Deployment of SAR and GMTI Signal Processing on a Boeing 707 Aircraft using pMatlab and a Bladed Linux Cluster

Jeremy Kepner, Tim Currie, Hahn Kim, Bipin Mathew, Andrew McCabe, Michael Moore, Dan Rabinkin, Albert Reuther, Andrew Rhoades, Lou Tella and Nadya Travinin

September 28, 2004

This work is sponsored by the Department of the Air Force under Air Force contract F19628-00-C-002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Outline

- **Introduction**
  - ***LiMIT***
  - ***Technical Challenge***
  - ***pMatlab***
  - ***"QuickLook" Concept***
- **System**
- **Software**
- **Results**
- **Summary**

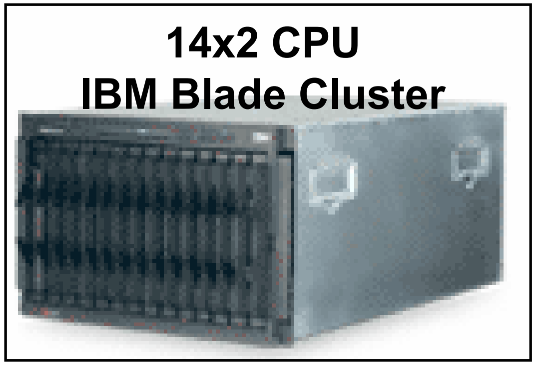
LiMIT

- **Lincoln Multifunction Intelligence, Surveillance and Reconnaissance Testbed**
  - **Boeing 707 aircraft**
  - **Fully equipped with sensors and networking**
  - **Airborne research laboratory for development, testing, and evaluation of sensors and processing algorithms**
- **Employs Standard Processing Model for Research Platform**
  - **Collect in the air/process on the ground**

Processing Challenge

- **Can we process radar data (SAR & GMTI) in flight and provide feedback on sensor performance in flight?**

- **Requirements and Enablers**
  - **Record and playback data**
    - **High speed RAID disk system**
  - **High speed network**
  - **High density parallel computing**
    - **Ruggedized bladed Linux cluster**
  - **Rapid algorithm development**
    - **pMatlab**



