KASSPER
Real-Time Embedded Signal Processor Testbed

Ed Baranoski
Glenn Schrader

Massachusetts Institute of Technology
Lincoln Laboratory

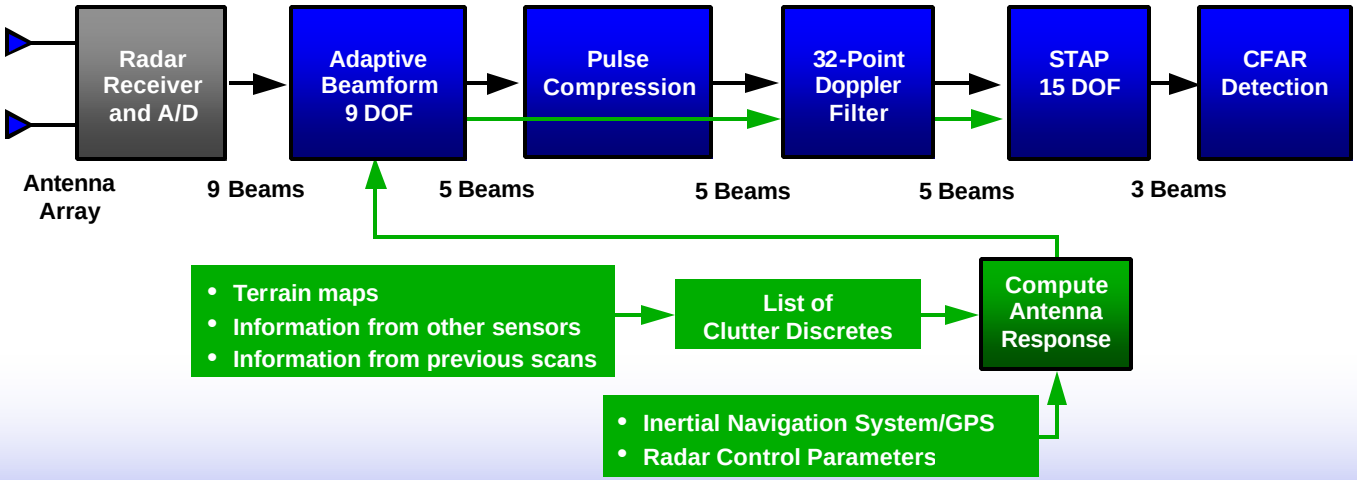
3 April 2002

50 years
MIT LINCOLN LAB

Outline

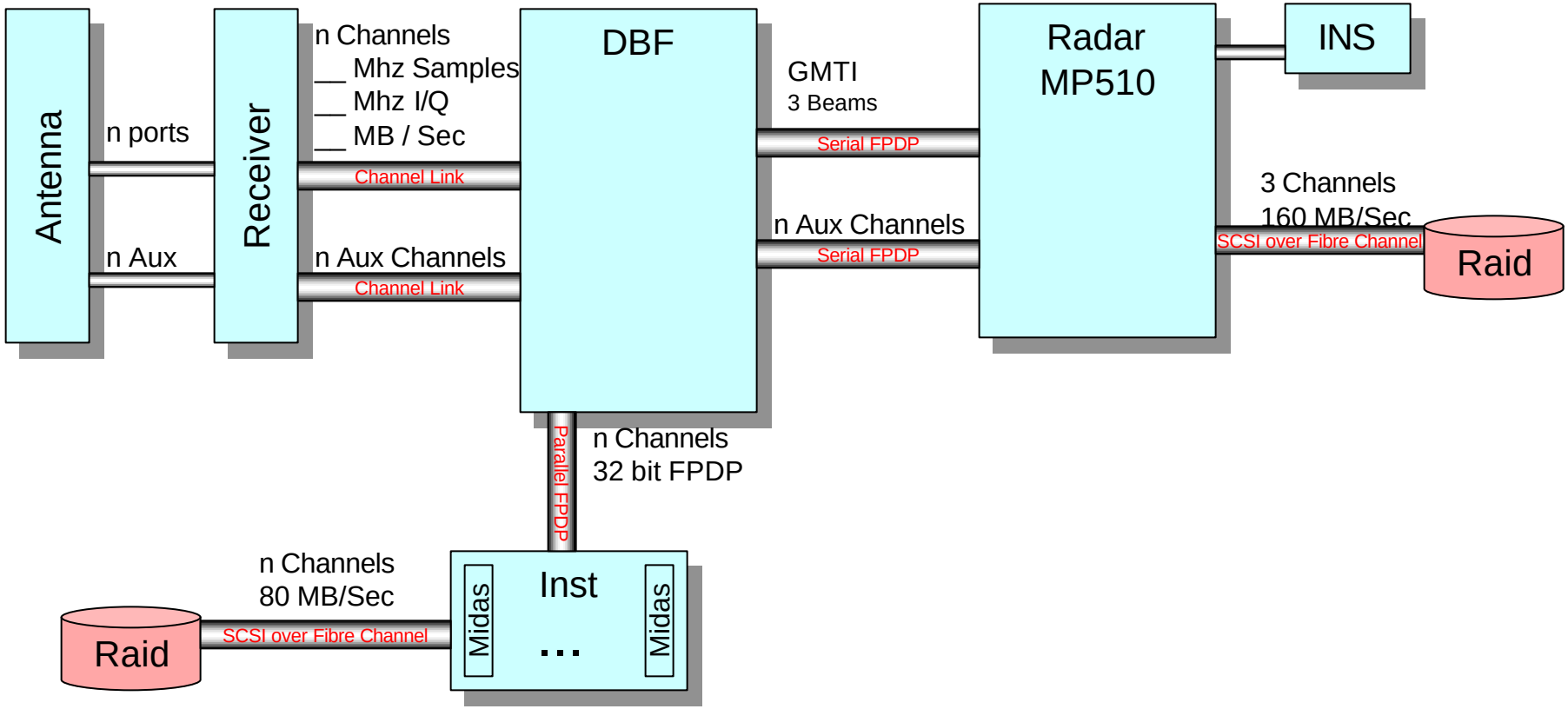
- Introduction
- Processor sizing
- Parallel Vector Library
- Multi-phase processor acquisition strategy
- Algorithm/implementation coupling
- Summary

