Accelerating Sequential Monte Carlo Method for Real-time Air Traffic Management

Thomas C.P. Chau,
James Targett, Marlon Wijeyasinghe,
Wayne Luk, Peter Y.K. Cheung
Imperial College London
Benjamin Cope
Altera Europe
Alison Eele, Jan Maciejowski
University of Cambridge

13 July, 2013

Overview

Acceleration of the Sequential Monte Carlo method

- Application to air traffic management

Accelerator design

- Novel particle stream structure
  - Evaluating constraints and weights rapidly
- Separate control from data path to promote scalability

Overview

Speedup

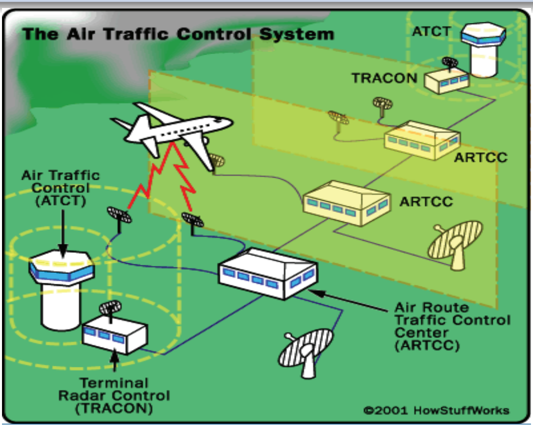
- Altera Stratix V 5SGSD8 FPGA at 150MHz
- 35 times faster, 133 times more energy efficient than a CPU
- 2.3 times faster, 13.5 times more energy efficient than a GPU

First to meet real-time requirement

- 4 aircraft in 2000s [IFAC 2011]
- 4 aircraft in 3.9s, extensible to 45 aircraft in 30s [This work]

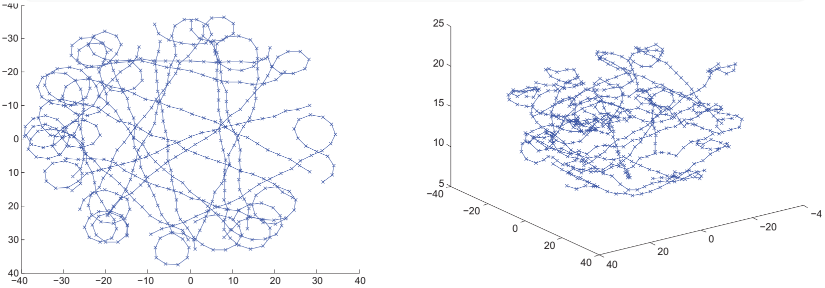
Objectives of air traffic management

- Schedule - Route aircraft in airspace
- Safety - Maintain separation between aircraft
- Efficiency - Minimise fuel usage and time of arrival



Difficulties of air traffic management

- Complex
  - Constrained: speed, altitudes, safety
  - Multi-aircraft: non-linear, non-convex
  - Uncertainty: wind, human and control error
- Slow
  - Largely performed manually



Air traffic management of 20 aircraft

Eele A. and Maciejowski J., Comparison of Stochastic Methods for Control in Air Traffic Management, IFAC World Congress, 2011

Difficulties of air traffic management

- Complex
  - Constrained: speed, altitudes, safety
  - Multi-aircraft: non-linear, non-convex
  - Uncertainty: wind, human and control error
- Slow
  - Largely performed manually

Current situation

- Conservative...
  - Lower capacity, longer time, more fuel
  - Worse passenger experience
- Previous attempts for automation...☹
  - Too slow for real-time practice
  - 4 aircraft in 2000s [IFAC 2011]

Sequential Monte Carlo method

- Sequential Monte Carlo / particle filter / survival of the fittest
- Input - a sequence of noisy measurements
- Filter - estimate the unknown states of a system that changes over time
- Example applications - air traffic management, city traffic control, robot localisation

Steps

1. Sampling - draw set of random **particles**
2. Importance - associate a **weight** to each particle
3. Resampling - reduce degeneracy of the particles
4. Simulation - compute estimation based on the samples + weights

Sequential Monte Carlo & air traffic management

Variables and parameters

- State S
- Control C
- Weight W
- Horizon H ([t, t+H])
- Number of particles N_P
- Number of aircraft A

