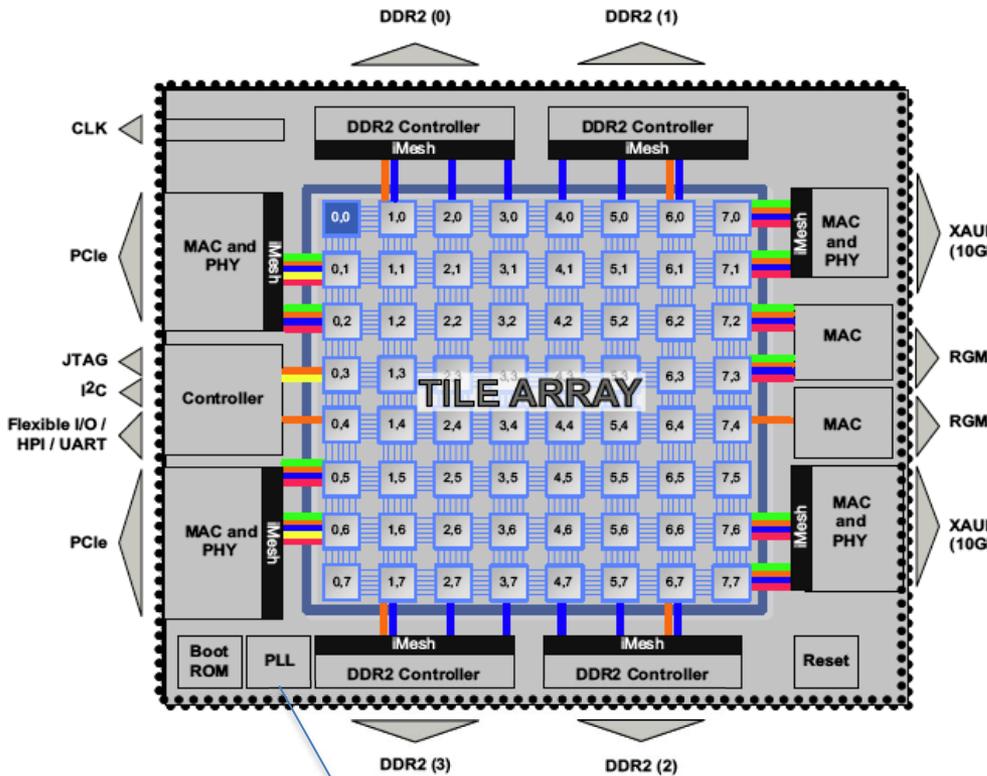
Goal



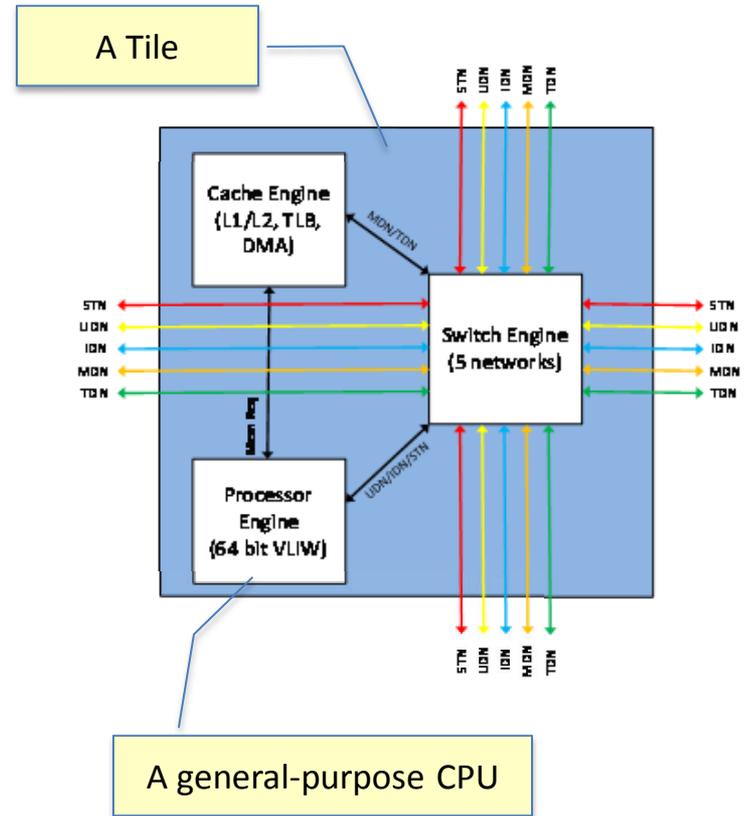
- Achieve fail-operational and graceful-degradation behavior in realistic flight mission scenarios, such as Mars Entry-Descent-Landing (EDL) and Primitive Body proximity operations.
- Follow a policy-based approach: reliability achieved depends upon the resources committed.



