



Resource Management for Digital Signal Processing via Distributed Parallel Computing

Albert I. Reuther & Joel Goodman

HPEC 2002

25 Sept 2002

This work is sponsored by the Defense Advanced Research Projects Agency (DARPA) 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 U.S. Government.

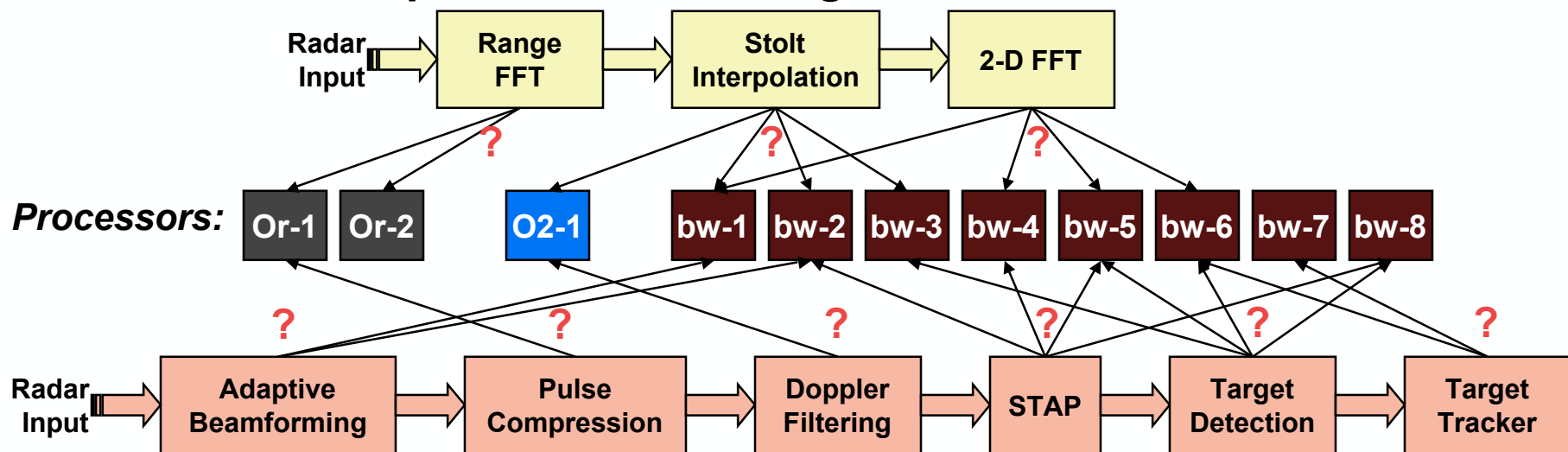
MIT Lincoln Laboratory



Problem: Mapping Task & Data Parallel DSP Computations onto Distributed Resources

- Large number of sensor streams
- Large number of resources
- Balance computational and network resource usage
- Fault tolerance