Sequential Monte Carlo & air traffic management

Trajectory planning

$$\begin{pmatrix} x' \\ y' \\ a' \\ V' \\ \chi' \\ M' \end{pmatrix} = \begin{pmatrix} x + \delta t V \cos(\chi') \cos(\tau) \\ y + \delta t V \sin(\chi') \cos(\tau) \\ a + \delta t V \sin(\tau) \\ \chi + \delta t L \sin(\phi)/(MV) \\ V + \delta t(\frac{T}{M} - \frac{D}{M} - g \sin(\tau)) \\ M - \eta \delta t T \end{pmatrix} \tag{1}$$

Score calculation

$$J(k) = \alpha_{distance} J_{distance}(k) + \alpha_{fuel} J_{fuel}(k) + \alpha_{altitude} J_{altitude}(k) \tag{2}$$

$$J_{distance}(k) = \Big(\sqrt{(x_0 - G_x)^2 + (y_0 - G_y)^2} + k\delta t V_{max} - \sqrt{(x - G_x)^2 + (y - G_y)^2 + (a - G_a)^2)}\Big)/(k+1) \tag{3}$$

Sequential Monte Carlo & air traffic management

Constraint handling

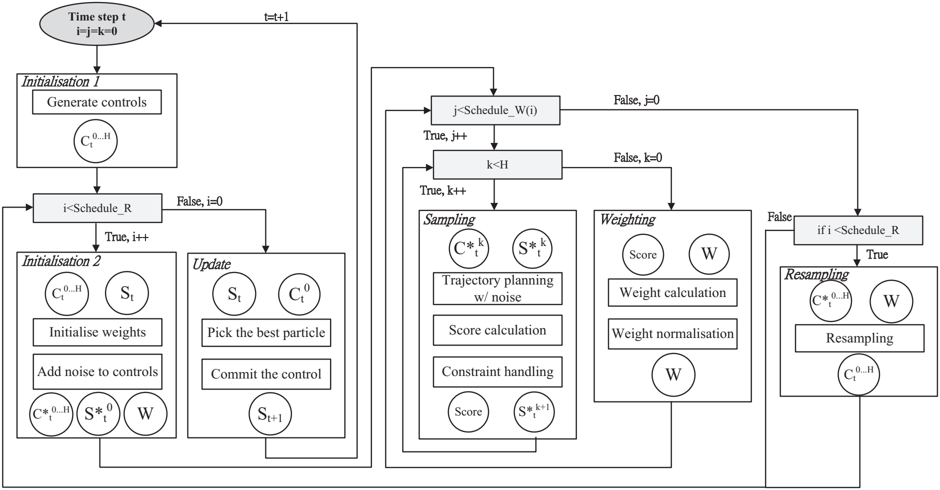
$$(\phi_{min} \leq \phi \leq \phi_{max}) \wedge (\tau_{min} \leq \tau \leq \tau_{max}) \wedge (T_{min} \leq T \leq T_{max}) \wedge (V_{min} \leq V \leq V_{max}) \wedge (a_{min} \leq a \leq a_{max}) \wedge (M_{min} \leq M) \tag{1}$$

$$\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < 2P_R\right) \wedge (|a_i - a_j| < 2P_H) \wedge (j \neq i) \quad \forall j \in \{0, \ldots, N_A - 1\} \tag{2}$$