What is Multicore?



Tiler TILE64
Multicore Chip





Where is Multicore?



Tile64 is available – but not space qualified.

Maestro processor – path-to-flight

- RadHard by Design
- Floating-point
- 49 cores at 44 GOPS (RAD750 is 0.2 GOPS)

Opera Software

- VxWorks (mostly)
- OpenMP, MPI, other libraries

Development Tools

- Tile-Eclipse + other tools

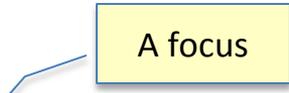


Observations



Multi-core processors can offer many advantages over a single-core architecture

- Parallel processing
- Power throttling
 - This is essential to all that follows
 - Real-time software fault tolerance and recovery



A focus

But they also bring their own issues to be solved

- Unprotected data packets when routing through the network and cores
- A failed core impedes data routing to the destination core along its coordinate



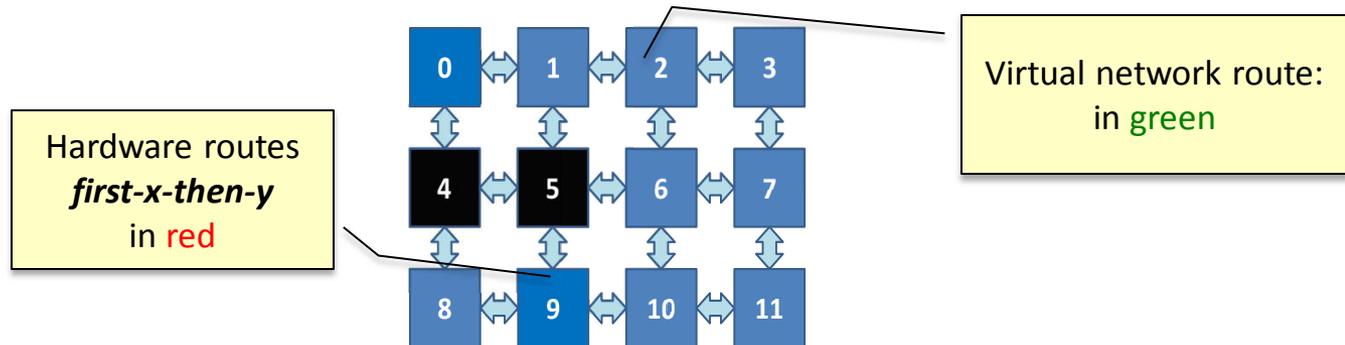
Some Network Fault Mitigations



Some single-bit errors can be corrected through Hamming Codes on address and data:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved	Dest_X												Dest_Y												Length				Word 0: Route Header			
Tag																																Word 1: Tag
Data ECC																				Length ECC												Word 2: ECCs
Dest_index																Length																Word 3: Route Data
Tag																																Word 4: Channel
Packet data																																Packet Data (1-124 words)

Node failures can be accommodated by using a virtual network:





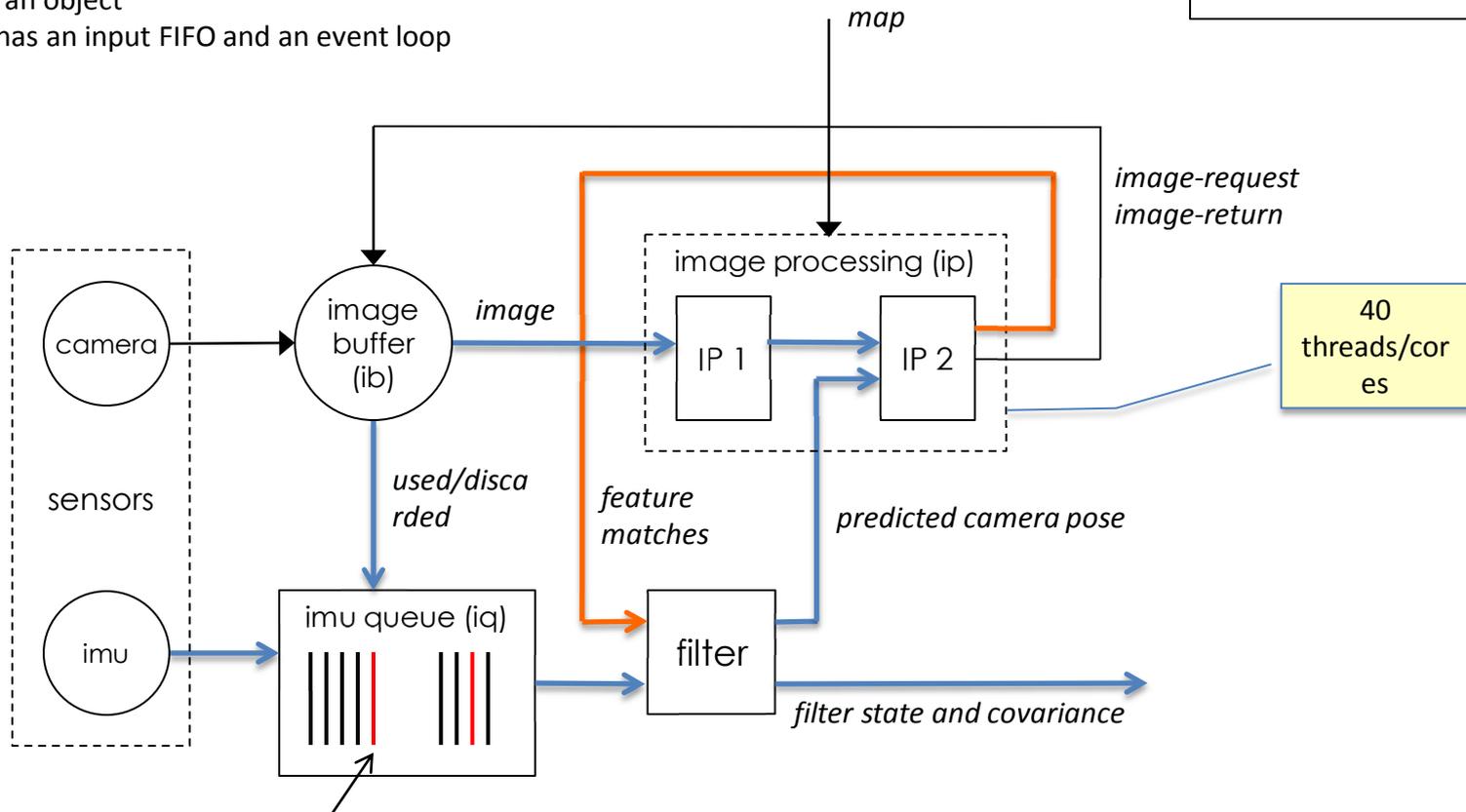
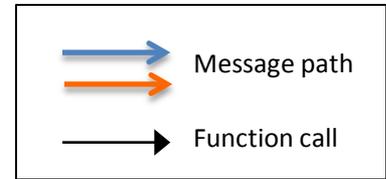
The Demonstration Application



TRN: Terrain-Relative Navigation for EDL

Notes:

- A box is one or more threads
- A circle is an object
- A thread has an input FIFO and an event loop