KASSPER: Knowledge-Aided Sensor Signal Processing Expert Reasoning



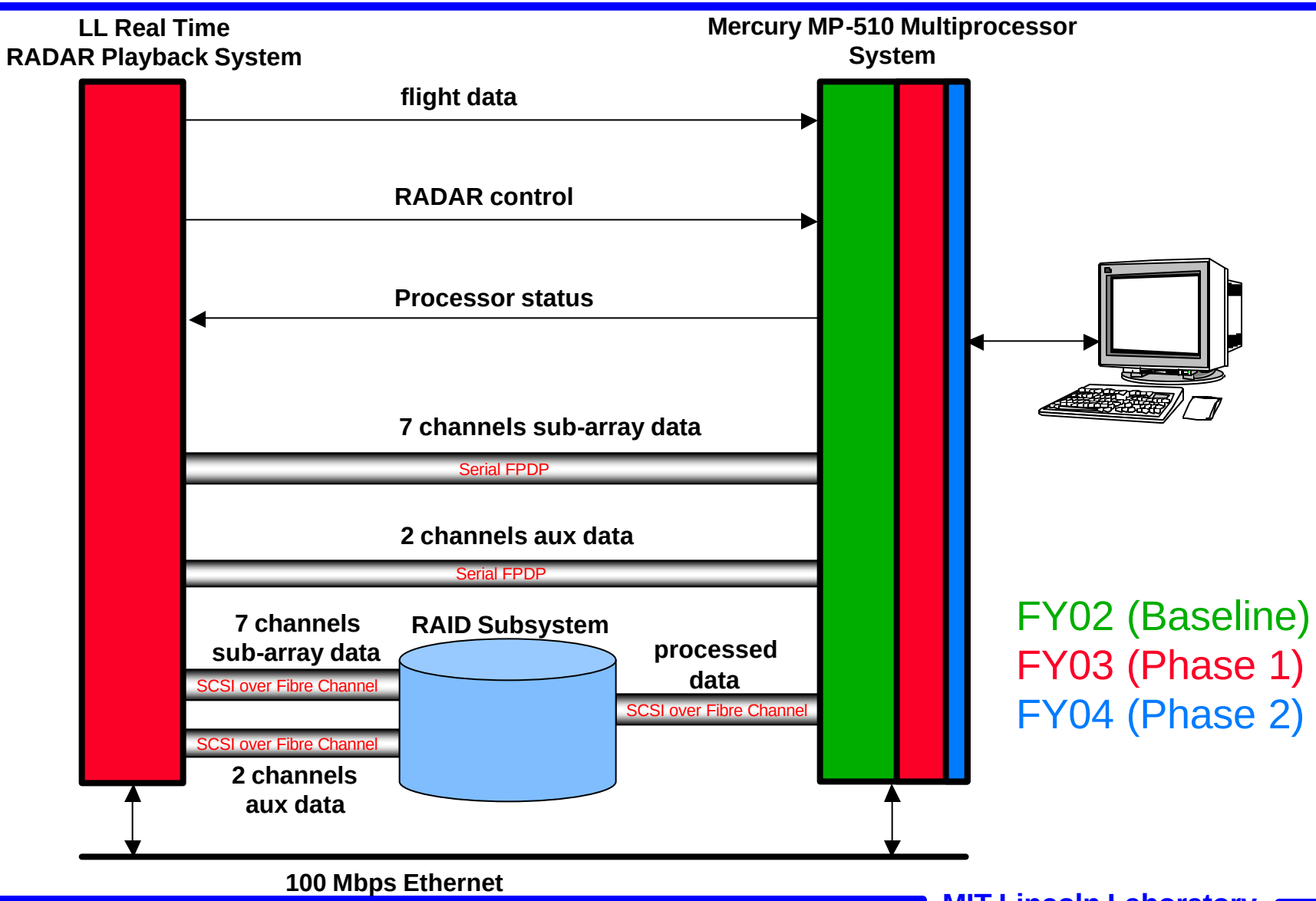
- MIT Lincoln Lab to develop advanced algorithms and real-time embedded processor
  - Integrate into Northrop Grumman X-band radar (4'x1' aperture) on a BAC1-11
  - Transition path to MP-RTIP or MC2A anticipated (potentially applicable to SBR)
- Provide capability to implement "best of breed" algorithms in real-time

BAC1-11 GMTI Airborne System



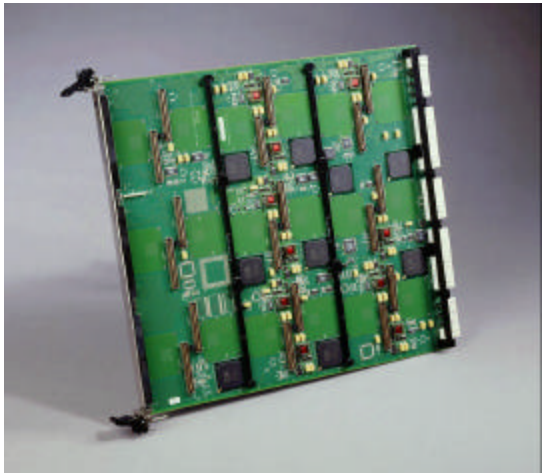
- **Receiver I/Q data, INS data, RADAR mode, waveform, etc. is recorded to the RAID storage**
- **The LL KASSPER system will use this recorded data as input**

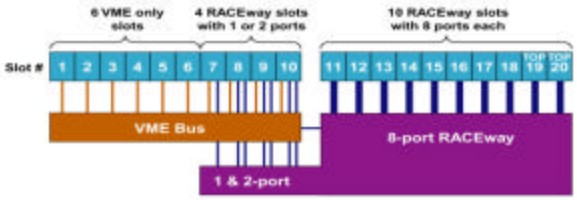
KASSPER real-time Embedded Processor Architecture

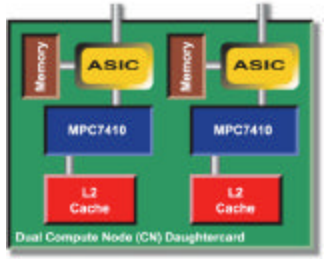


FY02 (Baseline)
FY03 (Phase 1)
FY04 (Phase 2)

Mercury PowerStream System







- 500 Mhz PowerPC G4 /w Altivec (4 Gflop/sec Peak)
- 10 Boards * 18 CPUs/board = 180 CPUs max (720Gflop/sec Peak)
- Bisection Bandwidth up to 8.5 Gbyte/sec

Outline

- Introduction
- Processor sizing
- Parallel Vector Library
- Multi-phase processor acquisition strategy
- Algorithm/implementation coupling
- Summary

KASSPER Signal Processor Estimates

Bandwidth estimates:

PROCESSING STAGE	INPUT DATA SIZE (ch X dop X rng)	Input BW (MB/sec)[1]	Output BW (MB/sec)	Total I/O BW (MB/sec)	Cache Overhead (Gflop/sec)[2]	MIN #CPUS Needed for I/O
Data Input	9x33x16K (cplx short)	692	692	1384	N/A	6
Beamforming	9x33x16K (cplx short)	692	768	1460	9.2	6
Pulse Compression	5x33x16K (cplx float)	768	768	1536	9.6	6
Doppler Filtering	5x33x16K (cplx float)	786	1489	2257	14.1	9
STAP	5x64x16K (cplx float)	1489	447	1936	12.1	8
CFAR	3x33x16K (cplx float)	447	1	448	2.8	2
Data Output	Detect list	1	1	2	0.1	<1

- **Careful processor sizing for eventual Mercury implementation**
- **I/O bandwidth and computational loading are both factors**

Processing estimates:

PROCESSING STAGE	Gflop/sec Estimate[3]	Estimated Efficiency[4]	Process Gflop/sec	Cache Overhead Gflop/sec	Total Gflop/sec	#CPUS Needed[5]
Data Input[6]	N/A	N/A	N/A	N/A	N/A	9
Beamforming	5.2	10%	52.0	9.2	61.2	16
Pulse Compress	16.0	15%	106.6	9.6	116.2	27
Doppler Filter	3.6	10%	36.0	14.1	50.1	13
STAP	8.8	10%	88.0	12.1	100.1	26
CFAR	0.1	1%	10.0	2.8	12.8	4
Data Output	N/A	N/A	N/A	0.1	0.1	1
Totals					340.5	96

1. Bandwidth assumes PRF = 1200 Hz, cmplx short = 4 bytes, cmplx float = 8 bytes, Individual CPU I/O BW = 266 MB/sec peak
2. Overhead estimate is 25 Flop/float/direction, based on Mercury's SAL timings
3. Based on complexity analysis and the expected data rates
4. Based on existing LL and Mercury benchmarks
5. 500MHz PPC CPUs provide 4 Gflop/sec peak
6. One I/O control CPU is needed per input channel

KASSPER Signal Processor Processing Stage Task Mapping

PROCESSING STAGE	MIN #CPUs Needed for Processing	MIN #CPUs Needed due to I/O
Data Input	9	6
Beamforming	16	6
Pulse Compress	27	7
Doppler Filter	13	10
STAP	26	8
CFAR	4	2
Data Output	1	<1

Software Tasks
Data Input Task
Beamforming Task
Filtering Task
STAP Detection Task
Data Output Task
Scheduler Task

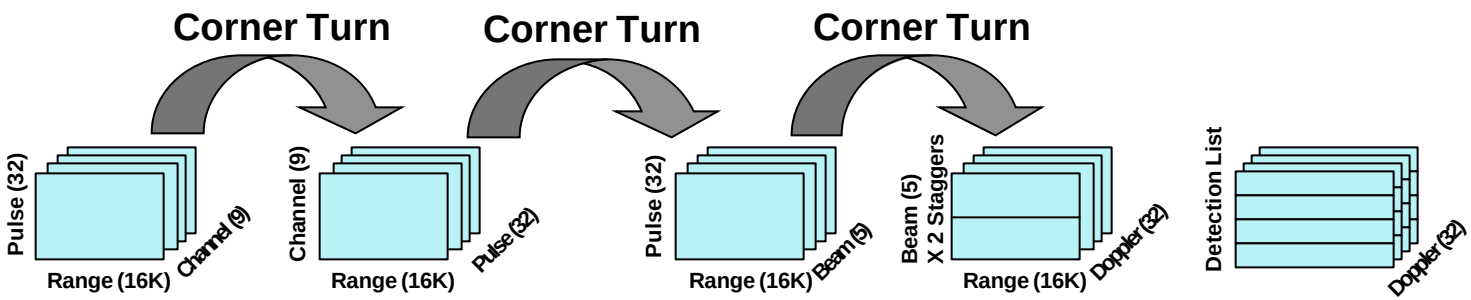
- Doppler filter stage is nearly I/O bound
- Combine pulse compression and Doppler filtering
- Combine STAP and CFAR