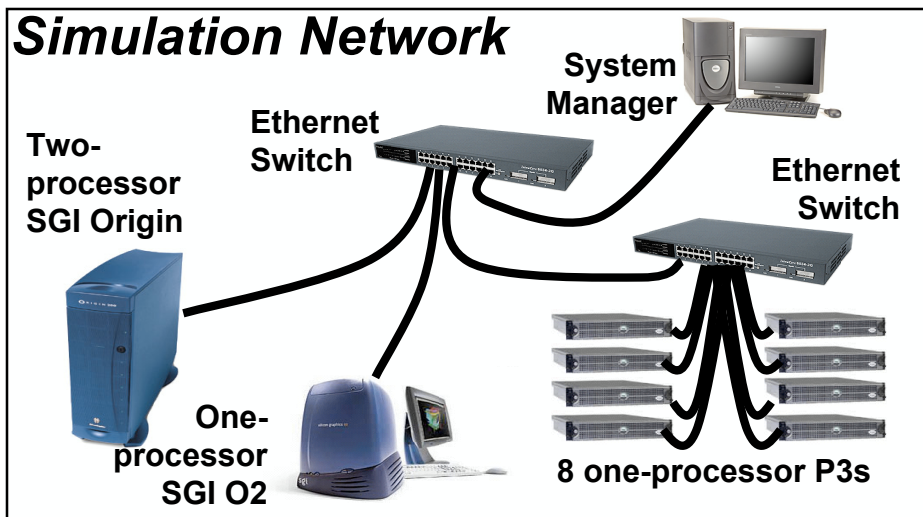
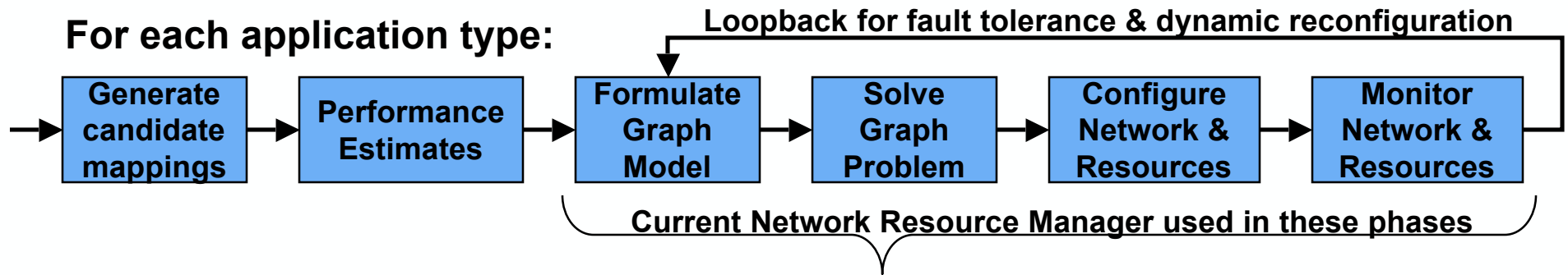
Multiple SAR Processing Chain



Multiple GMTI Radar Processing Chain



A Solution: Network Resource Manager



Network Resource Manager (NRM)

- Optimize for throughput, latency, or resource utilization
- Accommodate fault tolerance
- Manage multiple application instantiations

Future NRM

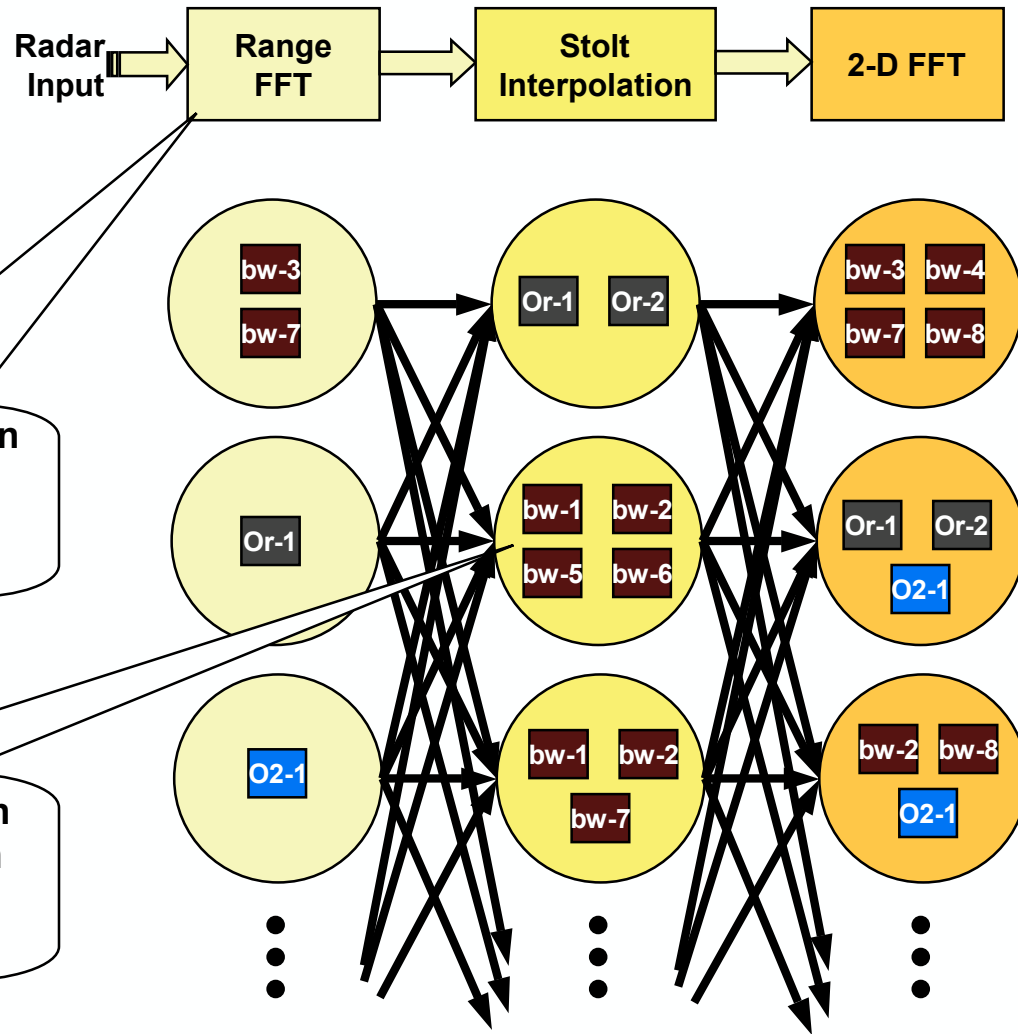
- Automatically generate candidate mappings
- Automatically conduct performance estimates



