

An Efficient Associative Processor Solution to Air Traffic Control

Mike Yuan*

Johnnie Baker*

Frank Drews#

Lev Neiman#

Will C. Meilander+

(* Kent State University, # Ohio University, + Retired, Goodyear Aerospace
& Kent State University)

Problems plague new air traffic control computers

By JOAN LOWY (AP) – April 22, 2010

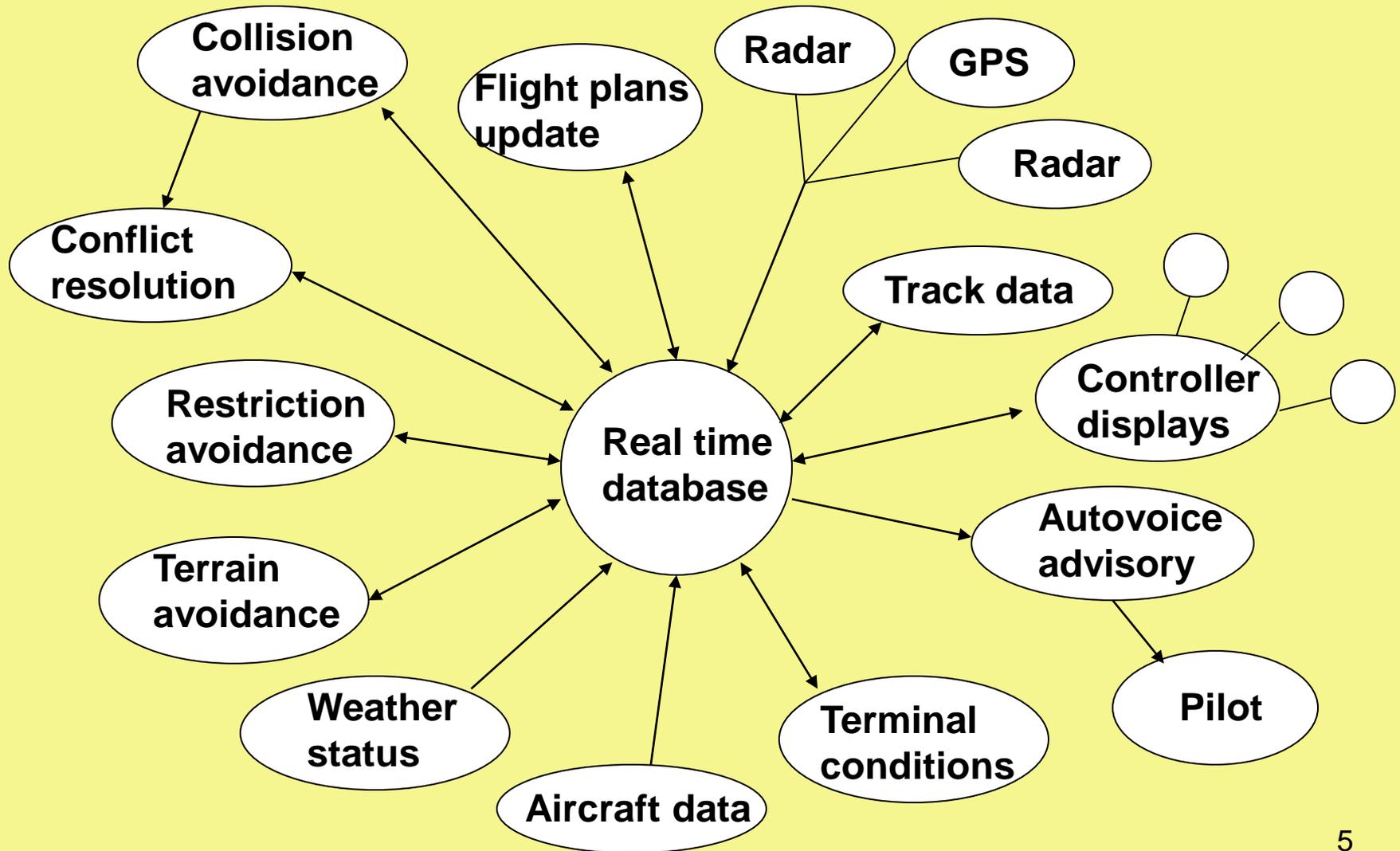
- WASHINGTON — “A government watchdog says new computers crucial to modernizing the U.S. air traffic control system have run into serious problems and may not be fully operational before the current computers are supposed to be replaced.”
- “Transportation Department Inspector General Calvin Scovel told a House committee on Wednesday that the \$2.1 billion computer system has misidentified aircraft and had trouble processing radar information.”