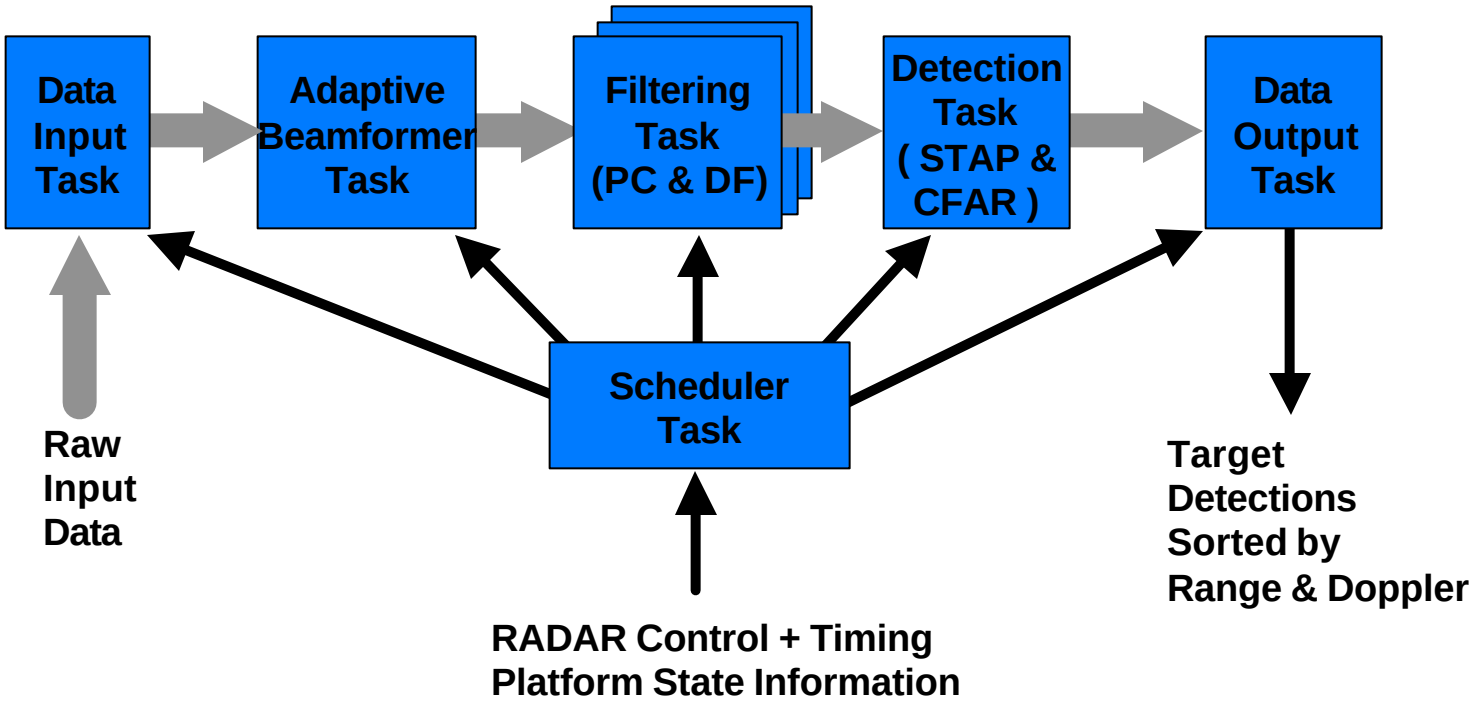
Software Tasks	CPUs needed for Processing	CPUs needed for I/O	Minimum # of CPUs needed	Data Distributed By	Actual # CPUs Needed[2]
Data Input Task	9	6	9	Channel (9)	9
Beamforming Task	16	6	16	Pulse (32)	17[4]
Filtering Task[3]	40	10	40	Beam (5)	40 (5x8)
STAP/Detection[3]	30	6	30	Doppler (32)	32
Data Output Task	1	<1	1	N/A	1
Scheduler				N/A	1
Housekeeping[1]				N/A	3

Total CPUs = 102
Total CPUs /w 50% reserve = 153
MAX CPUs = 174

1. Housekeeping is a catchall for system support processors (nameserver, etc.)
2. # of CPUs is adjusted so that the data spreads evenly across the CPUs
3. Total # of CPUs needed for I/O decreases when tasks are combined
4. Extra CPU dedicated to beamforming weight computation

KASSPER Signal Processor Software Block Diagram

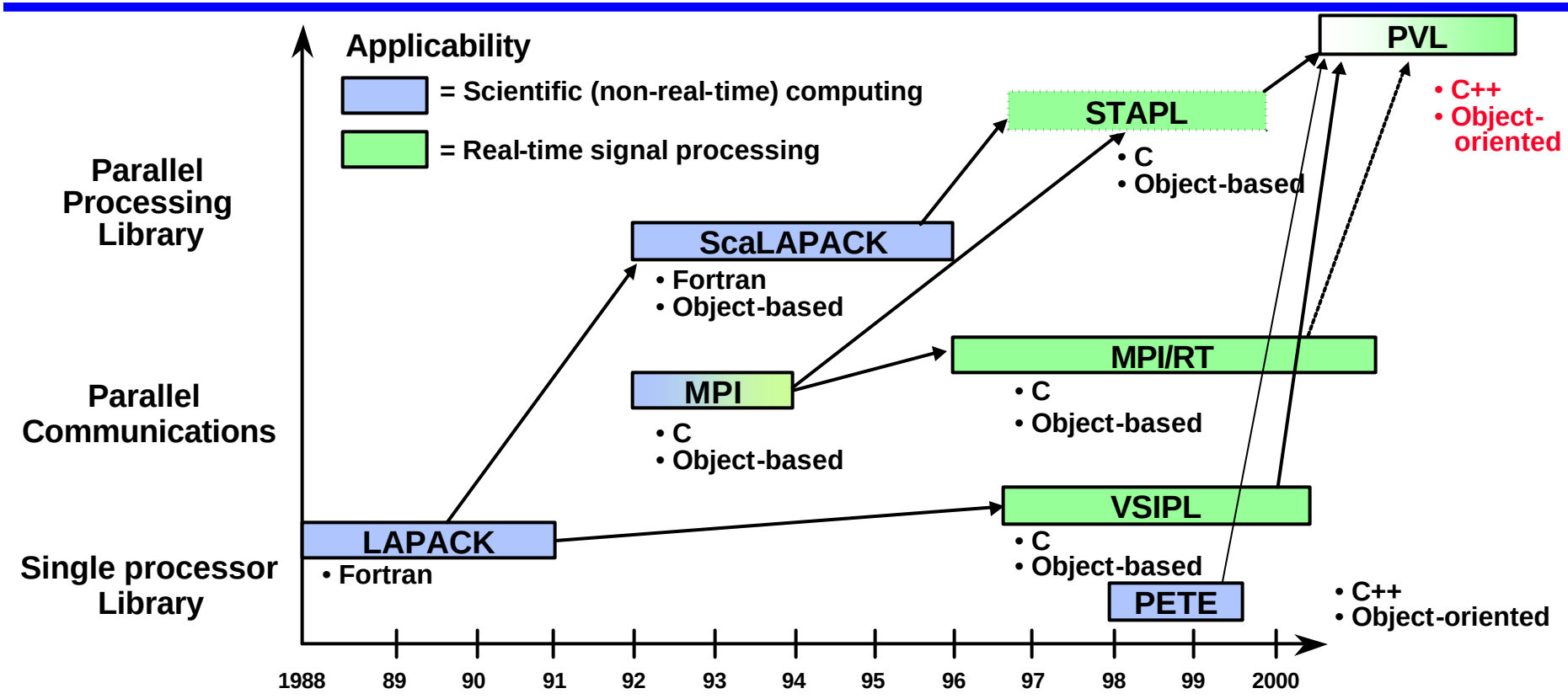




Outline

- Introduction
- Processor sizing
- Parallel Vector Library
- Multi-phase processor acquisition strategy
- Algorithm/implementation coupling
- Summary