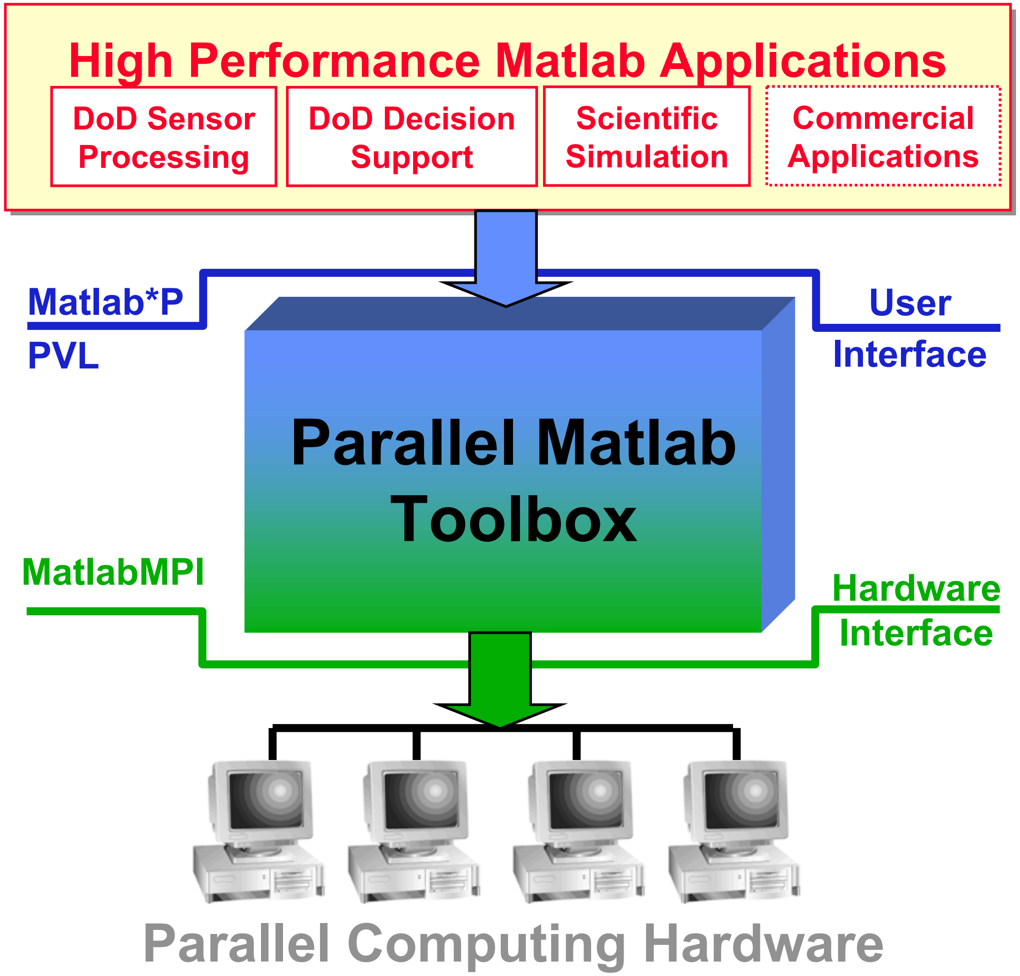
pMatlab: Parallel Matlab Toolbox

Goals

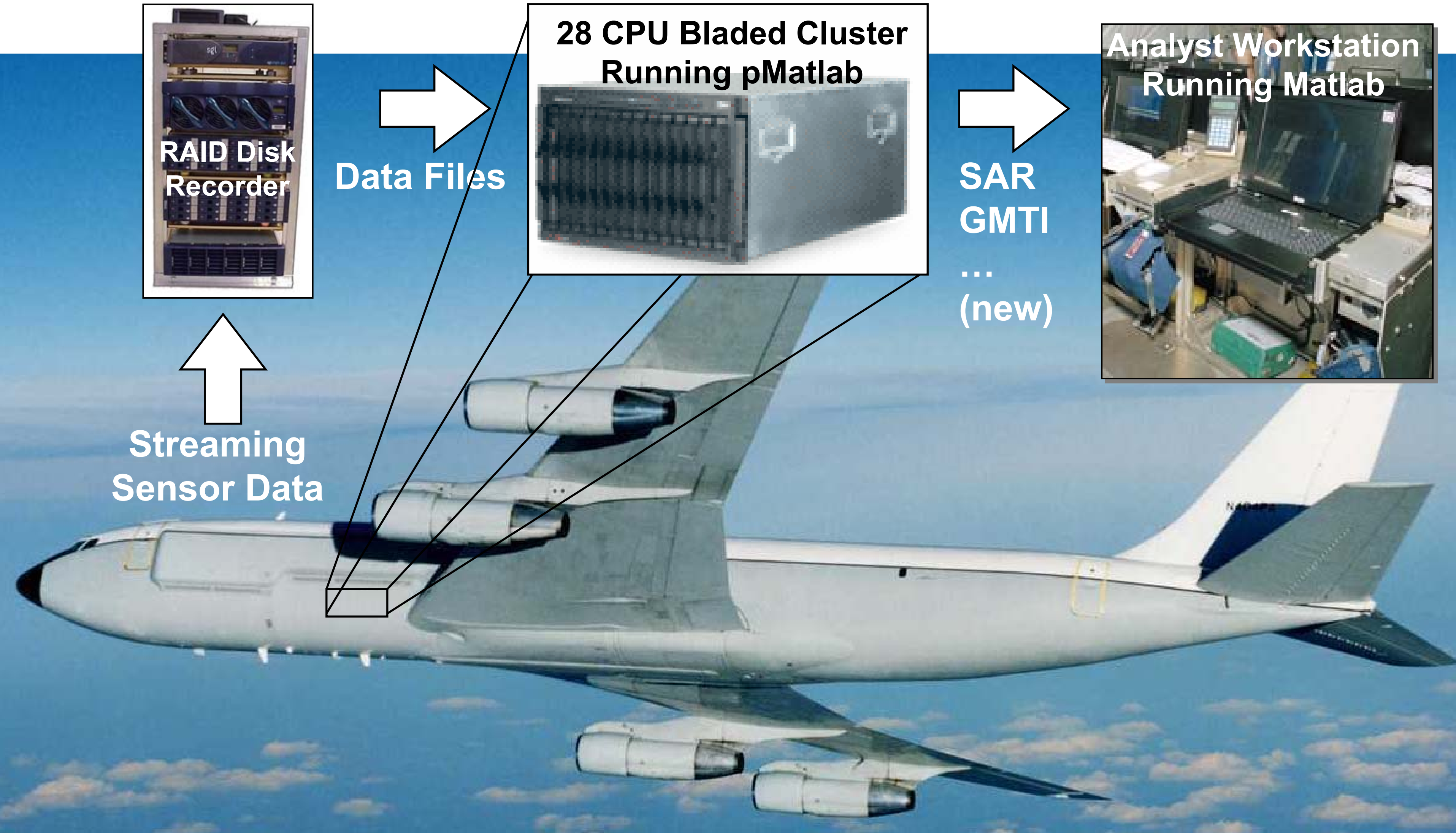
- Matlab speedup through transparent parallelism
- Near-real-time rapid prototyping

Lab-Wide Usage

- Ballistic Missile Defense
- Laser Propagation Simulation
- Hyperspectral Imaging
- Passive Sonar
- Airborne Ground Moving Target Indicator (GMTI)
- Airborne Synthetic Aperture Radar (SAR)



“QuickLook” Concept



Outline

- Introduction
- **System**
  - *ConOps*
  - *Ruggedization*
  - *Integration*
- Software
- Results
- Summary

Concept of Operations

Timeline

Record Streaming Data	~1 seconds = 1 dwell	
Copy to Bladed Cluster	~30 Seconds	
Process on Bladed Cluster	1st CPI ~ 1 minutes	2 Dwell ~2 minutes
Process on SGI	1st CPI ~ 2 minutes	2 Dwells ~1 hour

RAID Disk Recorder

600 MB/s (1x RT)

Streaming Sensor Data

Gbit Ethernet (1/4x RT rate)

Split files, Copy w/rcp

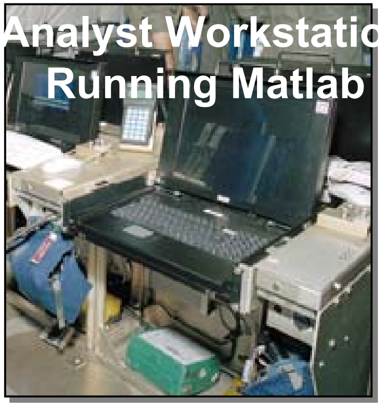
Bladed Cluster Running pMatlab

(1 TB local storage ~ 20 min data)