- Scovel stated “air traffic controllers in Salt Lake City where the system is being tested have also had difficulty transferring responsibility for planes to other controllers.”
- “Scovel warned that if the problems continue they could delay transition to an air traffic control system based on GPS technology instead of radar.”
- This is nothing new. These types of failures are typical for air traffic control.

Air Traffic Control Systems

- A real-time system that continually monitors, examines, and manages space conditions for thousands of flights by processing large volumes of rapidly changing data, due to reports by sensors, pilots, and controllers.
- Provides the best estimate of position, speed, and headings of every aircraft in the environment at all times.
- Consists of multiple real-time tasks, each of which must be completed before their individual deadline.
- Requires maintenance and interaction with an extremely dynamic database system.

Simplified ATC Real-Time Database



Conflict Detection & Resolution

- Free flight allow pilots to choose the best path to minimize fuel consumption and time delays.
- The most critical issue for free flight is CD&R, which is responsible for avoiding potential aircraft conflicts.
- CD&R is a time consuming and critical real-time task
- The Kalman filter is the central tracking algorithm for most CD&R algorithms
 - Does not predict well when aircraft make sudden turns, accelerations, etc.
 - Many of the algorithms consider only two aircraft and become inaccurate as the number of aircraft increases.
 - Not guaranteed to meet real-time deadline.

Assumed ATC Problem Size

Problem Size Per Region

- Controlled **IFR** flights 4,000
(instrument flight rules)
- Other flights 10,000
 - Uncontrolled **VFR** (visual flight rules) flights
 - IFR flights in adjacent sectors
- Total tracked flights 14,000
- Radar Reports each second 12,000

Total Regions

- 20 regions in contiguous USA plus one in Alaska and one in Hawaii.

Past ATC Implementation Difficulties

- All ATC software has repeatedly failed to meet the USA FAA specifications since 1963.
 - Central Computer Complex (CCC) in 1963.
 - Discrete Address Beacon System (DABS) or Intermittent Positive Control (IPC) in 1974–1983.
 - Automated ATC System (AAS) 1982-1994.
 - Standard Terminal Automation Replacement Systems (STARS) in 1994-?
 - **ADDED:** Current Problems in Salt Lake City

An Associative Processor for ATC

- An Associative Processor (AP) is a SIMD computer with a few additional associative features.
 - Associative properties are identified on the next slide.
- The associative features are supported in hardware
 - Used to enable rapid execution for dynamic database operations
- We assume the interconnection network supports at least the ring topology.
- Two associative architectures were built at Goodyear Aerospace during the 1970's and 1980's
 - STARAN – Chief architect was Kenneth Batcher
 - Built explicitly for Air Traffic Control.
 - ASPRO - A second generation STARAN.
 - Built for the Navy for related air defense systems.