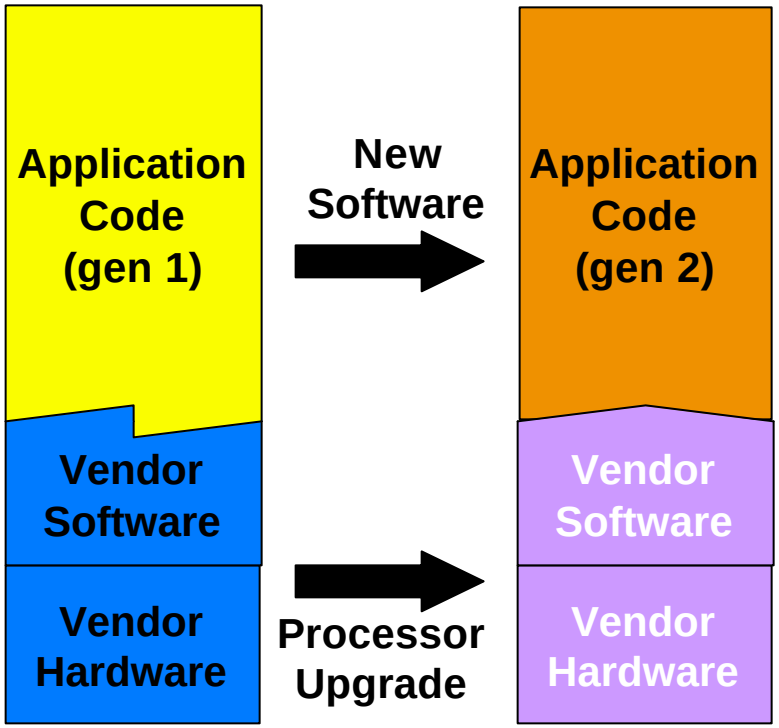
Evolution of Parallel Libraries



LAPACK = Linear algebra package
MPI = Message-passing interface
MPI/RT = MPI real-time
ScaLAPACK = Scalable LAPACK
VSIPL = Vector, Signal, and Image Processing Library
STAPL = Space-Time Adaptive Proc. Library
PETE= Portable Expression Template Engine
PVL = Parallel Vector Library

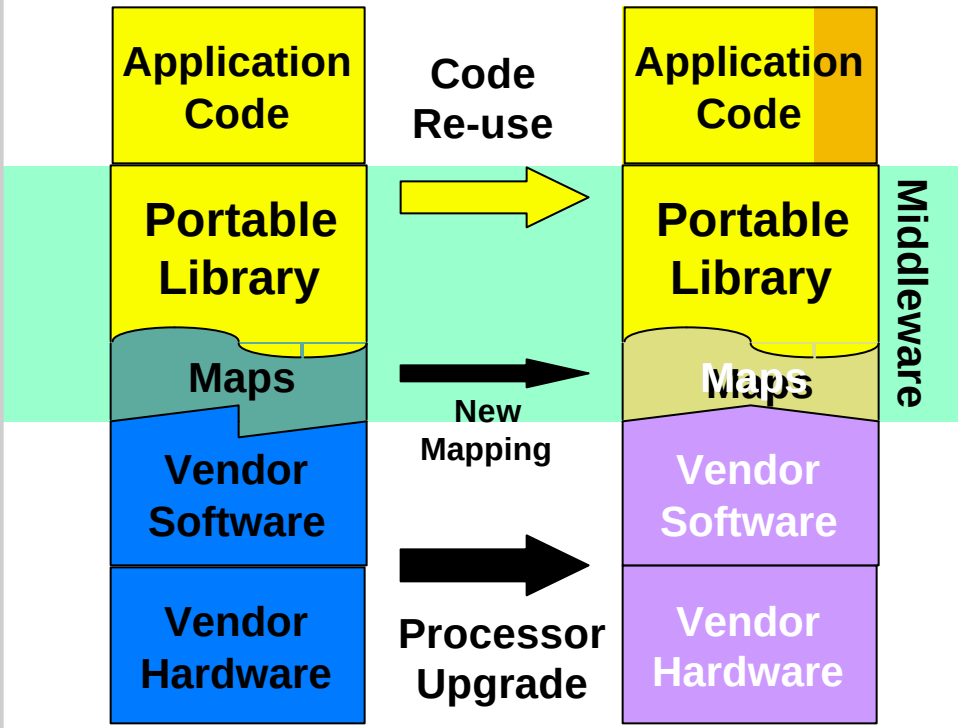
Middleware Concept

Old Upgrade Approach



- New hardware required new application software
  - "Point-design" in hardware replaced with "point-design" in software
- Large development cost

Upgrade Approach using Portable Standard



- Application code written with portable library
  - New software restricted to remapping to physical hardware, new features
- Much lower development cost

PVL Philosophy

Separate the job of writing a parallel application from the job of assigning hardware to that application

"Application Developer"

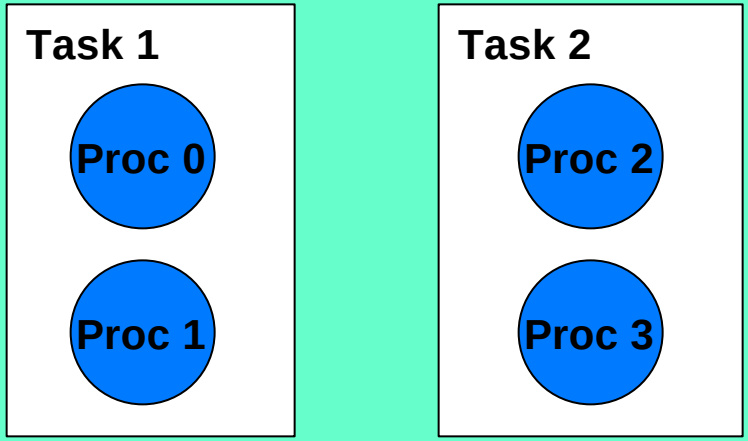
- Converts algorithm into code

```
while( !done )
{
    task1();
    task2();
}
```

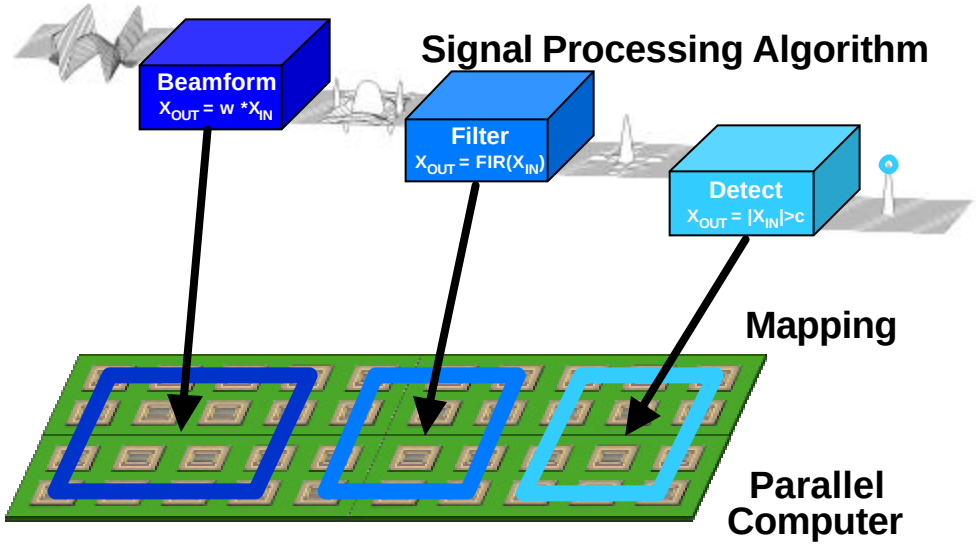
- Writes code once
- Easier to code, because only concerned with mathematics, not distribution

