The Air Traffic Control Real Time System:
SIMD vs MIMD Solutions

Johnnie W. Baker*
Mingxian Jin*
Will C. Meilander*

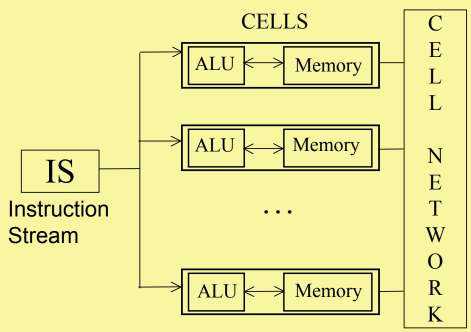
Kent State University

(* - *names in alphabetic order*)

Outline

- The ASC model or Associative Processor (**AP**)
  - An enhanced SIMD model
- Overview of a AP solution for the Air Traffic Control problem (**ATC**)
- Difficulties with ATC solutions using multiprocessors (**MP** or **MIMD**)
- Conclusions

The ASC (Associative Computing) Model



Architectural examples include Goodyear Aerospace's
STARAN
USN ASPRO

Associative Properties

- Broadcast data in constant time.
- Constant time global reduction of
  - Boolean values using AND/OR.
  - Integer values using MAX/MIN.
- Constant time data search
  - Provides content addressable data.
  - Eliminates need for sorting and indexing.
- Above properties supported in hardware with broadcast and reduction networks.

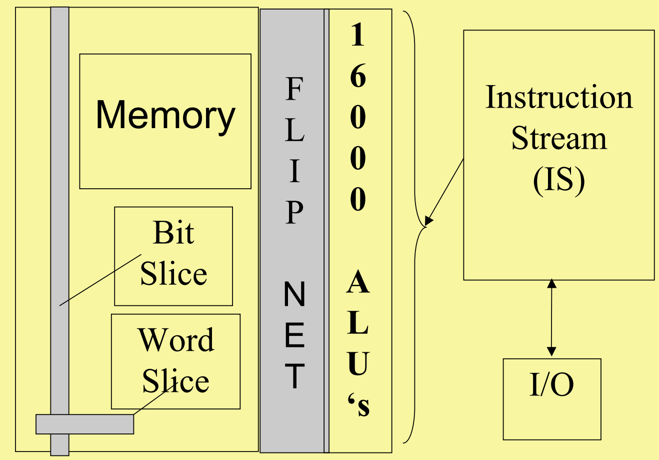
Reference: M. Jin, J. Baker, and K. Batcher, Timings of Associative Operations on the MASC model, *Proc. of the Workshop of Massively Parallel Processing of IPDPS '01,* San Francisco, CA, April, 2001, (Unofficial version at http://www.cs.kent.edu/~parallel/papers).

IS

	Make	Color	Year	Model	Price	On lot		Busy-idle
PE1	Dodge	red	1999			1		1
PE2						0		0
PE3	Ford	blue	1998			1		1
PE4	Ford	white	2001			0		1
PE5						0		0
PE6						0		0
PE7	Subaru	red	2000			1		1

The Associative Search

Possible AP for ATC (Air Traffic Control)

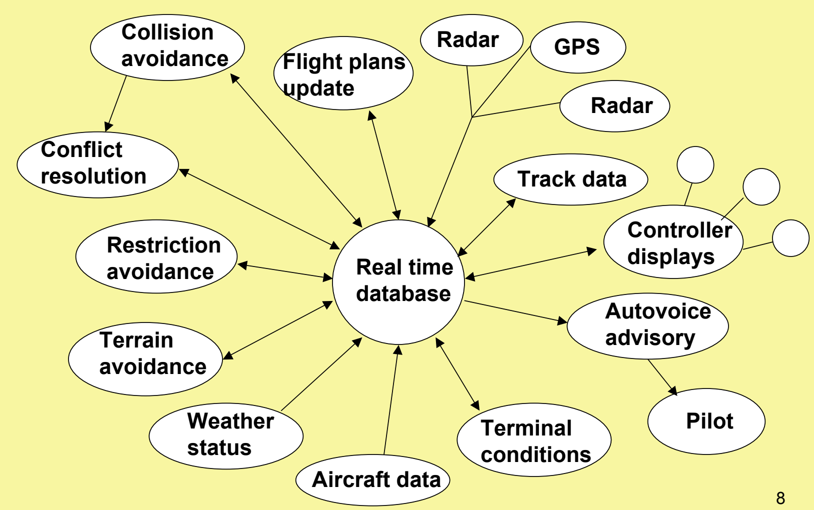


An AP Solution for the ATC Problem (Overview)

Basic Assumptions:

- Data for this problem will be stored in a real time database
- SIMD supports a relational database in its natural tabular structure, as first presented by E. F. Codd in 1970.
- The data for each plane will be stored together in a record, with at most one record per PE.
- Other large sets of records (e.g., radar) will also be stored in PEs with at most one per PE.