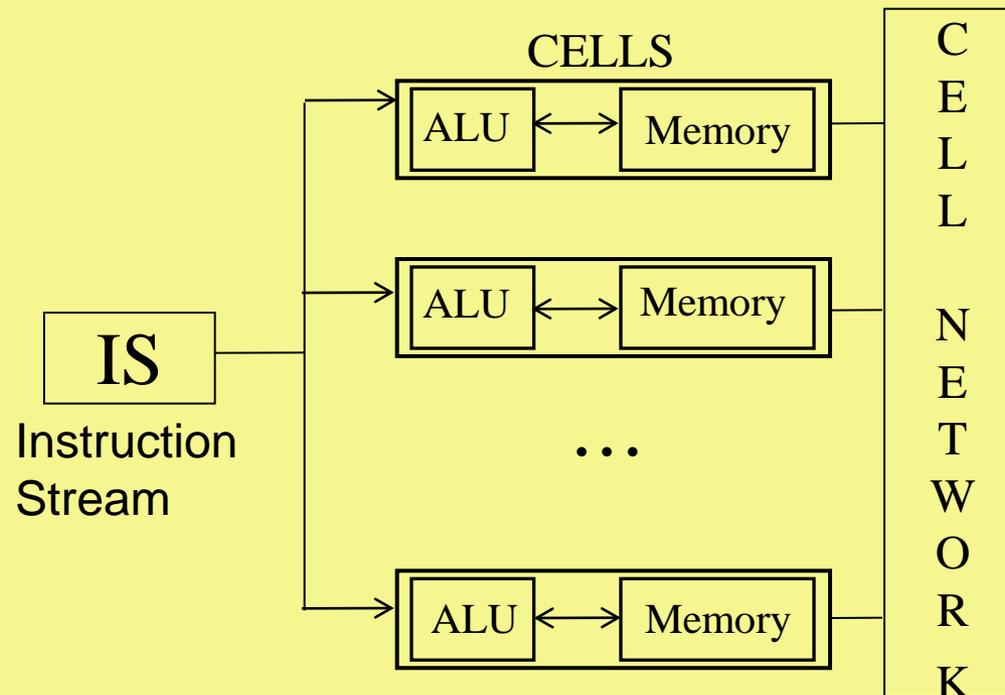
List of the Associative Properties

- Broadcast of data to all processors in constant time.
- Constant time global reduction of a parallel variable with
 - Boolean values using AND/OR.
 - Integer values using MAX/MIN.
- Ability to search for a data item in a parallel variable in constant time
 - Provides content addressable data.
 - Eliminates need for sorting and indexing.
- A constant time AnyResponders boolean function which identifies whether any parallel variable contains the data item used in the search.
- A constant time PickOne function which may be used if AnyResponders is true to return the location in a parallel variable that contains the data item.

- Above properties supported in hardware using a broadcast and a reduction network.
 - This can be one network, but is normally two.
 - Below reference provides proofs that above properties can supported in constant time.

Reference: M. Jin, J. Baker, and K. Batcher, Timings of Associative Operations on the MASC model, *Proc. of the Workshop of Massively Parallel Processing of IPDPS '01*, San Francisco, CA, April, 2001

The Associative Processor (AP)



Architectural examples include Goodyear Aerospace's
STARAN
USN ASPRO

Implementing ATC on an AP

- All records for each aircraft will be stored in a single processor.
 - Unnecessary movement of data between PEs wastes time.
- Assume initially each processor will store the records for at most one aircraft.
 - Reasonable, since the memory size and speed of processors in a large SIMD is typically small, due to cost restrictions.
- For ATC tasks, an AP with n processors can execute n instances of the same task in essentially the same time as it takes to execute 1 instance of this task.
 - This produces an optimal speedup $O(n)$ of roughly n .

- As long as there is no more than one aircraft per processor, the running time for the AP does not increase as the number of aircraft increase.
- Some argue that assigning a processor to at most one aircraft is inefficient,
 - Keeping the maximum number aircraft per processor very small is essential for real-time computing with short deadlines.
- If number of aircraft assigned to each processor increases from 1 to k
 - Running time will increase by at least a factor k .
 - The number of ATC tasks that can be executed during a major real-time cycle will decrease rapidly.
 - Processor memory size will restrict size of k .
- SIMD processors usually have a slower running time and small memories so that the cost of a large numbers of them is affordable.

The deterministic architecture of a SIMD will allow precise estimates of “worst case” running times.

- Partially due to deterministic movement of data on broadcast bus or interconnection network.
- Allows the use of static (instead of dynamic) scheduling.
- Avoids many time-consuming activities typical of MIMD implementations, primarily due to its single instruction stream
 - Dynamic scheduling, load balancing, indexing, linking, shared resource management, preemption, data locking, lock management, etc.
 - Assuring ACID properties of database transactions

Multiprocessor NP-hard Problems

- SIMDs are very different than multiprocessors
 - Illustrated by fact that most of the numerous, well-known “**NP-hard problems explicitly involving multiprocessors**” do not apply to SIMDs
 - Most proofs do not apply to SIMDs (or sequential computers) as they have only one instruction stream.
 - Exact or approximate software solutions to these type of problems are not needed as part of the solution of other problems.
 - Reference: M. Garey and D. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness*. W.H. Freeman, 65-66, 238-240, New York, 1979.