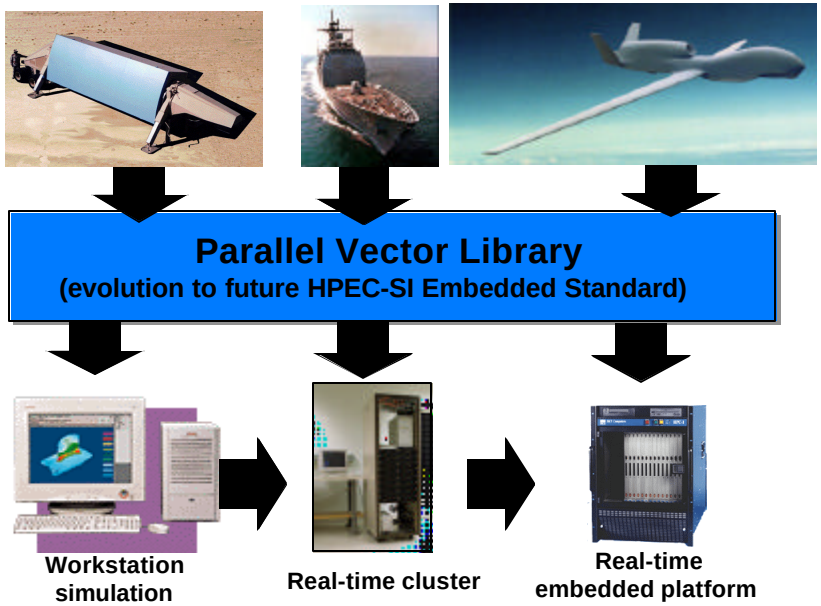
"Mapper"

- Maps code to hardware
- Creates new mappings when code is scaled or ported



Mapping Task & Data Parallel Algorithms onto Parallel Processors





- PVL provides basic linear algebraic and signal processing cord
  - Algorithm coding can proceed independent of machine mapping
  - Mapping will optimize to any parallel platform
- Current mappings to clusters, Mercury, Sky, Cray, etc.
  - Measured efficiencies for earlier applications:
    Filtering: ~50%, Beamforming: ~25%, Post-proc. (par. Est., etc.): ~10%
  - PowerPC AltiVec optimization required to get full benefit

Outline

- Introduction
- Processor sizing
- Parallel Vector Library
- **Multi-phase processor acquisition strategy**
- Algorithm/implementation coupling
- Summary

Processor Acquisition Strategy

- Perform initial development on a Linux based Beowulf cluster
- Purchase a single 9U motherboard's worth of PPC processors for delivery in late 3Q or early 4Q FY02
- Purchase the MP-510 and remaining FY02 PPC processors for delivery approx. 1Q FY03 (delivery constrained)
- Purchase the remaining PPC processors for in FY03 and FY04

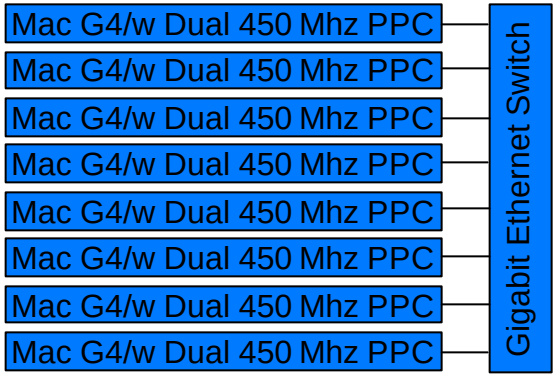
G4 cluster (now)

9U Mercury G4 card (4Q FY02)

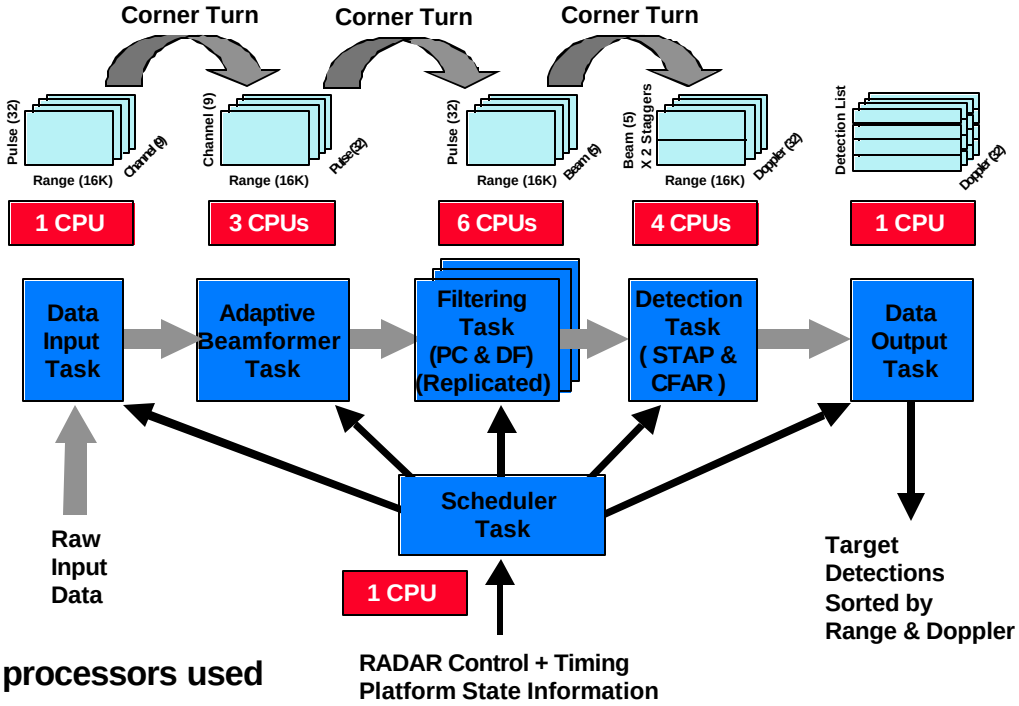
9U Mercury chassis
- Partially populated (FY03)
- Fully populated (FY04)

The scalability and cross-platform portability of PVL is what makes this possible

Initial Linux G4 Processor

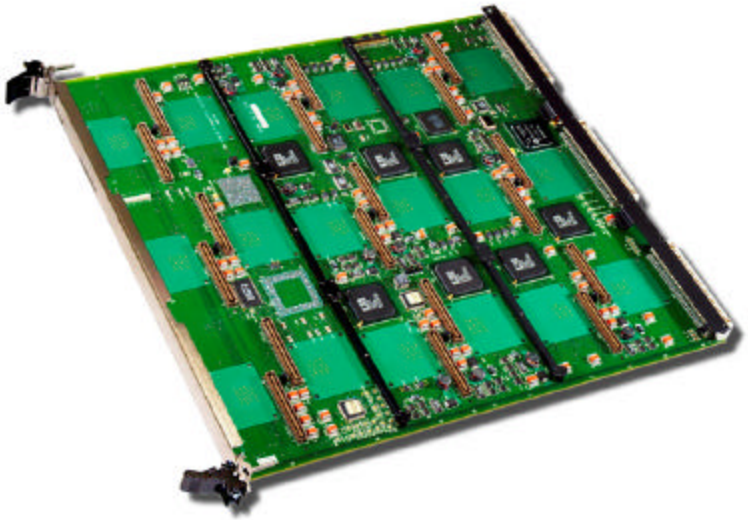


- **Eight Dual G4 Power Mac CPUs (16 CPUs)**
- **28.8 Gflop/sec Peak Processing**
- **Same PPC processor type as the Mercury but uses TCP/IP over Gigabit Ethernet for inter-processor communication**