To Other Systems

Xwindows over Lan

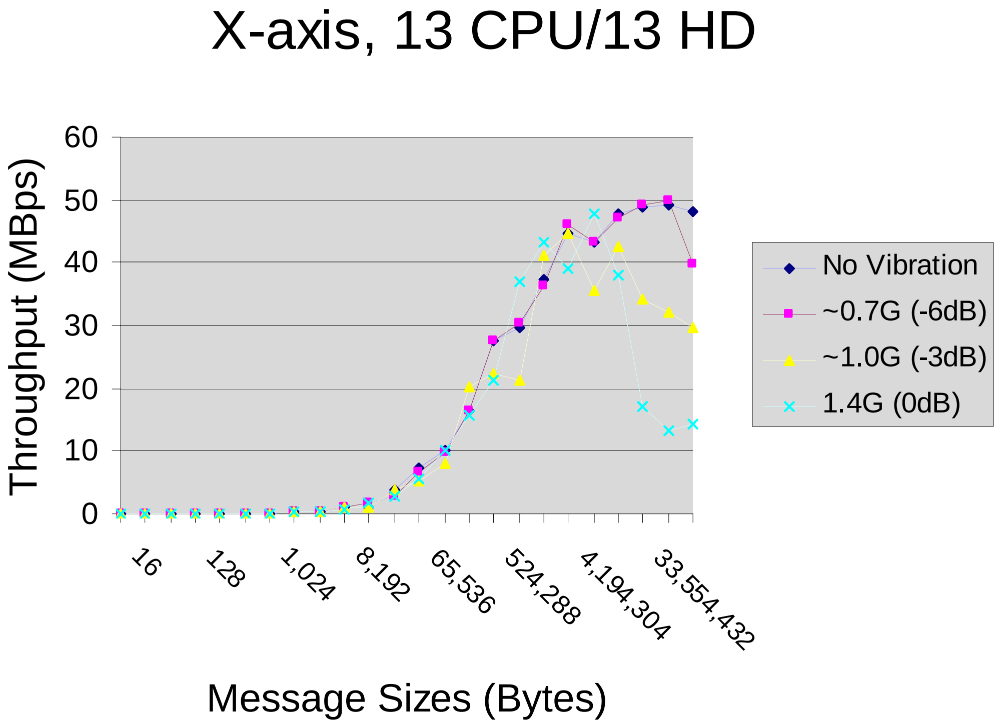
SAR
GMTI
…
(new)

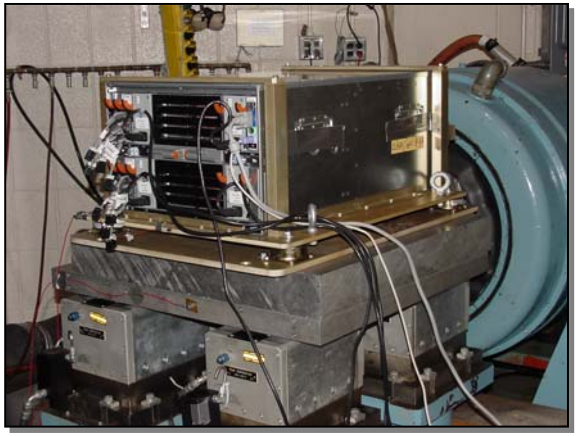


- **Net benefit: 2 Dwells in 2 minutes vs. 1 hour**

Vibration Tests

- **Tested only at operational (i.e. in-flight) levels:**
  - **0dB = 1.4G (above normal)**
  - **-3dB = ~1.0G (normal)**
  - **-6dB = ~0.7G (below normal)**
- **Tested in all 3 dimensions**
- **Ran MatlabMPI file based communication test up 14 CPUs/14 Hard drives**
- **Throughput decreases seen at 1.4 G**





Thermal Tests

- Temperature ranges
  - Test range: -20°C to 40°C
  - Bladecenter spec: 10°C to 35°C
- Cooling tests
  - Successfully cooled to -10°C
  - Failed at -20°C
  - Cargo bay typically ≥ 0°C
- Heating tests
  - Used duct to draw outside air to cool cluster inside oven
  - Successfully heated to 40°C
  - Outside air cooled cluster to 36°C

Mitigation Strategies

- **IBM Bladecenter is not designed for 707's operational environment**

- **Strategies to minimize risk of damage:**
  1. **Power down during takeoff/ landing**
     - **Avoids damage to hard drives**
     - **Radar is also powered down**
  2. **Construct duct to draw cabin air into cluster**
     - **Stabilizes cluster temperature**
     - **Prevents condensation of cabin air moisture within cluster**

Integration

SGI RAID System

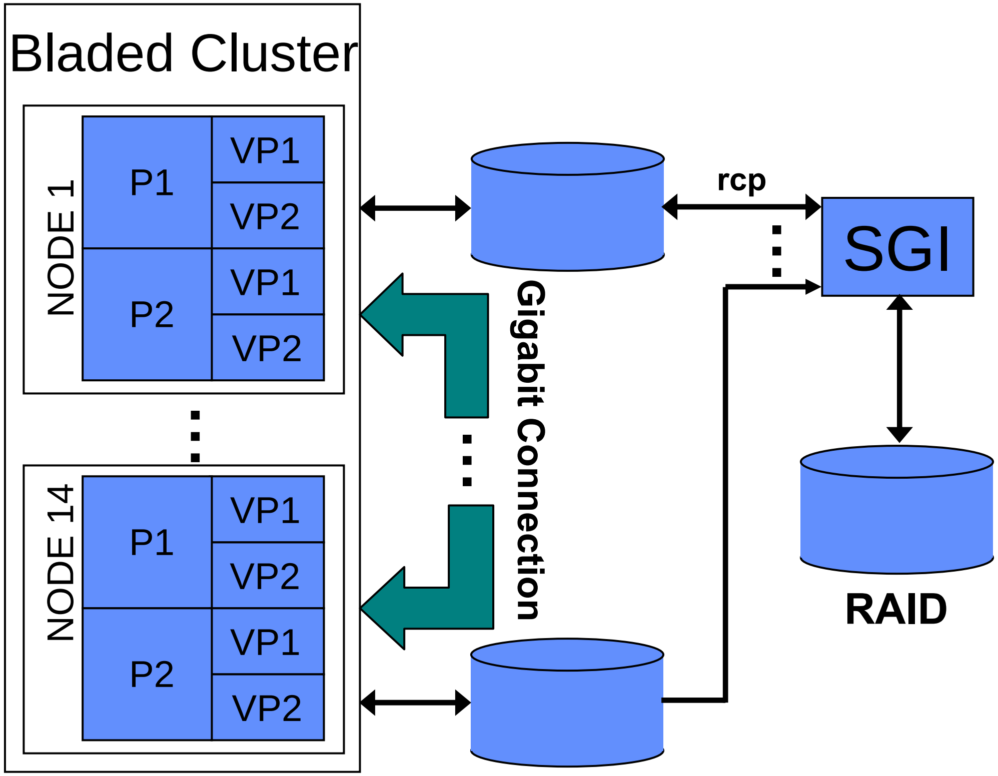
Scan catalog files, select dwells and CPIs to process (C/C shell)