ATC Real-Time Database



ATC Worst-Case Environment

- Controlled IFR flights 4,000 (= ***n***)
  (instrument flight rules)
- Other flights 10,000 (= ***m***)
  - Uncontrolled VFR (visual flight rules) flights
  - IFR flights in adjacent sectors
- Total tracked flights 14,000
- Radar Reports per Second 12,000

ATC Example -- Conflict Detection

- A conflict occurs when aircraft are within 3 miles or 2,000 feet in altitude of each other.
- A test is made every 8 seconds for a possible future conflict within a 20 minute period
- Each flight's estimated future positions are computed as a space envelope into future time.
- An intersection of all pairs of envelopes must be computed.

Associative Processor Jobsets

- The AP compares each of the 4,000 controlled flights with all remaining (13,999) flights in constant time.
  - First, the envelope data for a controlled flight is broadcast
  - The PE for each of the other flights simultaneously check if this envelope intersects the envelope for their aircraft.
- Since the comparison in each PE corresponds to a ***job***, we call this AP set operation a ***jobset***.
- The ATC Conflict Detection algorithm for the AP requires 4,000 jobsets

MIMD Algorithm for Conflict Detection

- Typically, MIMDs treat each envelope comparison as a separate job.
  - There are 13,999 jobs per controlled flight.
  - This approach requires a total of roughly 56 million jobs.
- Recall the AP algorithm required 3,999 jobsets.
  - Each AP jobset required constant time.
- The AP algorithm is $O(n)$.
- Above MP algorithm is $O(n(n+m))$ or $\Omega(n^2)$

A polynomial AP solution for ATC (overview)

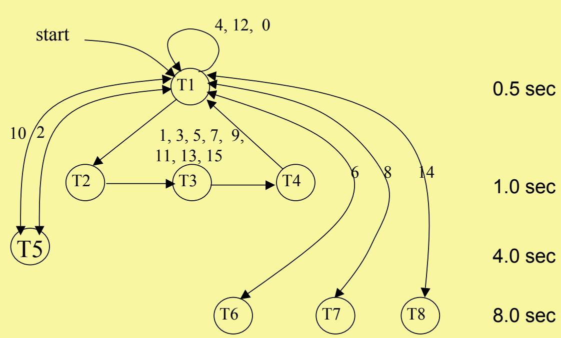
- A static table-driven scheduler is designed.
- The resulting fixed schedule of tasks allow ample time for worst-case executions of tasks to met their deadlines.
- Periodic tasks or jobsets run at their release times and each is completed by its deadline.
- A special task handles all the aperiodic or sporadic jobsets that have arrived within the last period.
- The execution time for each task is low-order polynomial

Static Scheduling Key ATC Jobsets

	Task	period	Proc time
1.	Report Correlation & Tracking	.5	1.44
2.	Cockpit Display 750 /sec)	1.0	.72
3.	Controller Display Update (7500/sec)	1.0	.72
4.	Aperiodic Requests (200 /sec)	1.0	.4
5.	Automatic Voice Advisory (600 /sec)	4.0	.36
6.	Terrain Avoidance	8.0	.32
7.	Conflict Detection & Resolution	8.0	.36
8.	Final Approach (100 runways)	8.0	.2

Summation of Task Times in an 8 second period 4.52

AP solution for ATC (3)



AP ATC schedule

Static Schedule for ATC Tasks

	.5 sec	1 sec	4 sec	8 sec
1	T1	T2, T3, T4		
2	T1		T5	
3	T1	T2, T3, T4		
4	T1			
5	T1	T2, T3, T4		
6	T1			T6
7	T1	T2, T3, T4		
8	T1			T7
9	T1	T2, T3, T4		
10	T1		T5	
11	T1	T2, T3, T4		
12	T1			
13	T1	T2, T3, T4		
14	T1			T8
15	T1	T2, T3, T4		
16	T1			