red: there is an image associated with this marked imu measurement

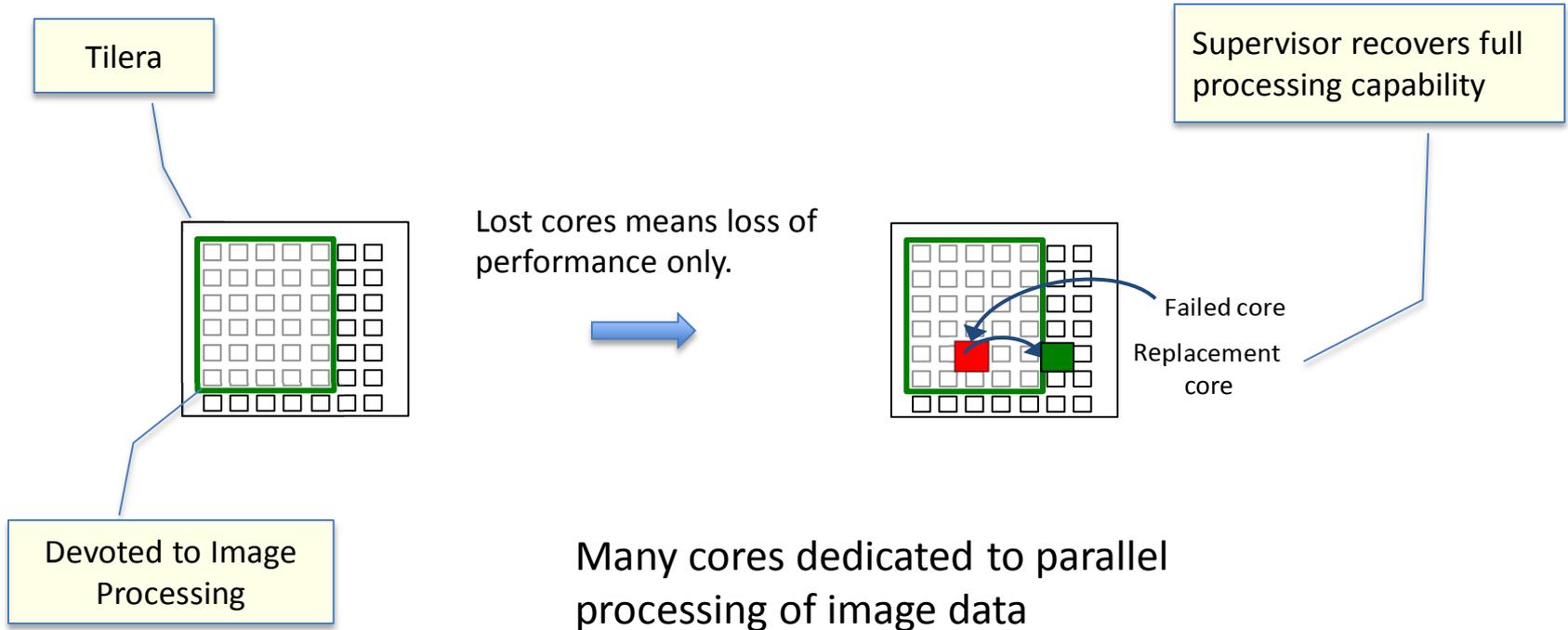


Graceful Degradation Fault Tolerance



Graceful Degradation

The nature of the vision processing in this application allows easy implementation