All 16 processors used

Interim 9U VME Mercury Processor

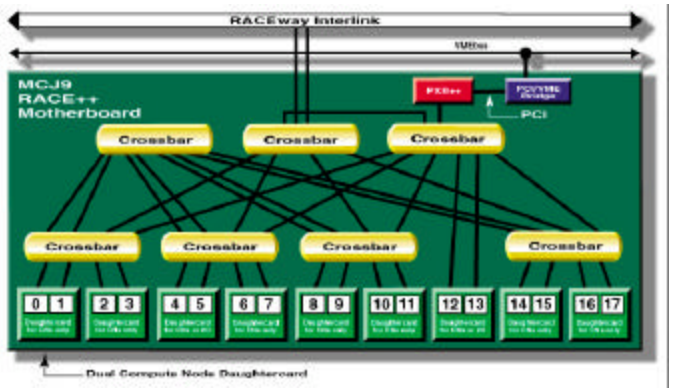


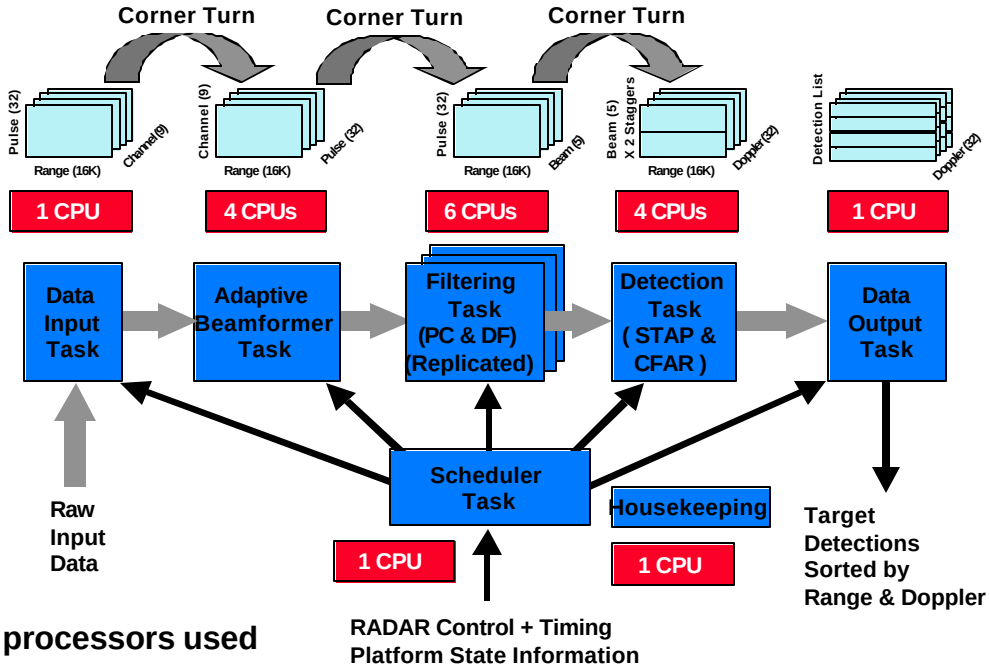
- Single 9U Race++ Motherboard with 18 500Mhz PPCs
- 36 Gflop/sec Peak Processing

Force 50 SBC /w Solaris

Motherboard/w 18 500Mhz PPC

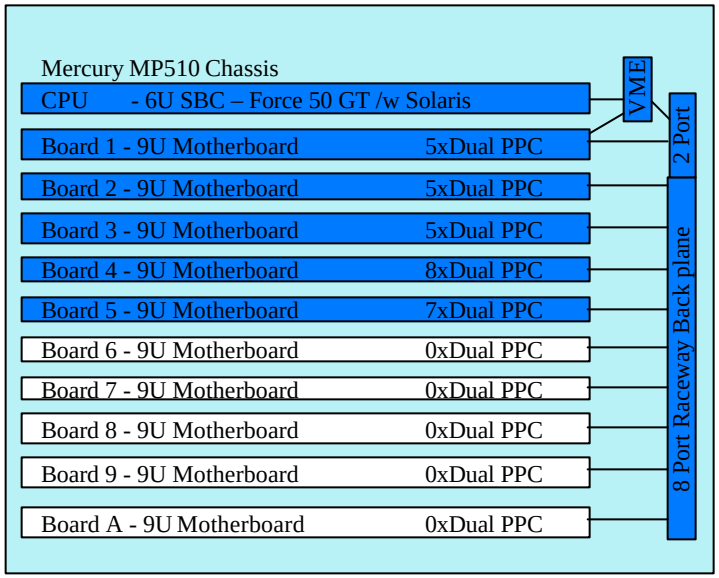
VME



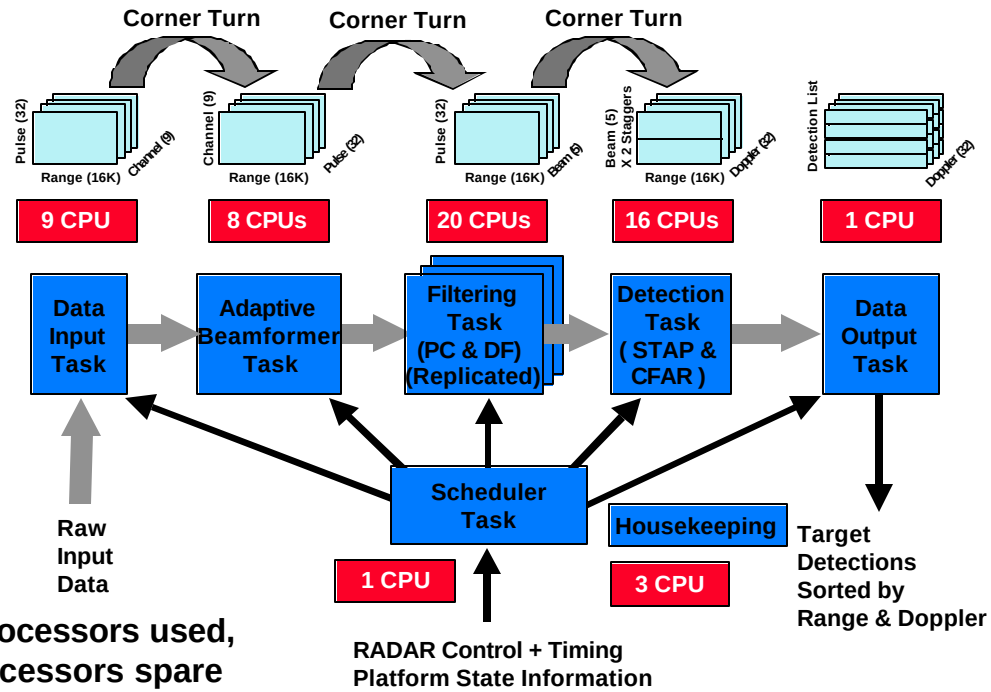


All 18 processors used

1Q FY03 Testbed Processor



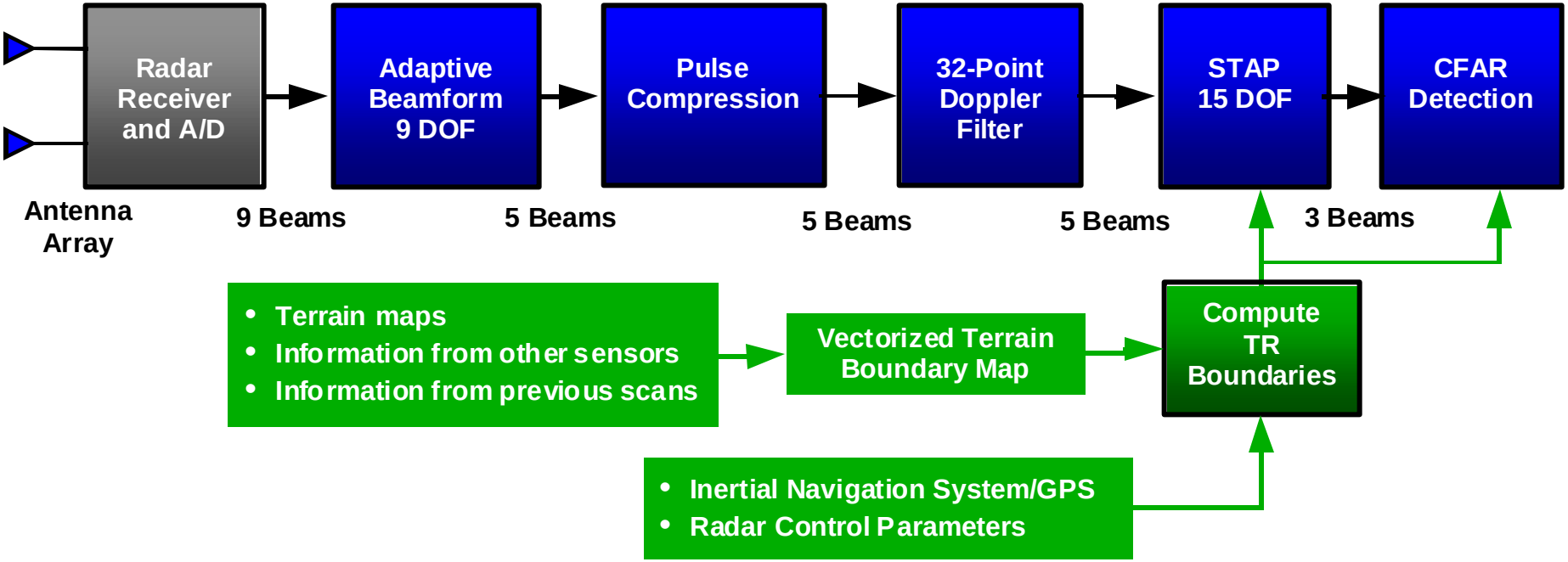
- **Ten Race++ PowerStream Motherboards with 60 500Mhz PPC**
- **240 Gflop/sec Peak Processing**