Assign dwells/CPIs to nodes, package up signature / aux data, one CPI per file. Transfer data from SGI to each processor's disk (Matlab)

IBM Bladed Cluster

Nodes process CPIs in parallel, write results onto node 1's disk. Node 1 processor performs final processing

Results displayed locally

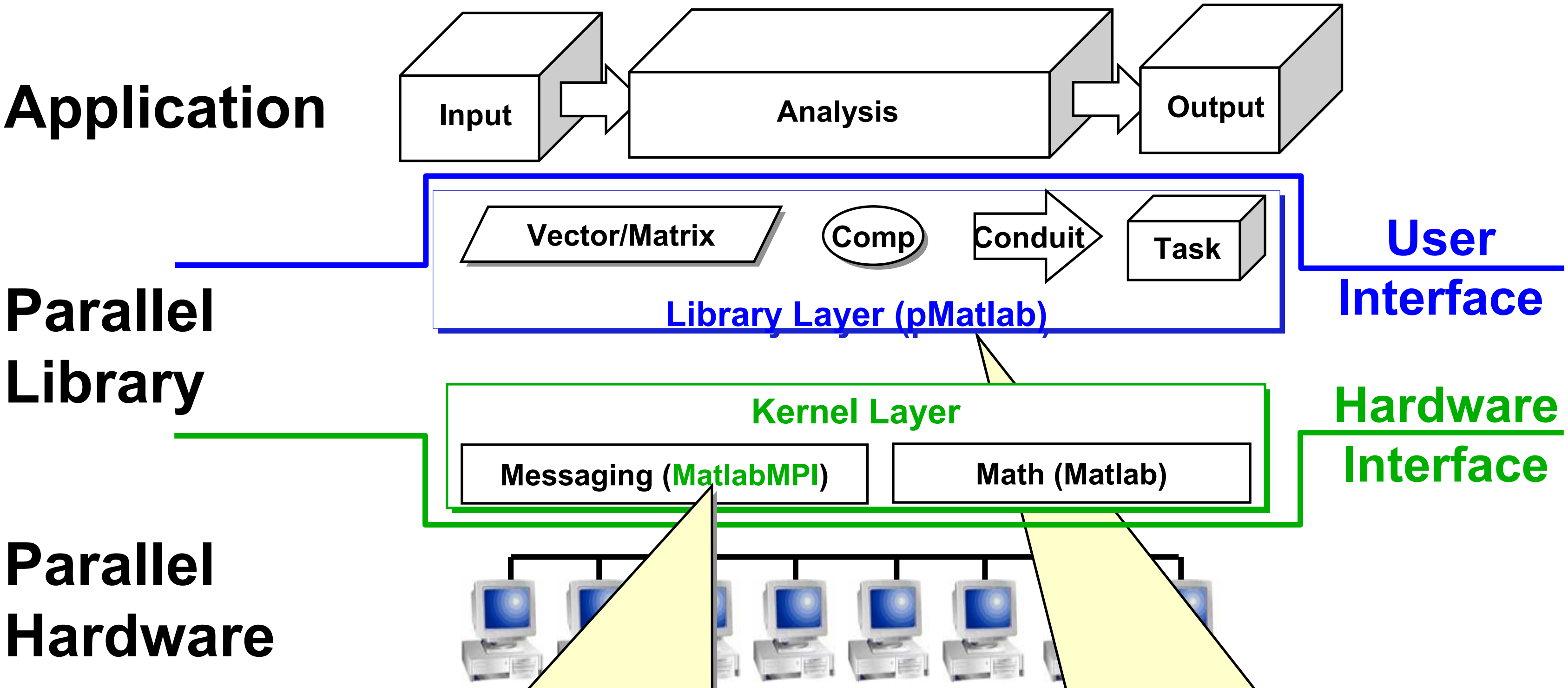


- **pMatlab allows integration to occur while algorithm is being finalized**

Outline

- Introduction
- Hardware
- **Software**
  - *pMatlab architecture*
  - *GMTI*
  - *SAR*
- Results
- Summary

MatlabMPI & pMatlab Software Layers



- Can build a parallel library with a few messaging primitives
- MatlabMPI provides this messaging capability:

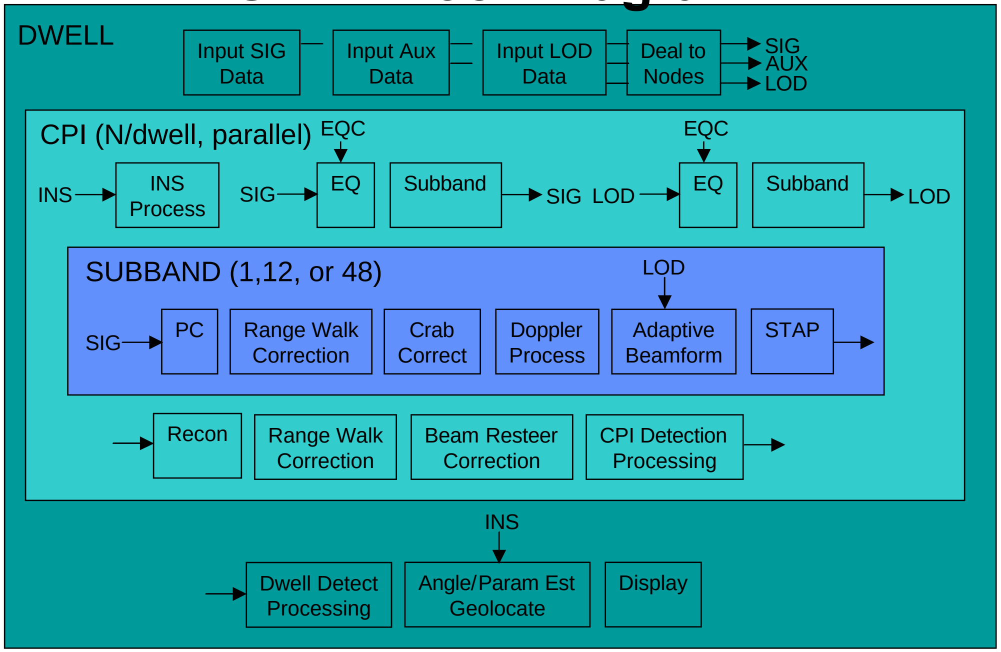
```
MPI_Send(dest,comm,tag,X);
X = MPI_Recv(source,comm,tag);
```