Processing Chains and Mappings

- Organize applications as processing chains
- Create mapping sets of processors

SAR Processing Chain

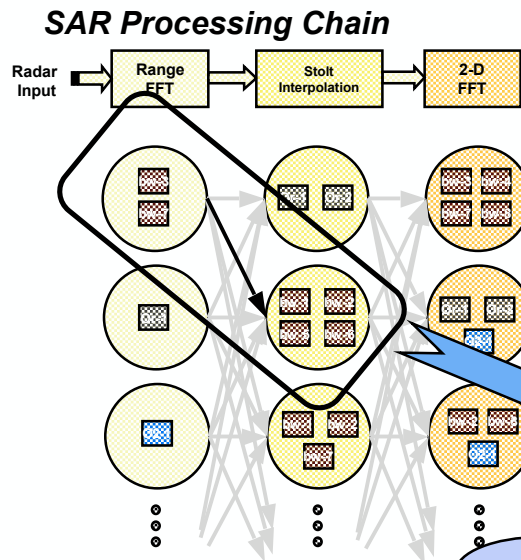


Sample processing chain for SAR - creates task parallel processing pipeline

Mapping of Stolt Interpolation stage in data parallel fashion onto processors bw-1, bw-2, bw-5, and bw-6

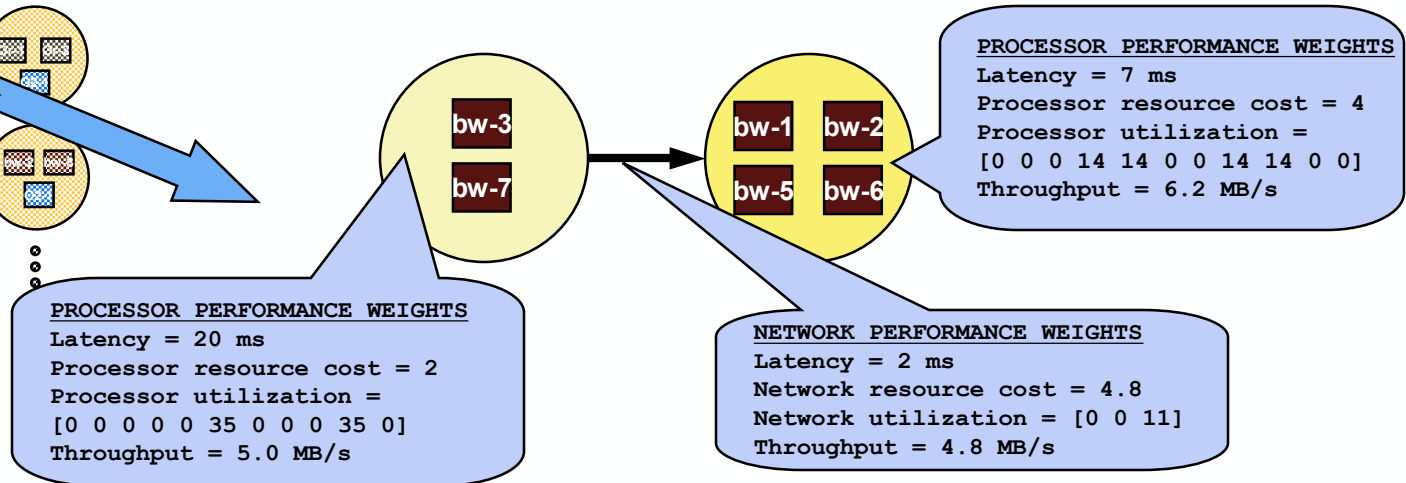


Estimate Performance



Estimate performance graph weight measurements using benchmarking code

- By column set pair mappings (for smaller graphs)
- By individual mappings (with network load est.)



System is dedicated to NRM needs:

- Dedicated processors (no extraneous services)
- Dedicated network links using dynamic VPNs with guaranteed QoS



Graph Search of Mapping Graph

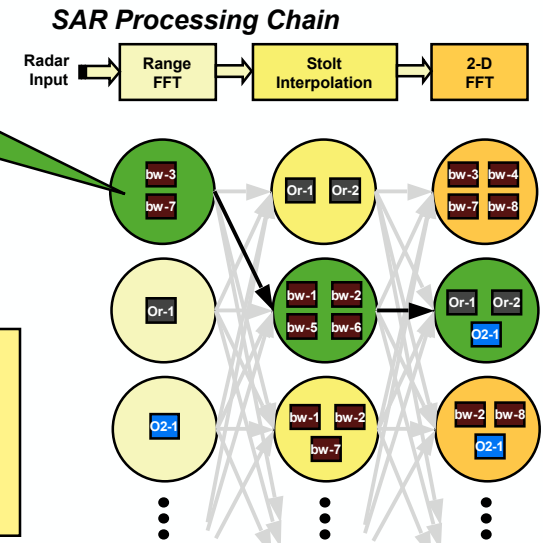
Must find path through mappings

- that optimize on one performance weight,
- while satisfying the constraints of the other performance weights

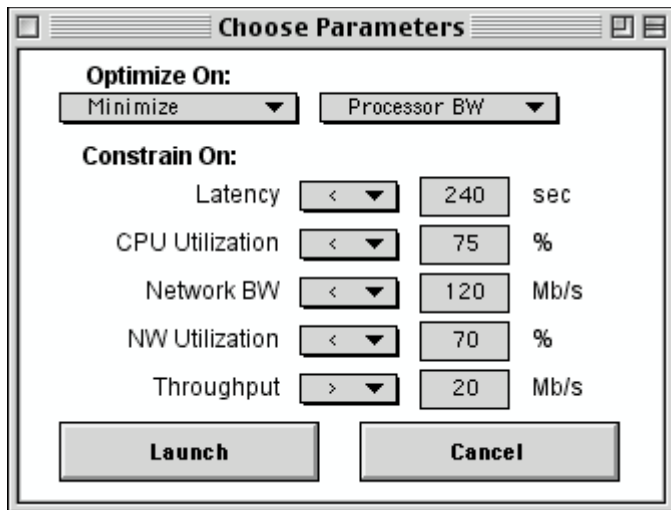
Finding best path through huge graphs is NP-Complete

- Must use a heuristic algorithm
- Decision Directed Learning (DDL) algorithm

Best path through mappings



These graphs typically have 1,000s of vertices



Performance Weights

- Latency
- Processor resource cost (aggregate bw)
- Processor resource utilization
- Network resource cost (aggregate bw)
- Network resource utilization
- Throughput
- *Memory (future)*



DDL Algorithm vs. Genetic Algorithm

Considered application-tailored exhaustive search, dynamic programming, DDL, and genetic algorithm graph search solutions

<u>Attribute</u>	<u>DDL algorithm</u> ^[1]	<u>Genetic Algorithm</u> ^[2]
Evaluation of paths	Each link individually	Whole path at a time
Quickly finds feasible solution	Yes	No
Searches for optimal solution	Yes	Yes
Dynamic reallocation of resources (for fault tolerance & managing multiple apps)	Quick and easy	Time consuming and difficult
Summary of functionality	<ul style="list-style-type: none">•Positive/negative reinforced scoring for each link•Adaptively modifies link scores based on quality of path selection	<ul style="list-style-type: none">•Crossover and mutation of paths in attempt to find better path•Elitism keeps best solution for next generation