Is Massive Parallelism Useful for ATC?

- Earlier, 14,000 aircraft was indicated as the maximum number of assumed tracked flights in one region.
- Many professionals consider parallel systems with less than 100K processors as not being massively parallel.
- However, it is reasonable to believe that APs with 100K or more processors may be needed in ATC
 - See next slide.

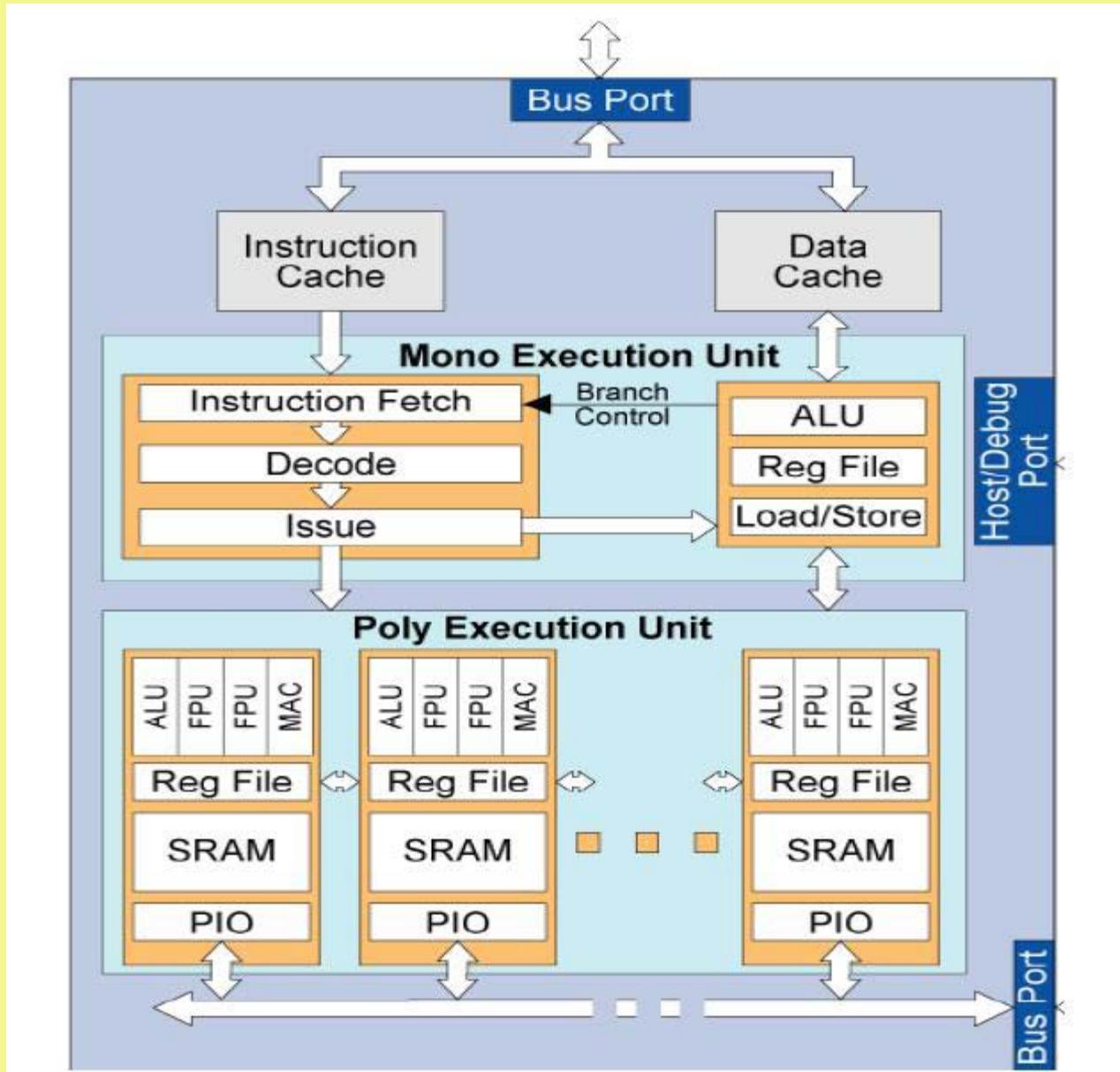
Reasons 100K Processors May be Needed for ATC