Demo of AP Solution

- A demo of this hardware-software ATC system prototype was given for FAA at a Knoxville terminal in 1971 by Goodyear Aerospace:
  - Automatic track
  - Conflict detection
  - Conflict resolution
  - Terrain avoidance
  - Automatic voice advisory for pilots
- The 1971 AP demo provided ATC capabilities that are still not possible with current systems

ATC Reference: Meilander, Jin, Baker, Tractable Real-Time Control Automation, *Proc. of the 14th IASTED Intl Conf. on Parallel and Distributed Systems (PDCS 2002)*, pp. 483-488. (Unofficial version at http://www.cs.kent.edu/~parallel/papers)

Possible MP Solutions to ATC

1. A typical MP approach using dynamic scheduling for ATC tasks
   - ➢ Dynamic scheduling is NP-hard
2. MP simulation of the AP solution
   - ➢ Executes in SIMD fashion
   - ➢ Uses static scheduling
3. Perhaps other MP solutions

ATC can be Represented as a Relational Database

- SIMD is the only architecture that can implement a relational database in a tabular structure, as first presented by E. F. Codd in 1970.
  - ➢ There is no specific order required in rows or columns.
- Implementing the same database in the MP is a very difficult task, and may be a contributing factor for failure of the MP system to manage ATC data adequately.
- In either case serializability of jobs is essential in order to maintain a coherent database

General MP Problems for ATC Software (Avoided by AP solution)

- Each PE will contain a large number of records
  - ➢ e.g., There are 14K records just for planes
- If multiple database records of the same type (e.g., plane records) are stored in a single PE, these records be processed sequentially.
- A distributed database must be supported so as to
  - ➢ Assure data serializability
  - ➢ Maintain data integrity
  - ➢ Manage concurrency
  - ➢ Manage data locking
- One or more dynamic task scheduling algorithms are needed
  - ➢ Normally dynamic scheduling is used to schedule ATC tasks
  - ➢ Data base maintenance activities must also be scheduled

General MP Problems for ATC Software (Avoided by AP Solution – cont.)

- Synchronization
- Load balancing between processors
- Complex data communication between processors
- Maintaining multiple sorted lists and indexes required for fast location of data
- Most MP solutions for ATC tasks have higher complexity (by a factor of *n*) than corresponding AP solution.
  - Details on next slide

A typical MIMD Approach

- This is the approach that has been used since early 1980's.
- Uses dynamic scheduling of tasks
  - An NP-complete problem
- Data is stored in a dynamic database,
  - Many records per PE
- Multiple sorting and indexing is needed to locate data
- Complex data movements used
- Has repeatedly failed to meet FAA specifications since 1963.

Difficult for MP to simulate the AP solution

- The AP solution of the ATC problem are saturated with the use of AP constant-time operations
  - broadcast, AND/OR and MAX/MIN reductions, associative search, responder processing
  - Hardware support (e.g., reduction circuits) is required in AP for constant-time operations
  - Software support for these in MP would be at least Ω (log n) and likely higher
- A significant slowdown in the MP simulation of the AP constant time operations is almost certain to result in missed deadlines in the ATC static schedule.
- The other MP problems mentioned earlier will also be present.

Comparison of some required ATC operations

(Excluding MP data management overhead software)

Operation	**MP**	**AP**
Report to track correlation	$O(n^2)$	$O(n)$
Track, smooth and predict	$O(n)$	$O(1)$
Flight plan update and conformance	$O(n)$	$O(1)$
Conflict detection	$O(n^2)$	$O(n)$
Conflict resolution	$O(n^2)$	$O(n)$
Terrain avoidance	$O(n^2)$	$O(n)$
VFR automatic voice advisory	$O(n^2)$	$O(n)$
Cockpit situation display	$O(n^2)$	$O(n)$
Coordinate transform	$O(n)$	$O(1)$

Does a Polynomial Time MP Algorithm Exist?

- Large teams of experts have worked on MP solutions to the U.S.A. version of the ATC problem almost continually for over 40 years.
- Many highly respected companies (e.g. IBM, Mitre, TRW, Lockheed, etc.) have participated in these efforts.
- All ATC software has repeatedly failed to meet the U.S.A. FAA specifications since 1963.
  - CCC in 1963, DABS/IPC in 1974, AAS in 1983, STARS in 1994
- If a polynomial time MP solution is possible using <u>current computers</u>, it seems likely that it would have been discovered.
- It is generally believed that a polynomial time solution is impossible.