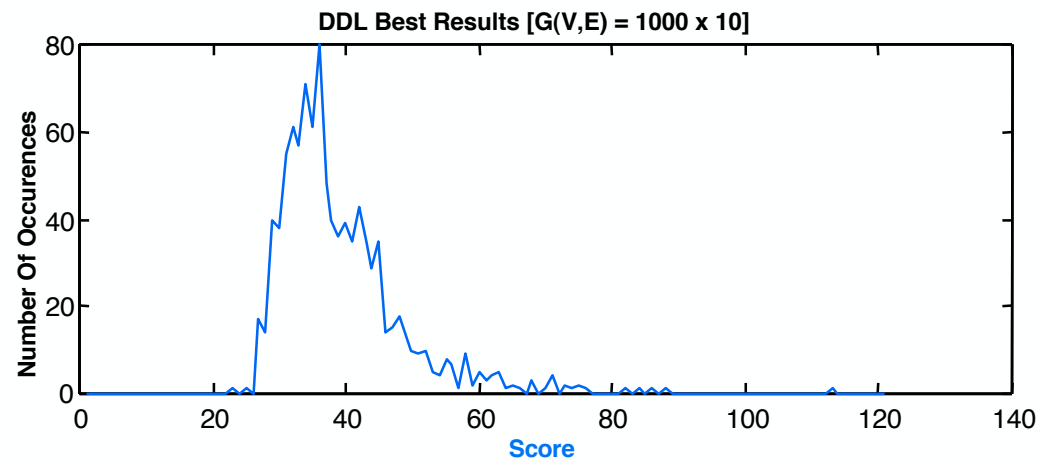
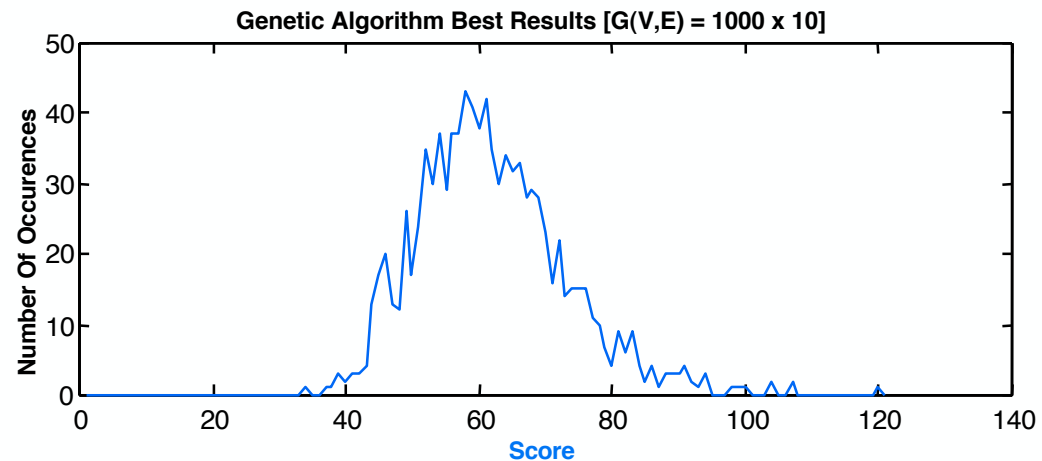
NRM Search Result Performance

Comparing DDL and 'Graph Search-Optimized' Genetic Algorithms (GA)

1000-Randomly Generated Weighted Graphs – 1000 vertices by 10 stages

- **Objective:** Minimize **Score** of Processor Resource Cost
- **Inequality Constraints:** Processor Utilization, Network Resource Cost, Network Utilization, Latency and Throughput

- **Genetic Algorithms:** 5% of cases had no solution
- **Decision Directed Learning** always finds solution
- **DDL:** ~35% improvement in solution quality over GA



(Greater concentration toward left is better.)