58 processors used,
2 processors spare

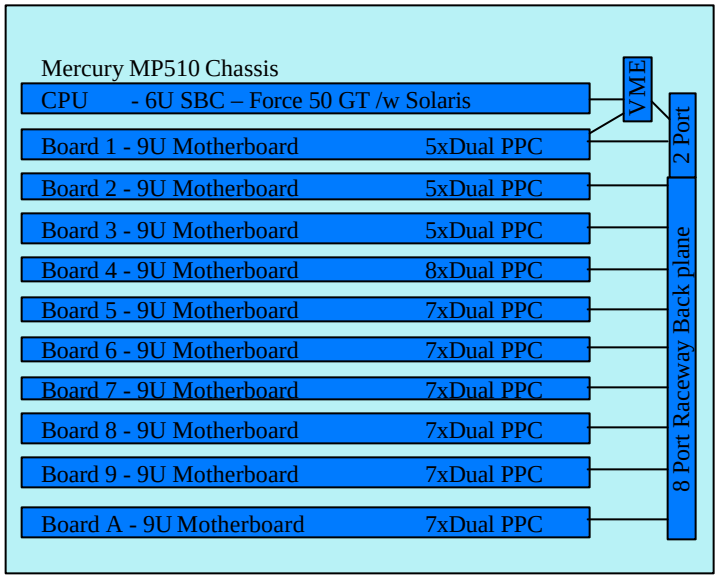
Phase 1 Signal Processing Architecture:

Terrain Boundary Training Set and Detection Modification

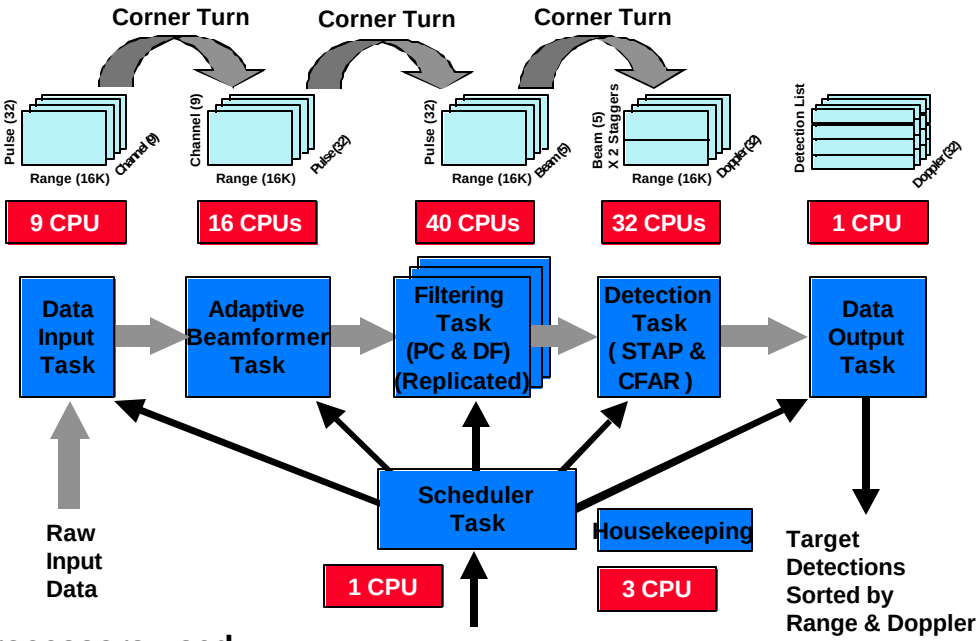


- **Full STAP algorithm stream implemented in Phase 1**
- **Processor sizing allows modifications to training set selection, detection strategies**

Phase 1 Testbed Processor



- **Ten Race++ PowerStream Motherboards /w 120 500Mhz PPC**
- **480 Gflop/sec Peak Processing**

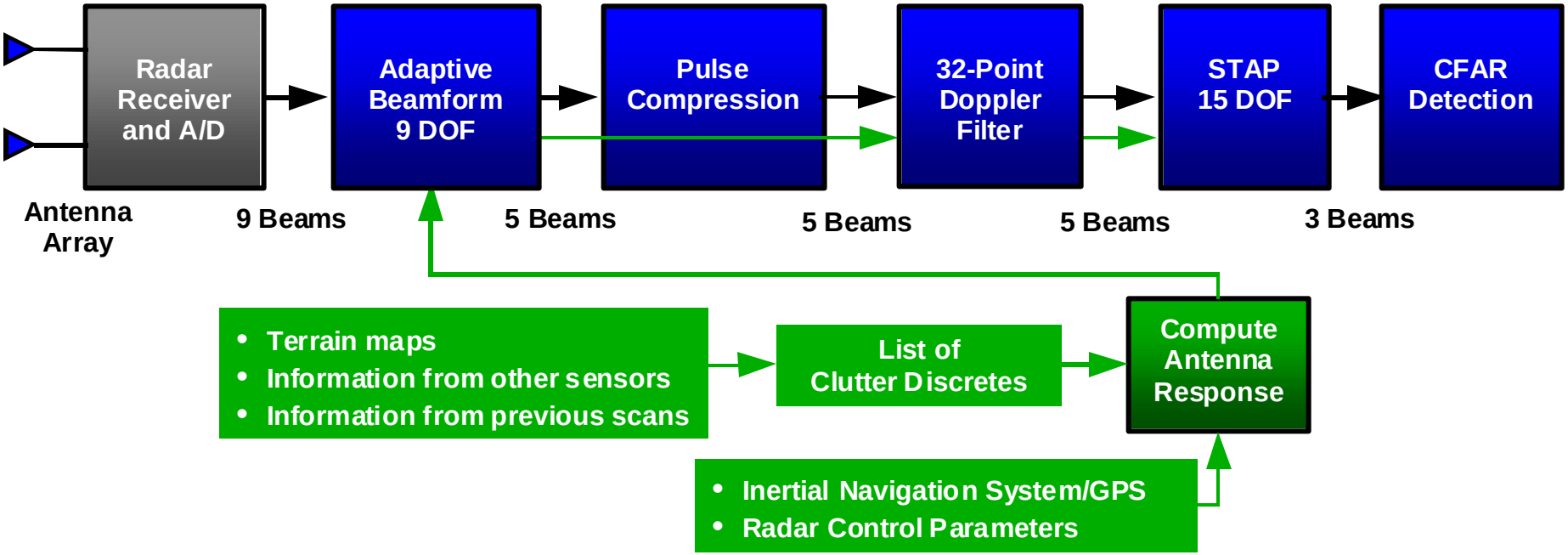


102 processors used, 18 processors spare

KASSPER SIGNAL PROCESSOR
Phase 2 Processor Sizing

- The current estimate is that the STAP processing will increase from 8.8 Gflop/sec to approx 30 Gflop/sec. This may change based on the actual Phase 2 processing algorithm which is selected.
- At 10% efficiency and 4 Gflop/sec/processor, this will require an additional 53 processors in the STAP Detection Task for a total of 155 processors
- The processor efficiency achieved in Phase 1 will be measured to allow more informed sizing decisions for Phase 2
- In FY03 and FY04 the number of CPUs will be increased to 160 processors.
- If Phase 2 algorithms grow in processing demand, the range extent to be processed can be reduced