Weight calculation

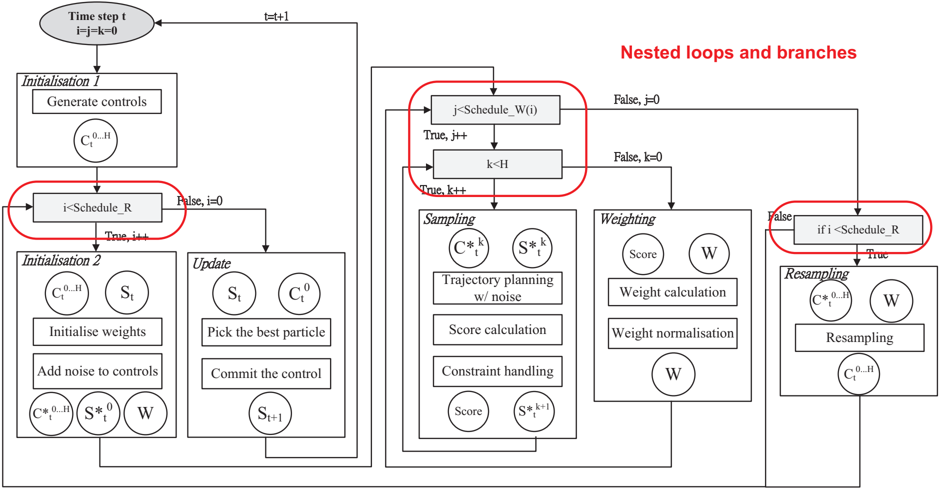
$$W = \sum_{k=0}^{H-1} J(k) \tag{3}$$

Mapping to FPGA



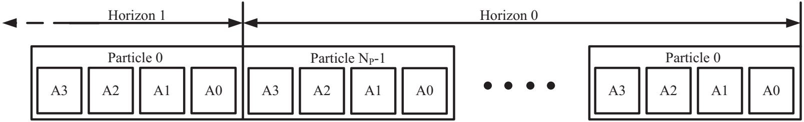
Challenge 1: Nested loops and branches

- Design control logic? Promote data reuse?



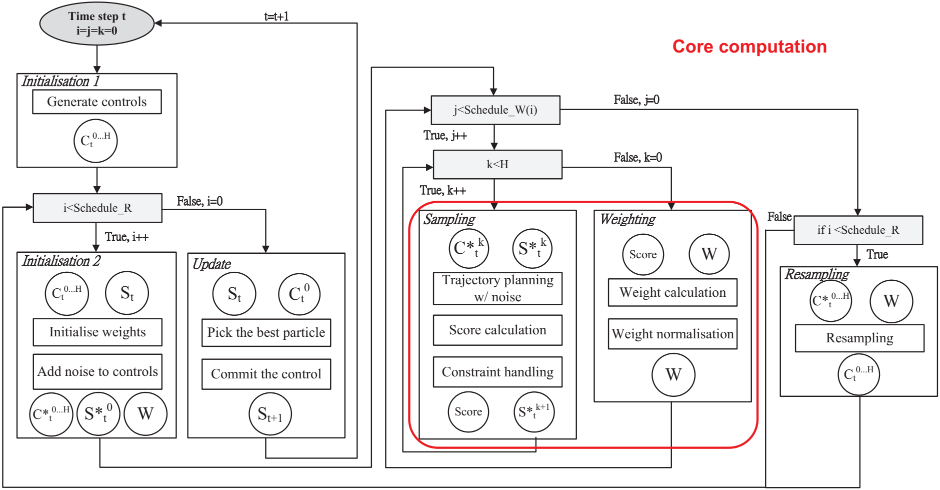
Solution to challenge 1: Particle stream

- Organise particles as a stream
- Divide to H horizons
- Each horizon has N_P particles
- Each particle contains information for A aircraft
- Allow pairwise comparison between aircraft
- Maintain streaming: one datum per clock cycle



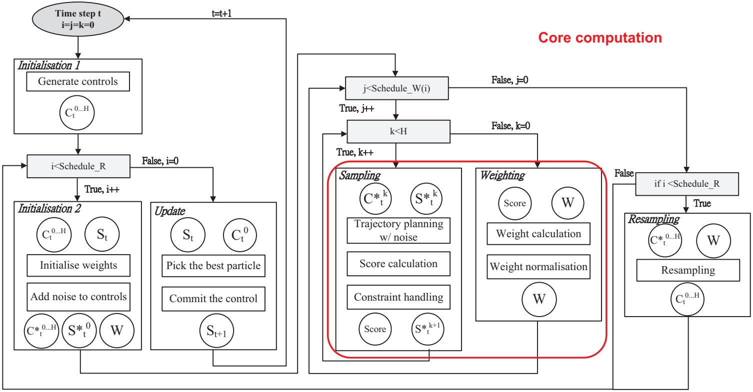
Challenge 2: Iterative processes

- Identify hotspots? Maximise throughput?