- Can build applications with a few parallel structures and functions
- pMatlab provides parallel arrays and functions

```
X = ones(n,mapX);
Y = zeros(n,mapY);
Y(:,:) = fft(X);
```

LiMIT GMTI

GMTI Block Diagram



Parallel Implementation

Approach

Deal out CPIs to different CPUs

Performance

TIME/NODE/CPI	~100 sec
TIME FOR ALL 28 CPIS	~200 sec
Speedup	~14x

- **Demonstrates pMatlab in a large multi-stage application**
  - ~13,000 lines of Matlab code
- **Driving new pMatlab features**
  - Parallel sparse matrices for targets (dynamic data sizes)
    - Potential enabler for a whole new class of parallel algorithms
    - Applying to DARPA HPCS GraphTheory and NSA benchmarks
  - Mapping functions for system integration
  - Needs expert components!

GMTI pMatlab Implementation

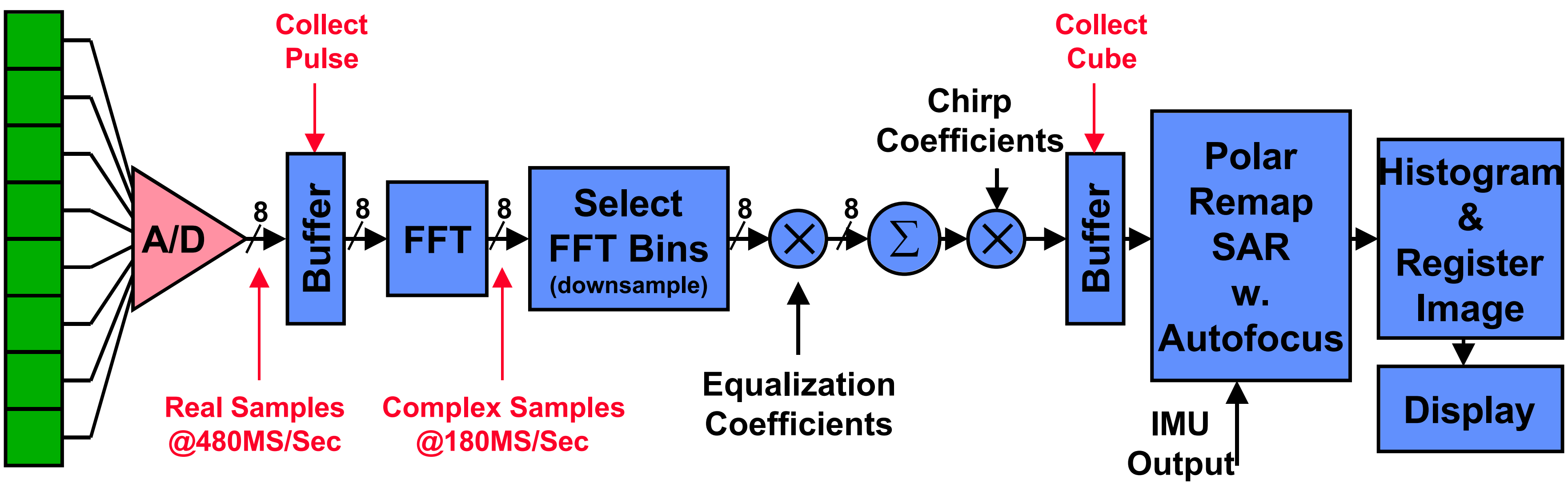
- **GMTI pMatlab code fragment**

```
% Create distribution spec: b = block, c = cyclic.
dist_spec(1).dist = 'b';
dist_spec(2).dist = 'c';

% Create Parallel Map.
pMap = map([1 MAPPING.Ncpus],dist_spec,0:MAPPING.Ncpus-1);

% Get local indices.
[lind.dim_1_ind lind.dim_2_ind] = global_ind(zeros(1,C*D,pMap));

% loop over local part
for index = 1:length(lind.dim_2_ind)
        ...
end
```

- **pMatlab primarily used for determining which CPIs to work on**
  - **CPIs dealt out using a cyclic distribution**

LiMIT SAR

SAR Block Diagram



- **Most complex pMatlab application built (at that time)**
  - ~4000 lines of Matlab code
  - CornerTurns of ~1 GByte data cubes
- **Drove new pMatlab features**
  - Improving Corner turn performance
    - Working with Mathworks to improve
  - Selection of submatrices
    - Will be a key enabler for parallel linear algebra (LU, QR, …)
  - Large memory footprint applications
    - Can the file system be used more effectively