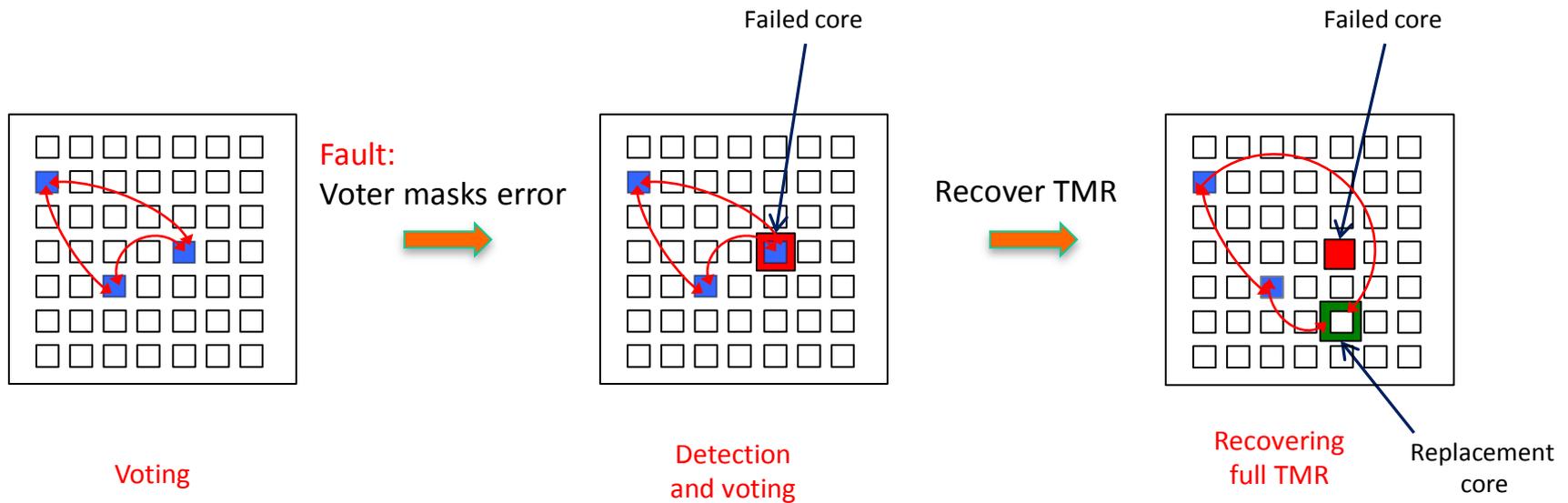


Fail-operational Fault Tolerance



Fail Operational

Applied to the filter





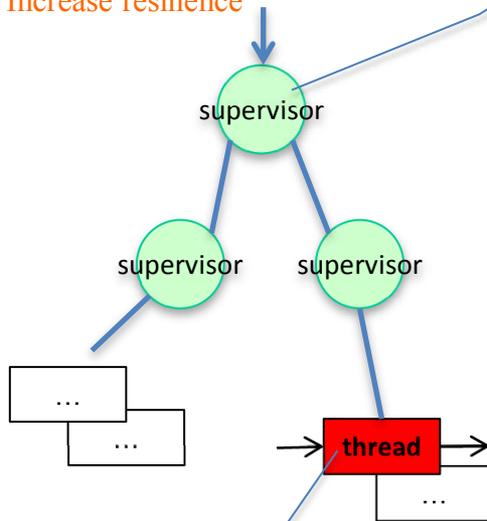
Policy-based Computing



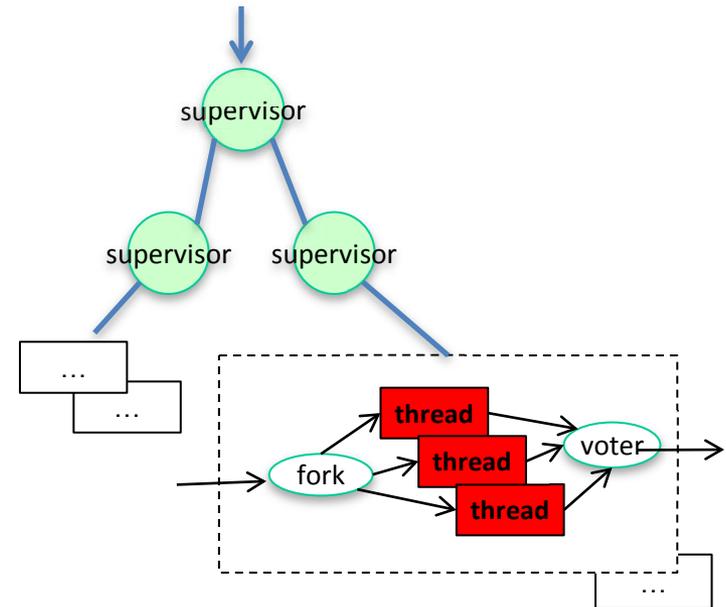
Supervisor

1. Initially creates the application
2. Monitors health and performs fault recovery
3. Carries out policies: power/cores/reliability

Policy change:
Increase resilience



Rewire filter for TMR.



Thread must be "repeatable", e.g., no shared memory – a function.



- Implemented
 - Hardware fault mitigations
 - TMR with voter
- Real-time, policy-based actions designed, now being implemented



References



- Joe Armstrong “Making reliable distributed systems in the presence of software errors” PhD Thesis, *Swedish Institute of Computer Science*, 2003.
- Kim P. Gostelow *Policy-based Computing and Extra-Functional Properties of Programs*. Presentation at the Software Working Group of the Fourth Workshop on Fault-Tolerant Spaceborne Computing Employing New Technologies 2011. Albuquerque, NM. May 22, 2011
- James Alexander, Yang Cheng, William Zheng, Nikolas Trawny, and Andrew Johnson, “A Terrain Relative Navigation Sensor Enabled by Multi-Core Processing”, Proceedings, 2012 IEEE Aerospace Conference, Big Sky, MT.
- Carlos Y. Villalpando, David Rennels, Raphael Some, and Manuel Cabanas-Holmen, "Reliable Multicore Processors for NASA Space Missions", Proceedings, 2011 IEEE Aerospace Conference, Big Sky, MT.