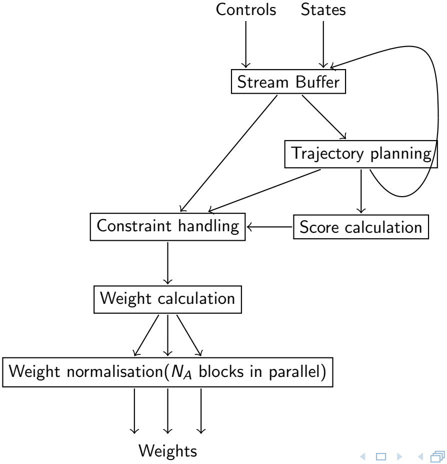
Solution to challenge 2: Kernel design

- Keep control decisions outside the kernel
- Datapath-oriented, fully-pipelined
- Replicate as many times as FPGA resources allow



Solution to challenge 2: Kernel design

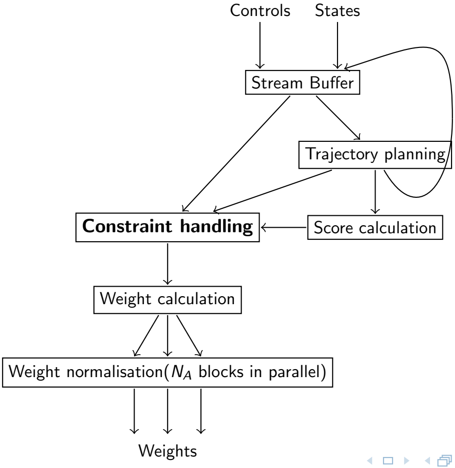
- Keep control decisions outside the kernel
- Datapath-oriented, fully-pipelined
- Replicate as many times as FPGA resources allow



Solution to challenge 2: Kernel design

Non-sequential data access ☹

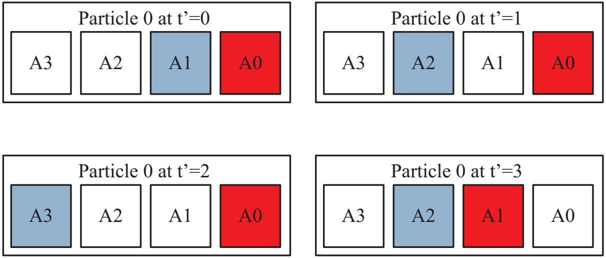
- Pairwise comparison of aircraft's location



Solution to challenge 2: Kernel design

Non-sequential data access ☹

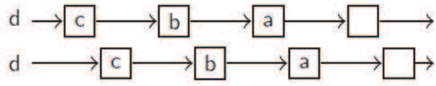
- Pairwise comparison of aircraft's location



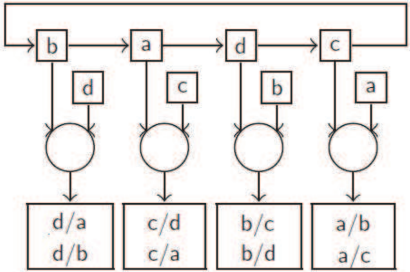
Solution to challenge 2: Kernel design

Non-sequential data access ☹ → Constraint checker ☺

1. Shift data of A aircraft (A cycles)
2. Check constraints in A pairs (A cycles)



Stage 1 - Shifting

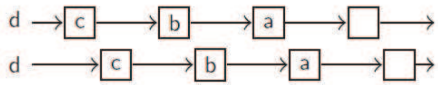


Stage 2 - Constraint checking

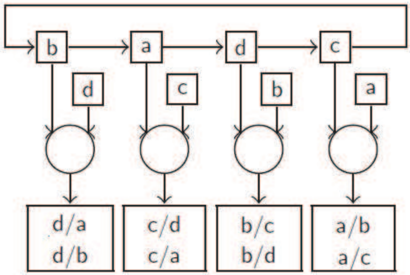
Solution to challenge 2: Kernel design

Throughput is halved ☹

- 2A cycles to process A aircraft



Stage 1 - Shifting

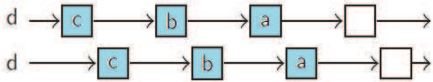


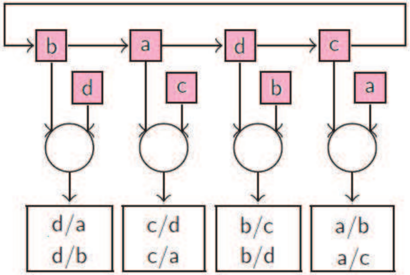
Stage 2 - Constraint checking

Solution to challenge 2: Kernel design

Throughput is halved ☹ → Double buffering ☺

- Duplicate the shift registers
- Multiplex the values to one constraint checking unit
- Parallelise: shifting stage and constraint checking stage