Real-Time Multiprocessor Scheduling Complexity

- John Stankovic..; "…complexity results show that most real-time multiprocessing scheduling is NP-hard."
- Mark Klein…; "…most realistic problems incorporating practical issues … are NP-hard."
- Garey, Graham and Johnson
  - "…all but a few schedule optimization problems are considered insoluble… For these scheduling problems, no efficient optimization algorithm has been found, and indeed, none is expected."

Real-Time Multiprocessor Scheduling (continued)

- A useful theorem in Gary and Johnson's classic book, *Computers and Intractability: A Guide to the Theory of NP-completeness (See* SS8 on pp.65 and pp. 238-240)
  - Let T be a nonempty set of tasks.
  - Let the length $l(t)$ of each task $t \in T$ and the deadline D be positive integers
  - The problem of whether there is a schedule for a multiprocessor with m processors for T that meets the overall deadline D is NP-complete for $m \geq 2$, assuming not all tasks have the same length.

Power of SIMD vs MIMD

1. It is generally believed that ***<u>MIMDs can efficiently simulate SIMDs and are more general and more powerful.</u>***
2. However, for the ATC problem:
   - A simple low-order polynomial time AP solution can be obtained
   - a polynomial time MP solution is not known and none is expected
3. As a result, it very likely that current ***<u>MIMDs cannot efficiently simulate SIMDs on some very important applications.</u>***
4. APs may also have important applications to the following general classes of problems containing ATC
   - Real time systems with hard deadlines.
   - Command and control problems

Conclusions

- A simple polynomial-time algorithm has been described for the ATC using a AP.
- A polynomial time ATC algorithm for the MP is currently not expected.
- Polynomial time algorithms should also be possible for perhaps many problems other "Command and Control" problems.
- Modern architectures and applications for APs and SIMDs should be reconsidered

WEBSITE

http://www.cs.kent.edu/~parallel

Follow the pointer to "papers"

References

1. 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. (Unofficial version at http://vlsi.mcs.kent.edu/~parallel/papers)
2. Meilander, Jin, Baker, Tractable Real-Time Control Automation, *Proc. of the 14th IASTED Intl Conf. on Parallel and Distributed Systems (PDCS 2002)*, pp. 483-488. (Unofficial version at http://vlsi.mcs.kent.edu/~parallel/papers)
3. J. A. Stankovic, M. Spuri, K. Ramamritham and G. C. Buttazzo, *Deadline Scheduling for Real-time Systems*, Kluwer Academic Publishers, 1998.
4. M. R. Garey and D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-completeness*, W.H. Freeman, New York, 1979, pp.65, pp. 238-240.

THE END

Additional PDCS-02 Slides

"Tractable Real-Time Air Traffic Control Automation"

Will Meilander, Mingxian Jin, Johnnie Baker

Kent State University

Selected Slides to from Will Meilander's presentation at PDCS'02 on "Tractable Real-Time Control Automation"

Predictability

Mark H. Klein et al, Carnegie Mellon Univ. Computer, Jan. '94 pg 24

"One guiding principle in real-time system resource management is *predictability.* The ability to determine for a given set of tasks whether the system will be able to meet all the timing requirements of those tasks."

ATC Fundamental Needs

- The best estimate of position, speed and heading of **<u>every</u>** aircraft in the environment **<u>at all times.</u>**
- To satisfy the informational needs of **<u>all</u>** airline, commercial and general aviation users.
- Some of these needs are:
  - Conflict detection and alert
  - Conflict resolution
  - Terrain avoidance
  - Automatic VFR voice advisory
  - Free flight
  - Final approach spacing
  - Cockpit display

Some ATC Facilities

Air Route Traffic Control Centers - 20

Terminal Radar Control systems - 186

Air Traffic Control Towers - 300

The first two facility types are supplied with radar data from about 630 radar systems.

ATC Automation Today

ATC implementations have been demonstrating since 1963 the complexity results in Real-Time scheduling theory. --

- Central Computer Complex (63 -)
- Discrete Address Beacon System/Intermittent Positive Control (74 - 83),
- Automated ATC System (82 - 94),
- Standard Terminal Automation Replacement System (94 -)

Jobset

We define a new term to describe the performance of an associative processor in real-time scheduling.