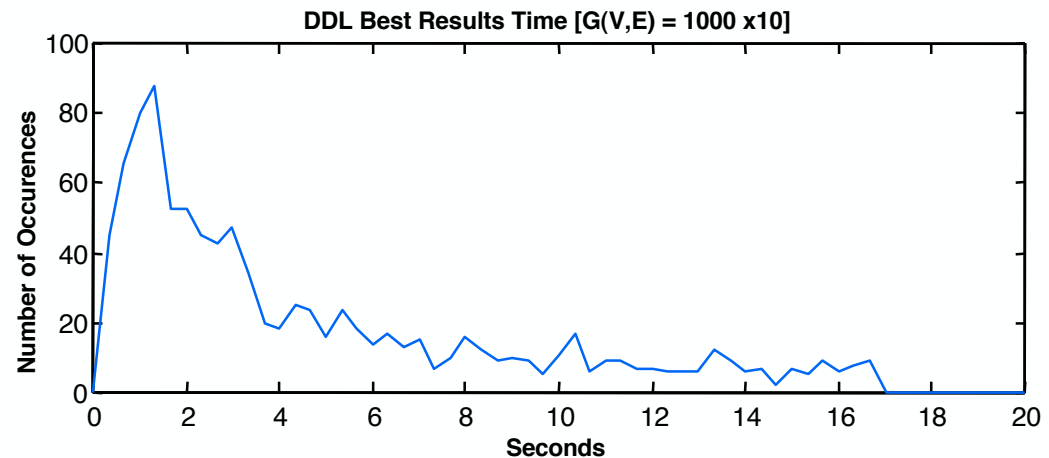
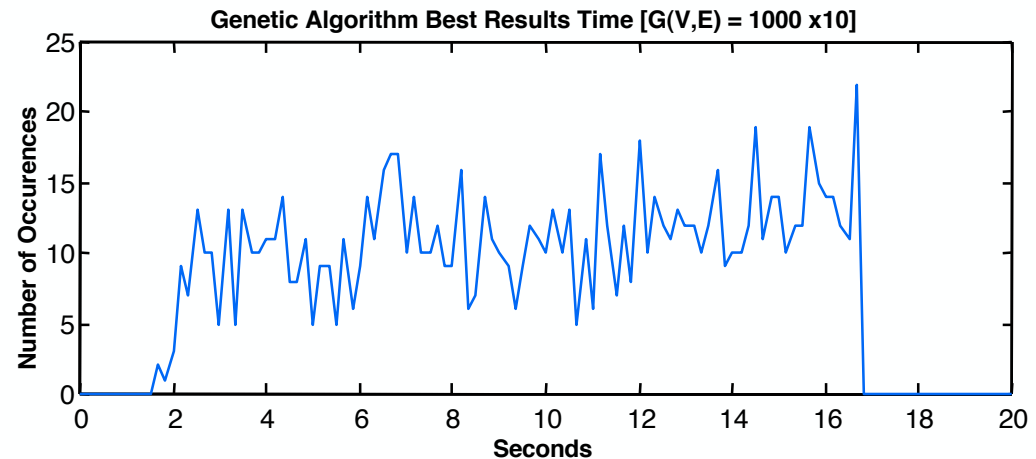
NRM Search Time Performance

Comparing DDL and 'Graph Search-Optimized' Genetic Algorithms (GA)

1000-Randomly Generated Weighted Graphs – 1000 vertices by 10 stages

- Objective:** Minimize Score of Processor Resource Cost
- Inequality Constraints:** Processor Utilization, Network Resource Cost, Network Utilization, Latency and Throughput

Decision Directed Learning found solution in less time on average than GA



(Greater concentration toward left is better.)

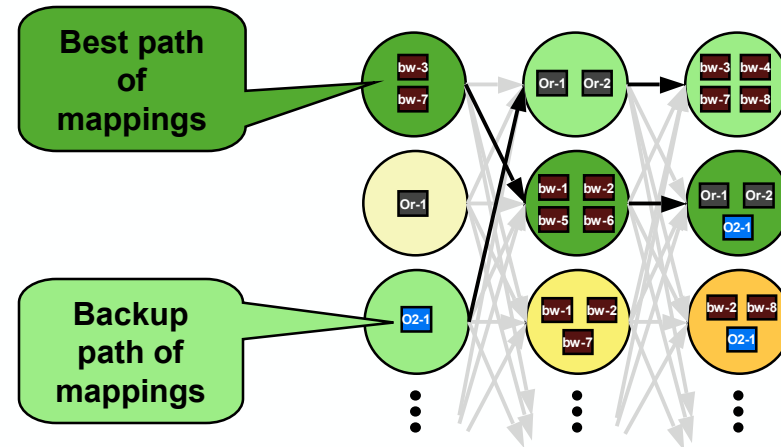
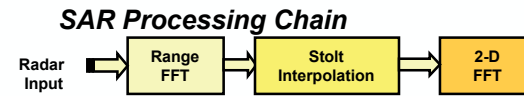