- As part of the current NextGen project, FAA wants to consolidate as many ATC activities as possible.
 - E.g., consolidate multiple regions to reduce the number of handoffs required for aircraft.
 - Backup computations for redundancy, e.g. for nearby regions
- Number of small aircraft is rapidly increasing
- Unmanned aerial vehicles (UAVs) and objects are increasing even more rapidly.
- Cars have recently been built that can also fly

CSX600 ClearSpeed Accelerator Board

- The CSX600 is a multi-core processor with
 - A PCI-X card equipped with 2 CSX600 coprocessors
 - Each coprocessor as 96PEs, connected with a swizzle (i.e., ring interconnection) network.
 - The multi-core section is called a multi-threaded array processor (MTAP), and is shown on next slide.
 - The PEs collectively have an aggregate bandwidth of 96 Gbytes (on-chip memory).
- Each PE has
 - 6 Kbytes of local memory
 - A clock speed of 250 MHz
 - Its own ALU

MTAP Architecture of CSX6000



Programming the CSX600 Board

- We are currently using only one of the two co-processors in order to obtain a more SIMD-like environment.
- At each step, all active PEs execute the same command synchronously on their individual data.
- The Cⁿ language is used on the ClearSpeed board is similar to standard C.
 - The main difference is that Cⁿ has two types of variables
 - The mono variable is equivalent to regular C variable and used by the control unit (or IS).
 - The poly variable are parallel variables and hold one value from each PE.

Emulating the AP on the CSX600

- The CSX600 coprocessor is SIMD, so only the associative functions need to be emulated efficiently
- We do not claim these can be supported in constant time
 - The following associative functions are available in assembler and have extremely fast implementations:
 - AND & OR reductions across a Boolean poly variable
 - Associative search across a poly variable.
 - AnyResponder (following an associative search)
 - The following associative functions also have fast implementations:
 - MAX & MIN reductions across a integer poly variable
 - PickOne (following a successful AnyResponder call)

Implementing Aircraft Tracking on ClearSpeed CSX600

1. All radar reports are transferred from the host to the mono memory. Next, the radar reports are transferred to the PE memories, with each PE receiving an equal share of the reports.
2. Boxes of sides of length 1nm are centered around each radar report and each track in each PE to accommodate report and track uncertainties.
3. Check intersection of each report box with every track box in each PE. If there is an intersection, the radar report and the track are correlated

Algorithm for Aircraft Tracking

4. The radar report in each PE are transferred to next PE using the swazzle (i.e., ring) network. After 96 iterations, all reports have been compared with all tracks.
5. Double the box sizes of tracks that have not correlated with any reports to increase their probability to intersect a report box and repeat the steps 3 & 4 above for unmatched reports
6. Triple the original box sizes of tracks that have not correlated and repeat the steps 3 & 4 above for unmatched reports.

Experimental Results: Goals

Requirements for the ATC Correlation Task:

1. The SIMD-based solution *scales well* with respect to the input size (i.e., number of planes/tracks)
2. Correlation performed every 0.5 second and consumes a largest amount of available time.
3. As a safety critical application, the correlation task offers a *predictable execution pattern*:
 - Offers tight upper bounds on the task's execution times. This is critical to enable real-time guarantees

Experimental Results: Methodology

- **Scalability:**

- Test 1: increase the number of planes from 4000 to 14000 in increments of 1000
- Take 50 samples for each instance
- We measured the maximum execution time over all samples

- **Predictability:**

- We measured the **coefficient of variation (COF)** which is a common normalized measure of dispersion, and is defined as the ration of the standard deviation to the mean
- Unlike the standard deviation, the COF is dimensionless