- In a MP implementation of a real-time database, a *job* is defined to be an instance of a task.
- In an AP, multiple instances of the same job are done simultaneously, with the same instructions being executed by all active PEs.
- This collection of all instances of the same jobs will be called a *jobset*.
- Observe this utilizes the AP as a set processor.

Flight ID	AC type	ETA	Destination	Controller #	Busy	
UAL 147	747	1100	CLE	17	1	PE
					0	PE
NW 1186	767	1107	ORD	26	1	.
					0	.
KLM 761	747	1105	CLE	8	1	.
AA 2345	A320	1135	ORD	17	1	.
					0	.
UAL 258	737	1112	CLE	9	1	.
AA 2744	737	1105	CAK	11	0	.
					0	
SW 377	767	1108	DET	8	1	PE
					0	.
						.

.
.
.
.
.
PE

<u>Example of a Jobset</u>

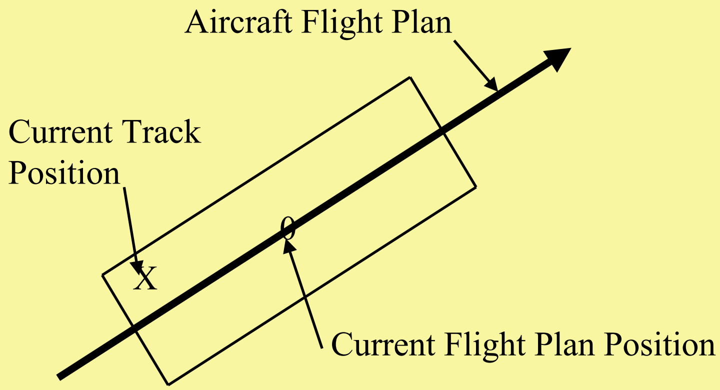
Find AC type where Busy = 1

And ETA is Between 1105 and 1110

And destination is CLE

Output AC type

A Second Jobset Example



Flight Plan Conformance Evaluation

Operations	Memory Accesses	Time (μs)
1 Identify all FP	2	
2 Get Xf, Yf, Hf, xd, yd, hd	116	
3 Xf_1 = Xf +xd, Yf_1 =Yf+yd, Hf_1 = Hf + hd	6	0.06
4 Get Xt, Yt, Ht, sin(hdg), cos(hdg)	116	
Calculate displacement of track from FP		
5 X'=(Xt-Xf)*cos(hdg)+(Yt-Yf)*sin(hdg)	8	0.12
6 Y'=(Yt-Yf)*cos(hdg)-(Xt-Xf)*sin(hdg)	8	0.12
7 Check X'>K1, if true set alert flag	3	0.06
8 Check Ht – Hf > K2, if true set alert flag	4	0.06
9 Check Y'>K2, if true, update FP	4	0.08
10 Store Xf_1, Yf_1, Hf_1	75	
Total	342	0.50

Total proc. time with 20 ns memory access time 7.34 μs

Table 1 Flight Plan Conformance Processing

AP Installations

The first installation of an AP by Goodyear Aerospace took place in the Knoxville terminal in 1969.

- It provided automatic radar tracking, conflict detection, conflict resolution, terrain avoidance, and display processing.

A 1972 STARAN demonstration by Goodyear Aerospace showed a capability to simulate and process 7,500 aircraft tracks performing the functions listed above.

A military version of the STARAN, called ASPRO, was developed and delivered in 1983 to the USN for their airborne early command and control system.

- Among other things it showed, as predicted, a capability to track 2000 primary radar targets in less than 0.8 seconds.

Goodyear Aerospace STARAN ATC Demonstration 1972

Full simulation of 7500 tracks per scan

SEQUENTIAL PROGRAMS		ASSOCIATIVE PROGRAMS	
NAME	COUNT		COUNT
Executive		Tracking System	881
Keyboard Interrupt		Flight Plan Simulation	415
Real Time Interrupt		Turn Detection	88
Live Data Input		Conflict Prediction	488
Automatic Voice Output		Conflict Resolution	296
		Automatic Voice Advisory	709
		Display Processing	1140
		Total	4017
		Field Definition Statements	-514
Net Operating Instructions	1600	Net Operating Instructions	3493