NRM and Fault Tolerance

NRM supports these fault tolerance modes

Two fault tolerance policies

- **Single-mapping fail-over**
 - Higher overhead,
 - But less resource switching
- **Full-path fail-over**
 - Lower overhead,
 - But more resource switching



Time to fail-over

- **Immediate**
 - Fast fail-over,
 - But not best possible path
- **Recalculate new path**
 - Slow fail-over,
 - But best possible path



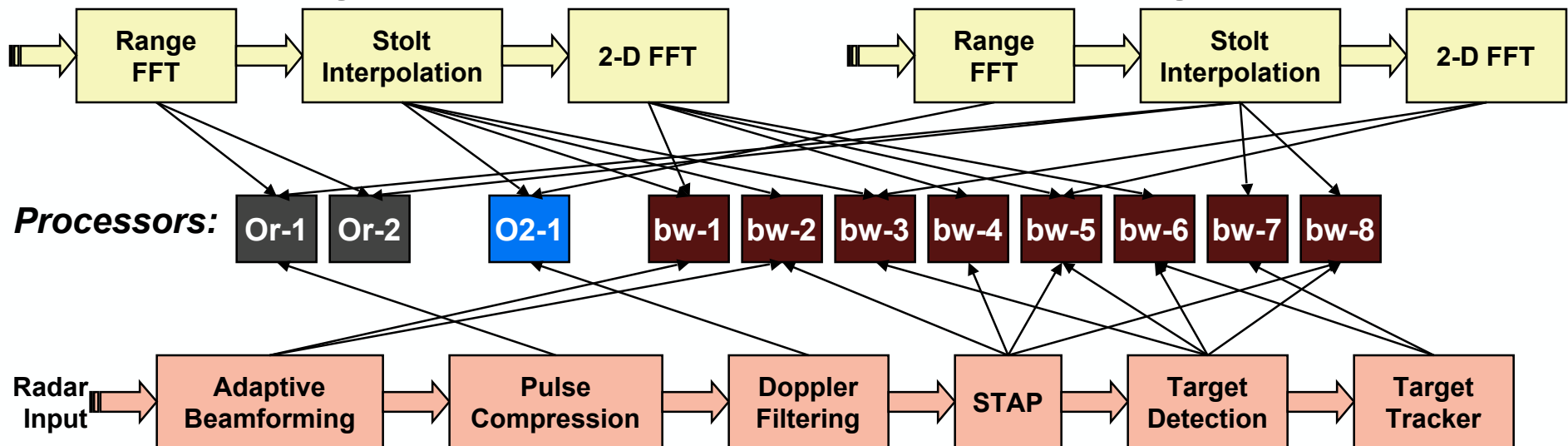
Mapping Multiple Applications on Shared Hardware

- Additive processor latency, bandwidth, and throughput
- Overlapping processor utilization for each processor
- Additive network resource latency, bandwidth, and throughput
- Overlapping network resource utilization for each resource

NRM keeps track of all latencies, bandwidths, utilizations, & throughputs

SAR Processing Chain #1

SAR Processing Chain #2



GMTI Radar Processing Chain



Summary & Future Work

- **Decision-Directed Learning algorithm best for finding best path of mappings in graph model**
 - DDL finds a feasible path every time
 - DDL finds paths that have 35% better quality over Genetic Algorithm
 - DDL finds best paths significantly faster than Genetic Algorithm and Brute Force
- **Network Resource Manager accommodates:**
 - different fault tolerance modes and
 - mapping multiple applications onto shared resources
- **Future Work**
 - Integrate with Lincoln Parallel Vector Library (PVL)
 - Study candidate mapping generation and performance estimation
 - Develop hierarchy and federation of Network Resource Managers (WAN, LAN managers)