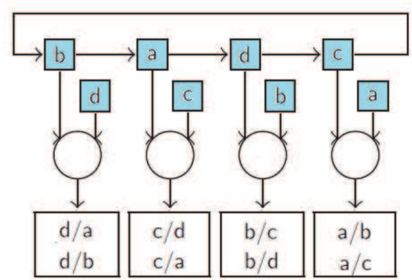


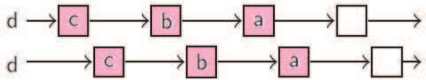


Solution to challenge 2: Kernel design

Throughput is halved ☹ → Double buffering ☺

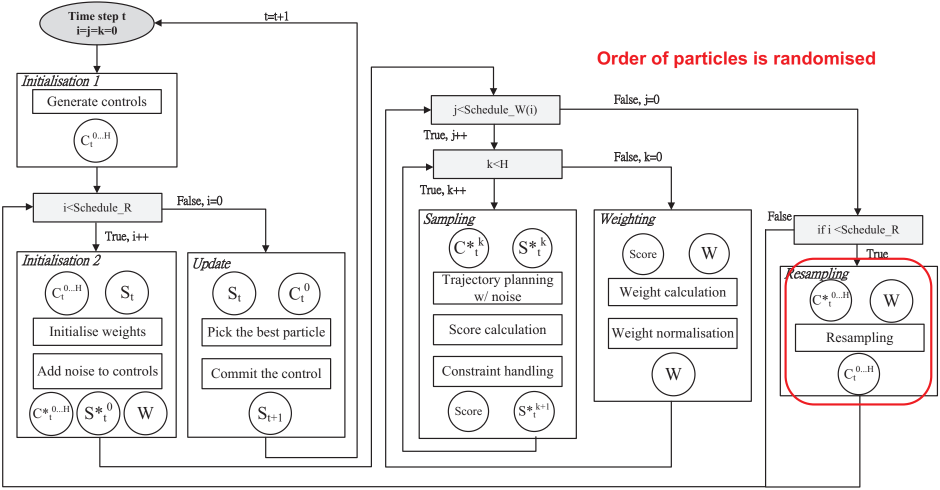
- Duplicate the shift registers
- Multiplex the values to one constraint checking unit
- Parallelise: shifting stage and constraint checking stage





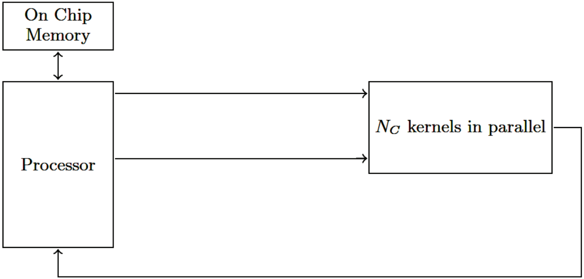
Challenge 3: Random ordering of data due to resampling

- No streaming?



Solution to challenge 3: Instruction-based processor

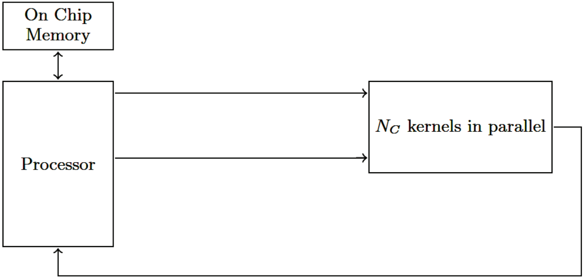
- Control the iterative invocation of kernels
- Coordinate the streaming of data between each iteration
- Allow random addressing of particle data during resampling



Solution to challenge 3: Instruction-based processor

Interfacing problem ☹

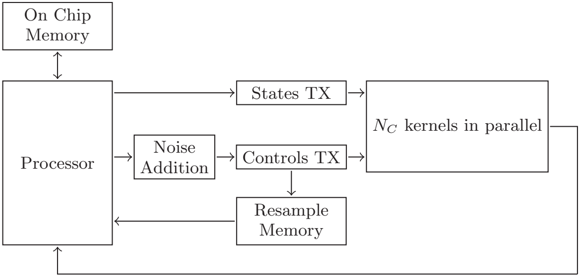
- One processor covers multiple kernels
- Processor writes data serially with limited width
- Each kernel accepts a larger data width and receives the same set of data many times



Solution to challenge 3: Instruction-based processor

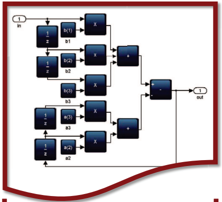
Interfacing problem ☹ → Custom interface ☺

- Avoid the bottleneck: processor transfer large amounts of data for once only.
- The interface transfers data in burst through DMA.