Phase 2 Signal Processing Architecture
Full STAP Mitigation Approach



- **Phase 2 processor sizing allows algorithm changes throughout the processing chain**

Development Schedule

	FY02				FY03				FY04			
	1Q	2Q	3Q	4Q	1Q	2Q	3Q	4Q	1Q	2Q	3Q	4Q
Develop Baseline Software on G4 Cluster												
Procure 9U VME System												
Develop Phase 1 Software on 9U VME System												
FY02 Phase 1 Software Demo												
Procure Phase 1 Processor												
Phase 1 Software Optimization												
Procure Partial Phase 2 Processor Upgrade + Playback Subsystem												
Phase 1 Software Running at real-time rates												
Develop Phase 2 Processor Software + Playback Subsystem Software												
Phase 2 Software Optimization												
Procure Final Phase 2 Processor Upgrade + Raid Storage Subsystem												
Integrate Raid Storage Subsystem												
Integrate with data collected from the BAC1-11												

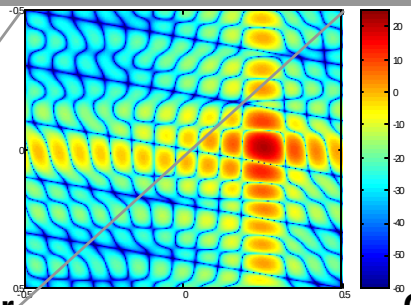
Outline

- Introduction
- Processor sizing
- Parallel Vector Library
- Multi-phase processor acquisition strategy
- Algorithm/implementation coupling
- Summary

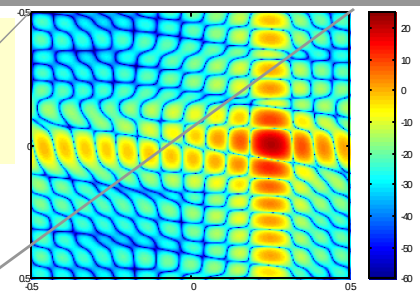
Knowledge-Aided Inter ference Rejection

Direct use of environmental knowledge base for heterogeneous clutter rejection

1st Stage Knowledge-Aided Pre-Filter Response



2nd Stage Conventional Filter



Nonstationary Clutter (plus Signal)

X

KA Pre-Filter

$R_{KA}^{?\frac{1}{2}}$

$Y \; ? \; R_{KA}^{?\frac{1}{2}} X$

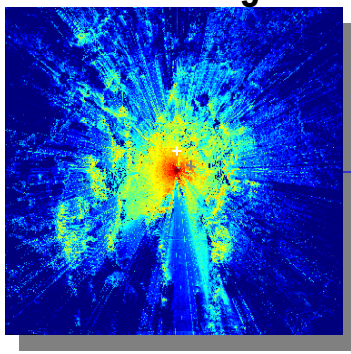
Reduced-Rank Conventional Filter

$\hat{R}_{SMI}^{?\frac{1}{2}}$

$Z \; ? \; \hat{R}_{SMI}^{?\frac{1}{2}} Y$

Detector

Clutter Knowledge Base



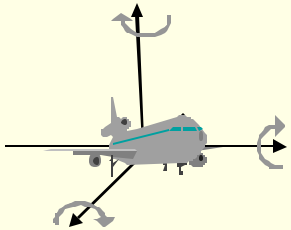
Clutter Cell Returns

$?\hat{r}_i ?$

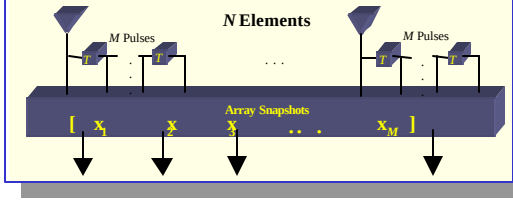
Clutter Steering Vectors

$?\hat{C}_i ?$

GPS/INS



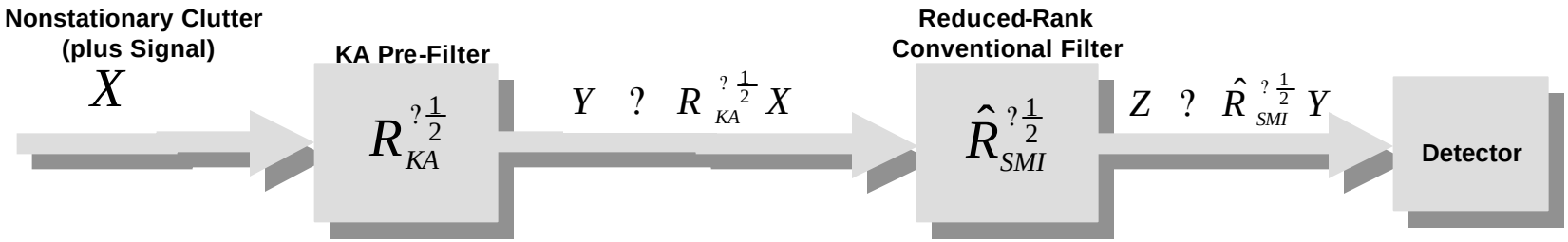
Sensor Characteristics



$$R_{KA} \; ? \; \sum_i ? \; ?_i^2 C_i C_i^H ? \; ? \; ?^2 I$$

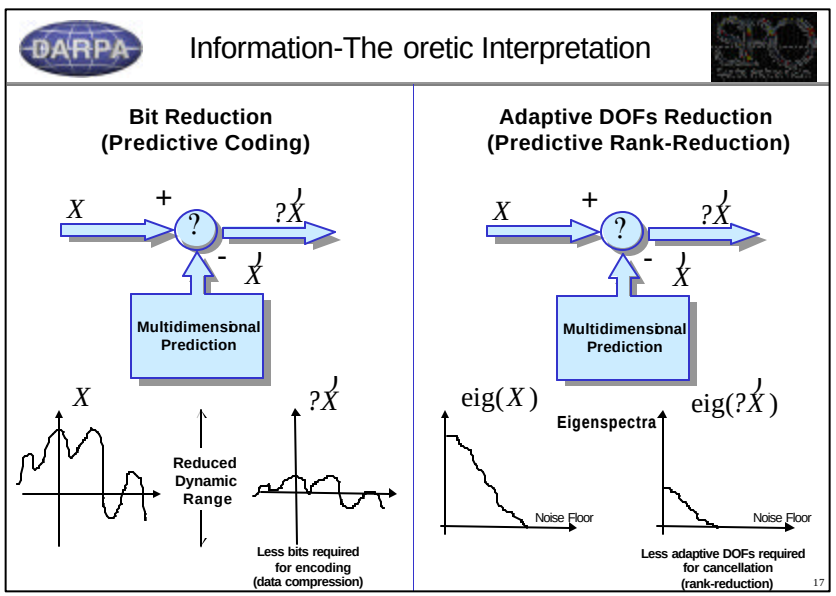
Sum Over Clutter Cells

Implementation Alternatives



$$\tilde{x} \; ? \; \mathbf{R}_{KA}^{?1/2} x$$

$$\tilde{w} \; ? \; \mathbf{R}_{KA}^{?1/2} \mathbf{R} \mathbf{R}_{KA}^{?1/2} \; ? \; ? \; \mathbf{I} \; ? \; \mathbf{R}_{KA}^{?1/2} v \; ? \; \mathbf{R}_{KA}^{1/2} \; ? \; \mathbf{R} \; ? \; ? \; \mathbf{R}_{KA} \; ? \; v$$



- **Pre-whitening by $R_{KA}^{-1/2}$ is equivalent to loading by R_{KA}**
- **DOFs not reduced for nulling, but sample support equivalent to pre-nulling**
  - **Cost of STAP pre-whitening twice can be absorbed in STAP filter**
  - **Implementational options should be considered**

Summary

- PVL will allow the staged development effort with portability to increasing size platforms
  - Cluster to single Mercury board to full chassis
- KASSPER Real Time Testbed Processor will be based on the Mercury MP-510 PowerStream system
  - 102 processors will be needed for the Phase 1 algorithm
  - 160 processors will be needed for the Phase 2 algorithm
- Interchange between algorithms and implementation necessary