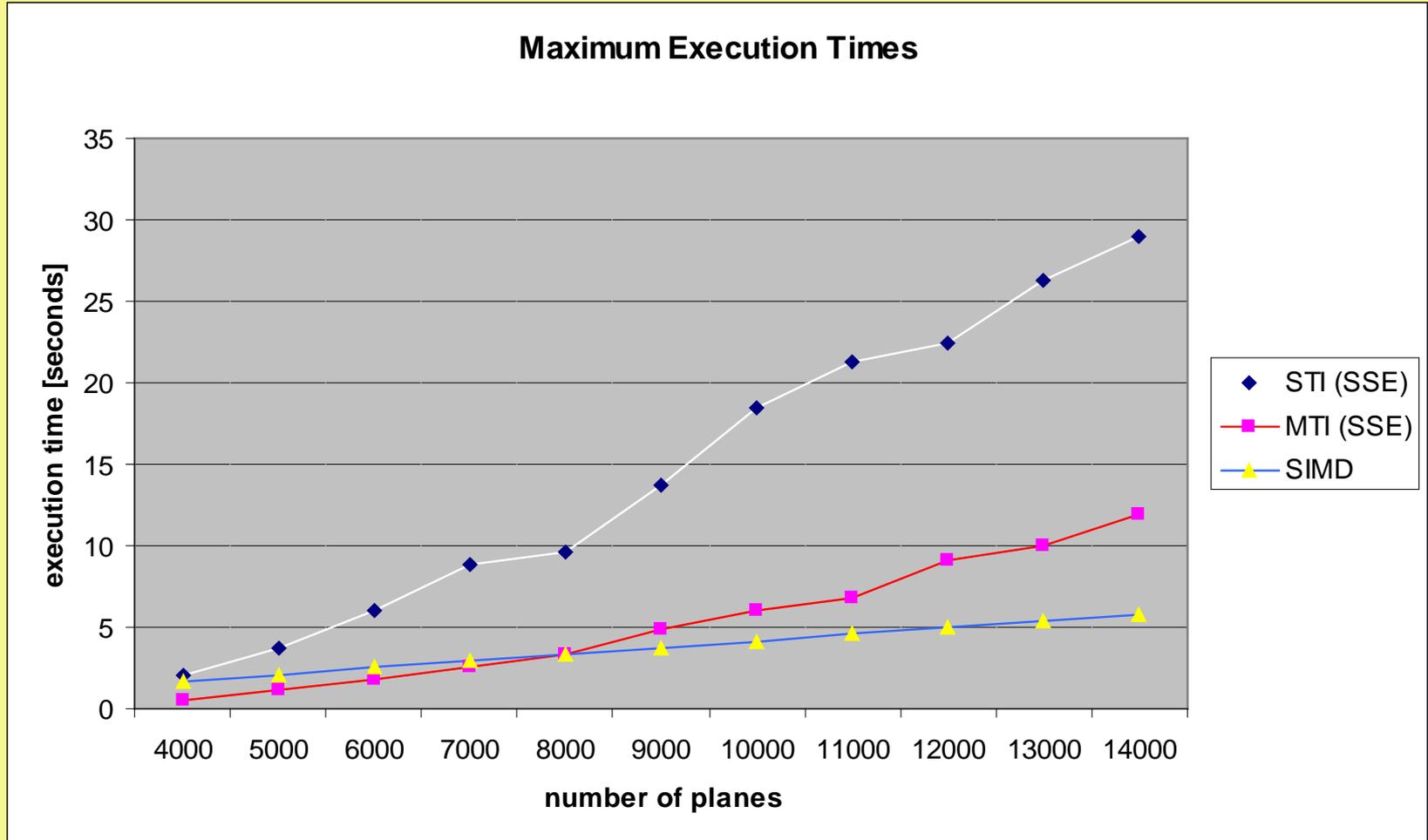
Experimental Setup

	SIMD	MIMD
Hardware	ClearSpeed CSX 600 96 PE's (only one of the two chips was used)	Dual Processor Xeon E5410 Quad Core 2.33 GHz system (total of 8 cores) with 32 GB of main memory and 6MB of L2 Cache for each CPU
Software	CSX600 SDK	Linux Kernel release 2.6.22 gcc compiler version 4.1.3 Intel Streaming SIMD extensions enabled

Information for Graphs

- SIMD will denote the CSX600 implementation
 - Considers only cases where processors contain records for a very large number of aircraft.
 - Best performance for SIMD is when there is at most one aircraft per processors.
- STI denotes a single threaded implementation, which is executed on a single core. (SSE)
- MTI denotes a multi-threaded implementation, based on POSIX Pthreads
 - Implementation was carefully designed to minimize typical performance limiting effects such as false sharing, cache-ping-pong, and high lock contention
 - The multi-threads were specifically designed to avoid locking whenever possible.
- Intel's streaming SIMD Extensions (SSE) were enabled for both STI and MTI