SAR pMatlab Implementation

- **SAR pMatlab code fragment**

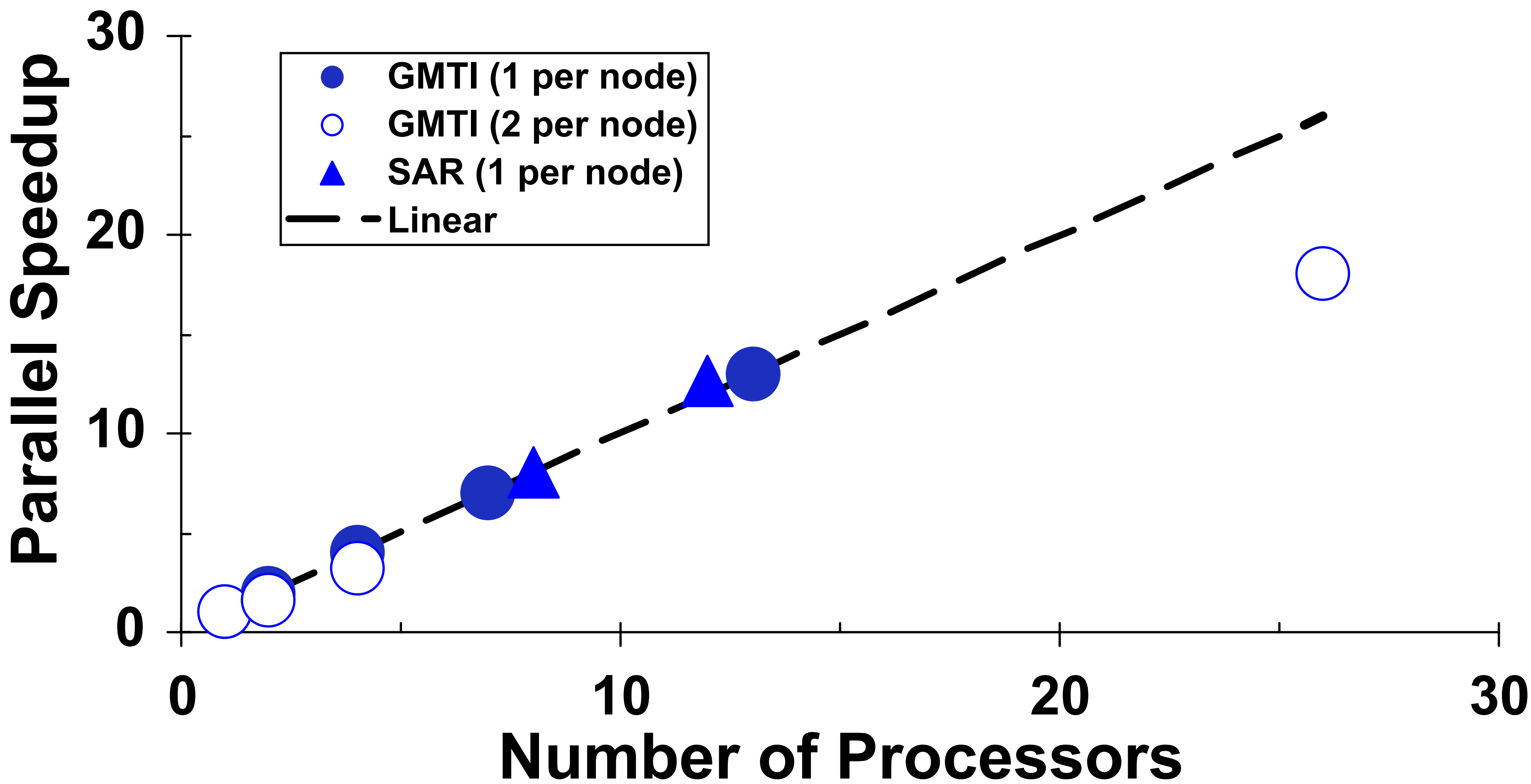
```
% Create Parallel Maps.
mapA = map([1 Ncpus],0:Ncpus-1);
mapB = map([Ncpus 1],0:Ncpus-1);

% Prepare distributed Matrices.
fd_midc=zeros(mw,TotalnumPulses,mapA);
fd_midr=zeros(mw,TotalnumPulses,mapB);

% Corner Turn (columns to rows).
fd_midr(:,:) = fd_midc;
```

- **Cornerturn Communication performed by overloaded '=' operator**
  - Determines which pieces of matrix belongs where
  - Executes appropriate MatlabMPI send commands