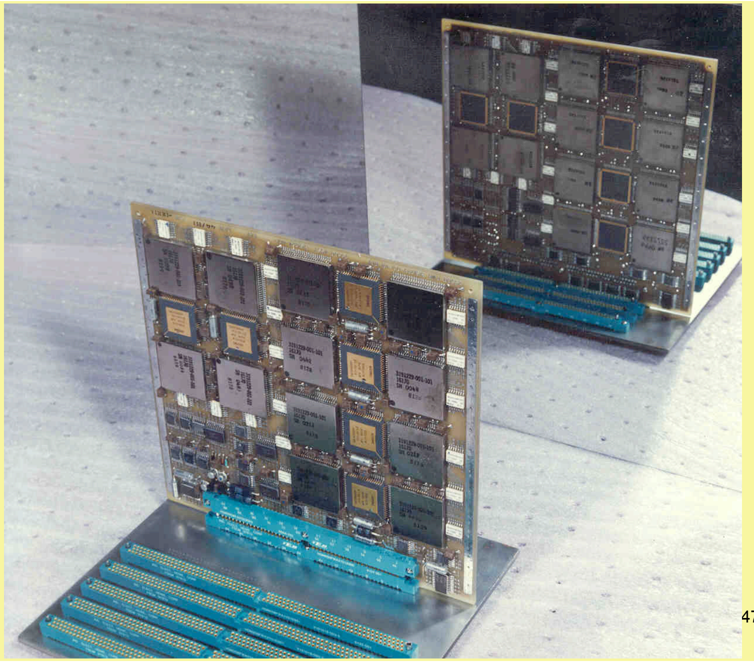
Limitations for Previous ATC Systems

- There is a fundamental flaw with all past and current ATC systems:
  - That flaw is the limited memory to processor bandwidth
  - Essentially **the von Neumann bottleneck**.
- Data cannot be processed faster than it can be moved between processor and memory
  - The limited bandwidth necessitates a multi-processor (MP) system.
  - The MP control overhead adds new problems that are intractable to the original ATC problem.
  - This is the direct cause of the system's inability to handle ATC processing needs.

USN ASPRO

1977 Initial design
1983 Delivery to the fleet
Characteristics
~2000 ALUs 500 bytes of 450 ns. memory per ALU
Space < 0.5 cu ft. including power supply and battery backup
< 250 watts power

Performance
276 Times improvement in track throughput



AP in Real-Time Air Traffic Control

The AP single thread instruction stream does not permit:

1. Shared resource conflicts!
2. Priority inversion problems!
3. Precedence constraint difficulties!
4. Preemption difficulties!
5. Processor assignment scheduling problems!
6. Data distribution problems!
7. Table, row or data element locks and lock management problems!
8. Concurrency difficulties!
9. Serializability problems!
10. Process synchronization problems!
11. Dynamic scheduling problems!
12. Memory and cache coherency management difficulties!

Two properties favor the AP.

1. The amount of physical AP hardware to do the ATC job is about 20% of that required for the best (inadequate) MP approach.

2. The amount of AP software is about 20% of that required for the best MP approach.

Ockham's Razor: "…entities must not be unnecessarily multiplied"


Air Force One radar glitch is typical
• Crowded skies cause planes to briefly 'disappear'
sweep takes approximately 12 seconds to complete.

This is not a radar problem.

The data from several radars that would have continuously supported the track was discarded.

The real problem: the multiprocessor is unable to process the radar data.

Given a Functional ATC Requirement

All ATC tasks are polynomial and scheduling them in a conventional processor is also polynomial

But, scheduling ATC tasks using a MP is generally believed to require solving new NP-hard problems

The AP static schedule for ATC avoids multithreading and thus can provide a polynomial time solution

Designing a Predictable Schedule for an AP

1. Find the time to execute each ***jobset*** in each task required by the system. (This is equivalent to the execution time for a job in an MP or uniprocessor.)
2. Then the time for each task is the sum over the worst case set of jobsets of the time for each jobset in the task.
3. Multiply the time for each task and the number of repeats of each task within the system deadline time.
4. Sum the resulting times for all the tasks. If this sum is less than the system deadline time a static schedule can be defined.

Conclusion

Using the AP, a polynomial time solution can be given to the ATC automation problem

This solution is simple and provides a realistic way to met the requirements of the USA ATC system

The AP is expected to be useful in providing efficient solutions to many other real-time database management problems