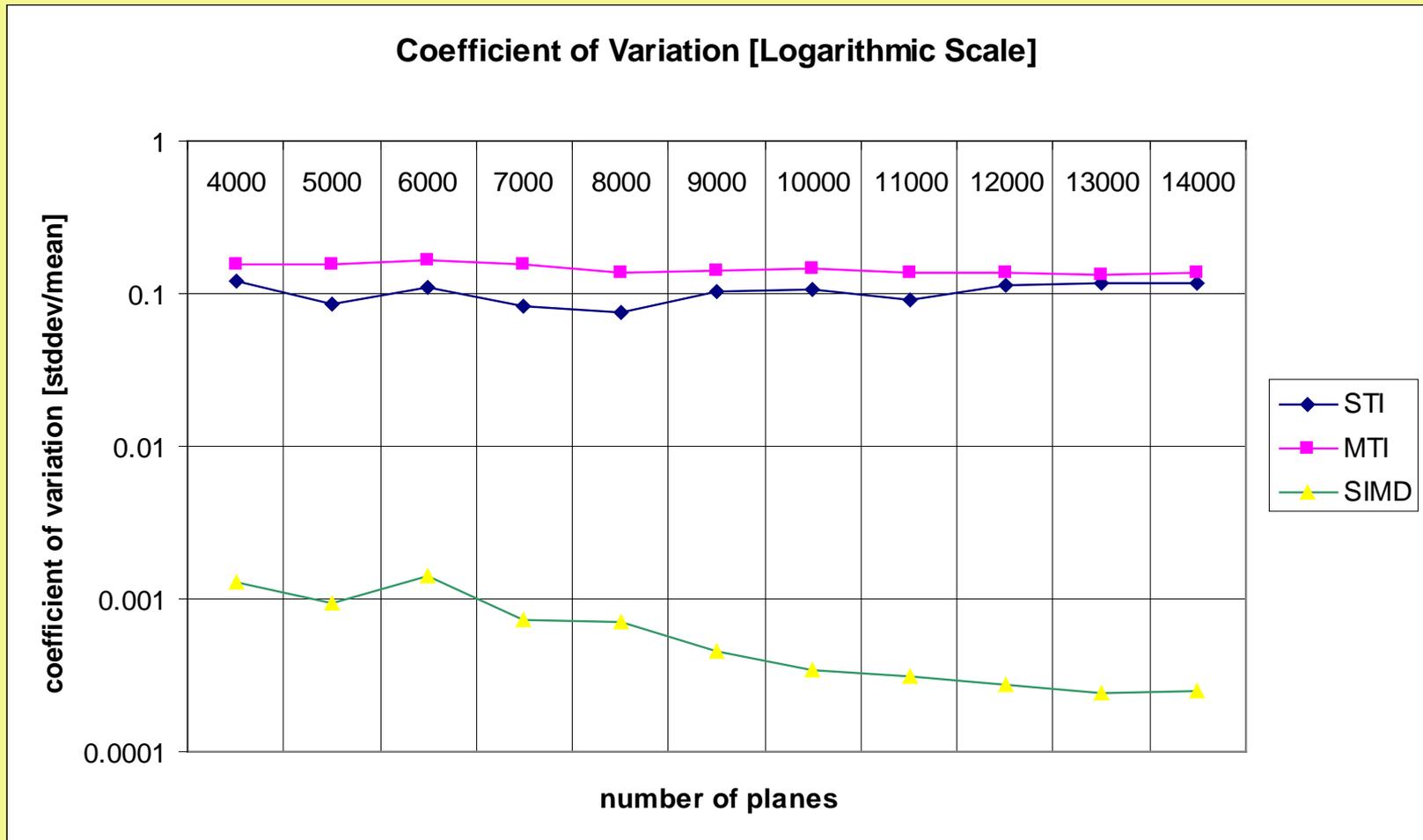
Results: Scalability



Scalability Results

- Shows the maximum value of the execution times for each of the experiments for the three approaches
- STI always takes the most time and increases the quickest
- MTI compared to SIMD
 - Takes less time for 4000-7000 aircraft
 - has similar time from 7000-8000 aircraft
 - takes more time and increases faster from 8000 on.
- SIMD displays a linear time per aircraft.
 - 1632 tracks are processed at the same time
 - 9 iterations needed in worst case
 - Most time for computation, little for data transfers