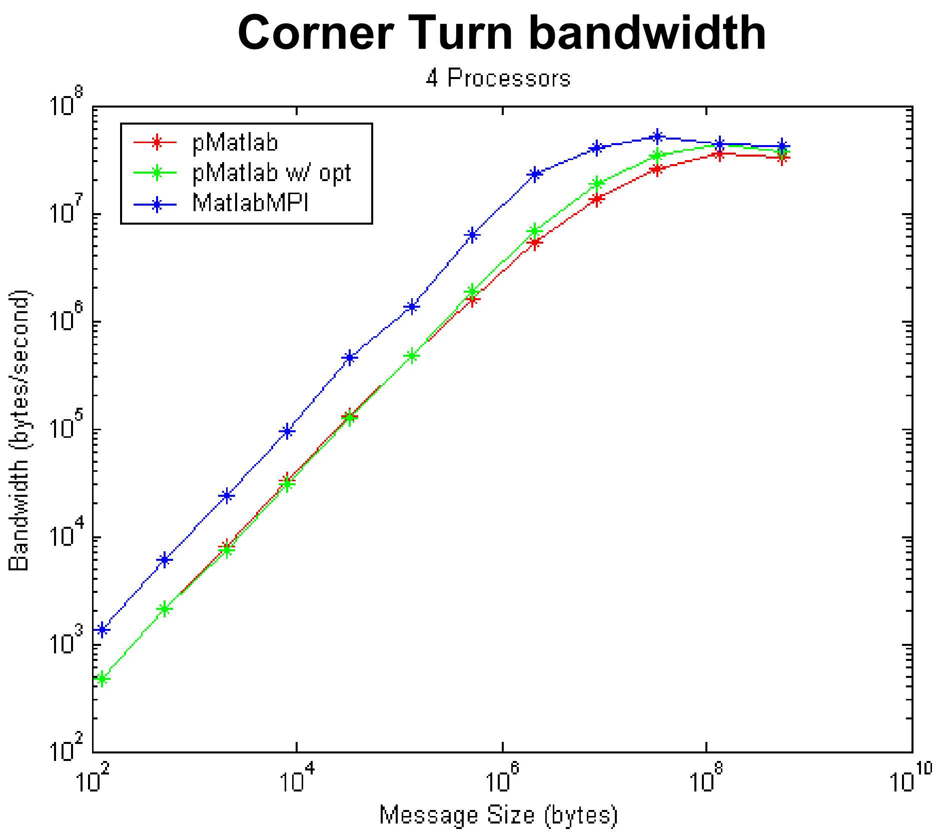
Outline

- Introduction
- Implementation
- **Results**
  - ***Scaling Results***
  - ***Mission Results***
  - ***Future Work***
- Summary

Parallel Performance



SAR Parallel Performance



- **Application memory requirements too large for 1 CPU**
  - **pMatlab a requirement for this application**
- **Corner Turn performance is limiting factor**
  - **Optimization efforts have improved time by 30%**
  - **Believe additional improvement is possible**

July Mission Plan

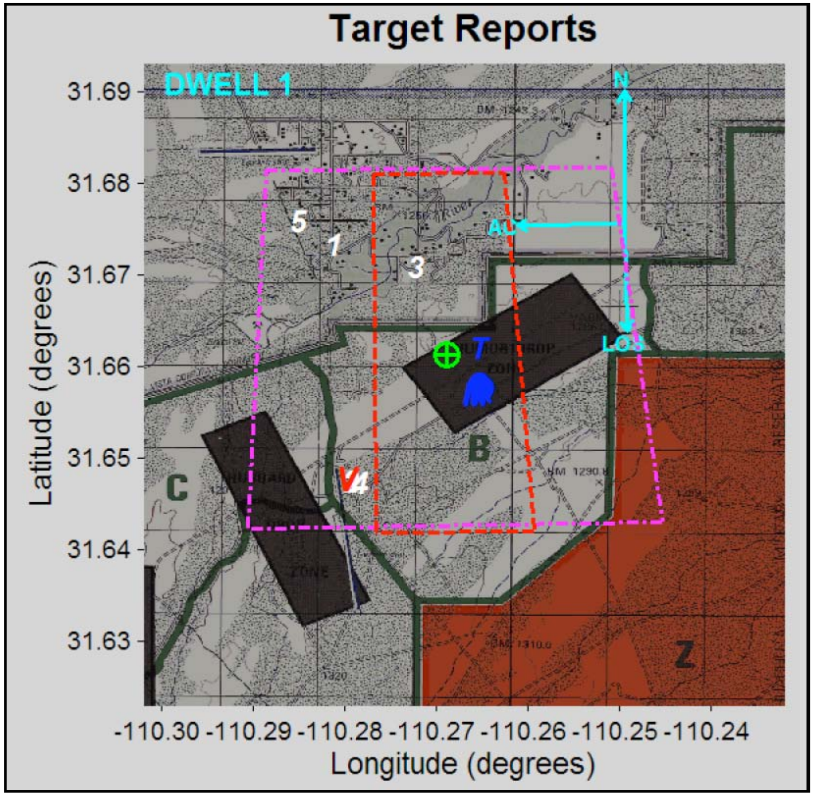
- **Final Integration**
  - Debug pMatlab on plane
  - Working ~1 week before mission (~1 week after first flight)
  - Development occurred during mission

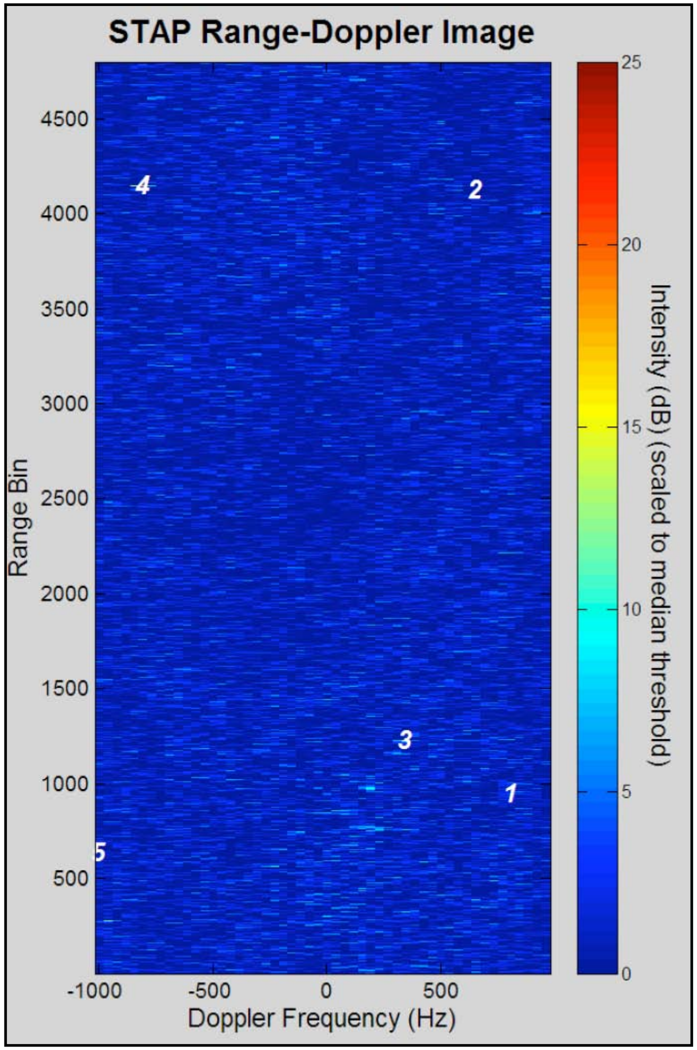
- **Flight Plan**
  - Two data collection flights
  - Flew a 50 km diameter box
  - Six GPS-instrumented vehicles
    - Two 2.5T trucks
    - Two CUCV's
    - Two M577's

July Mission Environment

- **Stressing desert environment**

July Mission GMTI results

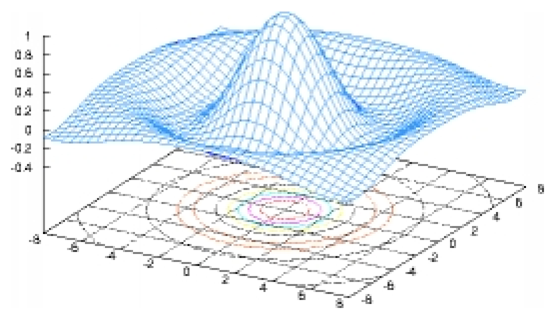




- GMTI successfully run on 707 in flight
  - Target reports
  - Range Doppler images
- Plans to use QuickLook for streaming processing in October mission

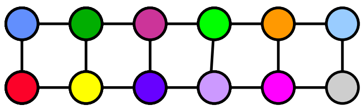
Embedded Computing Alternatives

- **Embedded Computer Systems**
  - Designed for embedded signal processing
  - Advantages
    1. Rugged - Certified Mil Spec
    2. Lab has in-house experience
  - Disadvantage
    1. Proprietary OS ⇒ No Matlab
- **Octave**
  - Matlab "clone"
  - Advantage
    1. MatlabMPI demonstrated using Octave on SKY computer hardware
  - Disadvantages
    1. Less functionality
    2. Slower?
    3. No object-oriented support ⇒ No pMatlab support ⇒ Greater coding effort

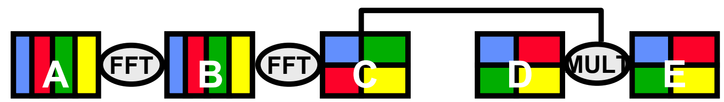


Petascale pMatlab

- **pMapper: automatically finds best parallel mapping**

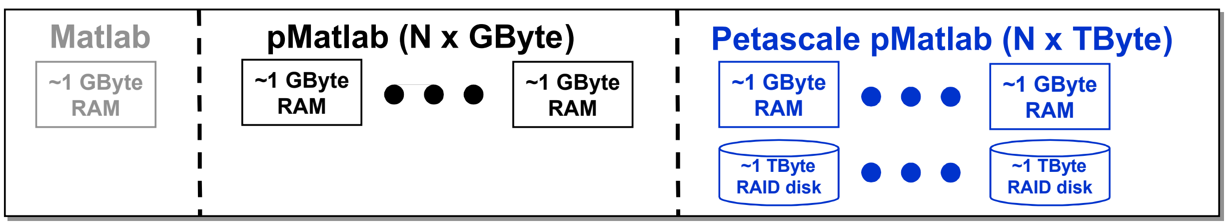


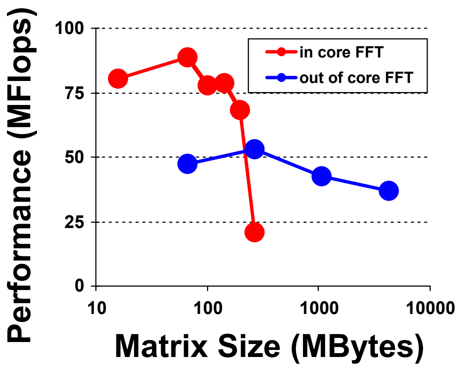
Parallel Computer



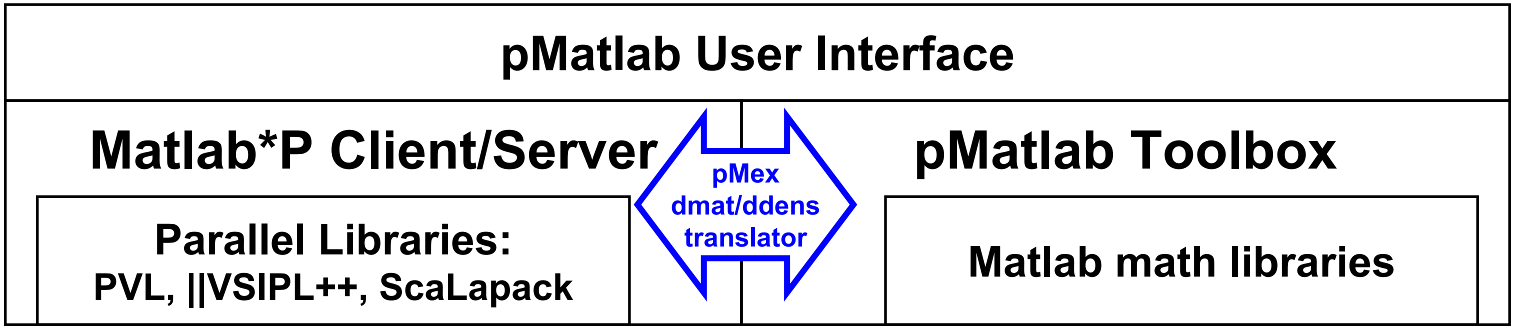
Optimal Mapping

- **pOoc: allows disk to be used as memory**





- **pMex: allows use of optimized parallel libraries (e.g. PVL)**



Summary

- **Airborne research platforms typically collect and process data later**
- **pMatlab, bladed clusters and high speed disks enable parallel processing in the air**
  - Reduces execution time from hours to minutes
  - Uses rapid prototyping environment required for research
- **Successfully demonstrated in LiMIT Boeing 707**
  - First ever in flight use of bladed clusters or parallel Matlab
- **Planned for continued use**
  - Real Time streaming of GMTI to other assets
- **Drives new requirements for pMatlab**
  - Expert mapping
  - Parallel Out-of-Core
  - pmex