Results: Timing and Predictability



Timing & Predictability Results

- An important factor guaranteeing that all ATC can be performed within time bounds.
- Coefficients of Variation (COV) is a common, normalized measure of dispersion.
- Note the y-axis uses a logarithmic scale
- Results clearly show that the COV values for SIMD are several magnitudes below the ones for STI and MTI.
- Fluctuations in execution times of both STI and MTI are, in part, due to operating system and hardware interference

Summary

- The goal of this paper was to demonstrate the feasibility of handling air traffic control using an associative processor.
- Multiple advantages of an AP over a MIMD for ATC have been discussed
- An emulation for the AP on one of the two chips with 96 processors in the ClearSpeed CSX600 series was implemented.
- Three algorithms for ATC (tracking & correlation, conflict detection, and conflict resolution) were implemented on the CSX600.

- The CSX600 can meet deadlines for these 3 tasks only up to a max of 17 aircraft per processor or 1500 aircraft.
 - More processor needed for additional aircraft.
- An ideal AP should have a minimum of 14K PEs.
 - AP processors of 100K and larger should be useful for ATC as current operations are combined and more redundancy added.
- The runtime for an AP with at most 1 aircraft per processor should not increase significantly as the number of aircraft increases up to max nr of PEs
 - Experimental tests needed to check validity of this claim on CSX600

- Other advantages of using an AP for ATC
 - Worst case running time can be accurately predicted by an AP
 - Our experiment showed that the variations in running time for the CSX600 is very small in comparison to MIMD.
 - In contrast, MIMD systems optimize average case running time and have highly unpredictable worst case running time.
 - Software used by the AP is substantially simpler and smaller in size
 - The Validation and Verification (V&V) is much simpler than for current MIMD software.
 - The hardware architecture of the AP is much simpler than current hardware.

Future Work

- Obtain additional timings on current implementations, e.g.,
 - Rate of increase of run-time on ClearSpeed with at most one aircraft per processor as # of aircraft increase 1-96 to see if this substantiates claims of being nearly constant.
 - Rate of increase of run-time on ClearSpeed as maximum number k of planes per processor increase.
- Implement the current 3 tasks on ClearSpeed also on STI & MTI (MIMD models) and get more comparative timings.
- Complete the implementation of the basic ATC tasks (about 8) on ClearSpeed CSX600. Then implement on MIMD systems of similar power and compare efficiency and predictability.
- Possibly implement the basic ATC tasks on Nvidia's new FERMI chip.
 - Has a lot in common with the MTAP approach of ClearSpeed.

REFERENCE SLIDES

ASPRO Predictability - circa 1979

Simulated environment 4,000 Reports - 2,000 Tracks

Routine	Instruction count	Time in milliseconds/scan	
		Predicted	Measured
Association pairing	415	*	640.0
Compare and sort	1012	*	14.0
Correlation	788	22.16	4.5
Tentative Track	555	16.68	12.5
Track Update	661	14.84	8.9
Hghtup	407	2.68	2.9
Range Prediction	640	37.04	24.77
Association gates	443	9.12	8.0
Kalman Tracking	1026	46.64	39.2
Track Quality	209	7.28	5.06
Air/Surface	326	*	0.66
Establish Track	407	0.88	0.71
Final Bookkeeping	243	15.98	6.6

Totals	7132		767.8 msec

* not predicted

113.14 msec for ATC tracking

(The L304 Processor took 212 seconds for same jobs with 10 second limit off)

Coefficient of Variation

- In probability theory and statistics, the **coefficient of variation (CV)** is a normalized measure of dispersion of a probability distribution. **It is defined as the ratio of the standard deviation to the mean.**
- The standard deviation of data must always be understood in the context of the mean of the data. So when comparing between data sets with different units or widely different means, one should use the coefficient of variation for comparison instead of the standard deviation.