Design tools

User



Algorithm

parameterisable input file

Tool

- **Optimise**
  - **Compiler**
  - **Cookbook**
- **Map to device**
- **Pipeline**
- **Schedule**
- **Generate**

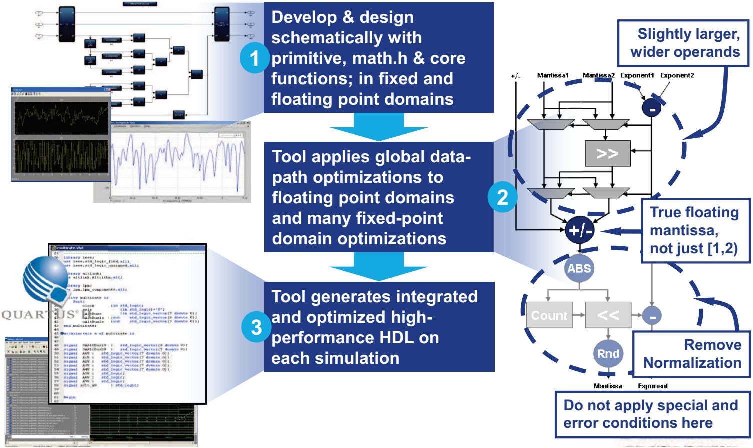
Implementation

HDL + hw.tcl

ALTERA MEASURABLE ADVANTAGE™

(from Altera)

Design tools



(from Altera)

Comparison with CPU and GPU: 4 aircraft scenario

	CPU [a]	CPU [a]	GPU [b]	FPGA1 [c]	FPGA2 [d]
No. cores	1	4	448	1	5
Comp. Time (s)	77	36.2	5.1	3.9	2.2*
Time eff.	1x	2.1x	15.1x	19.7x	35.0x
Active power (W)	175	247	265	26	-
Idle power (W)	133	133	153	19	-
Energy (J)	13475	8941	1352	101	-
Energy eff.	1x	1.5x	10.0x	133.4x	-

[a] CPU: Dual Intel Core i7-950 @3.07GHz, optimised by Intel Compiler.

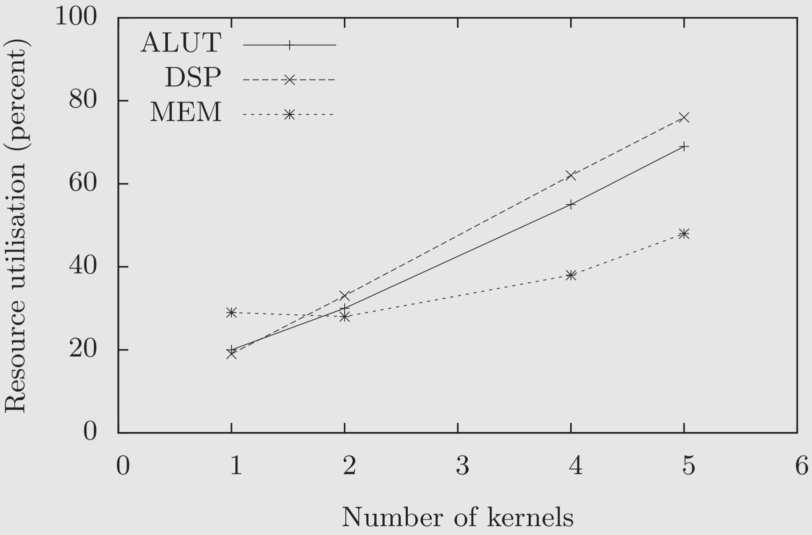
[b] GPU: Nvidia Tesla C2070.

[c] FPGA1: Altera Stratix IV EP4SGX530 (Kernel@170MHz, processor@190MHz).

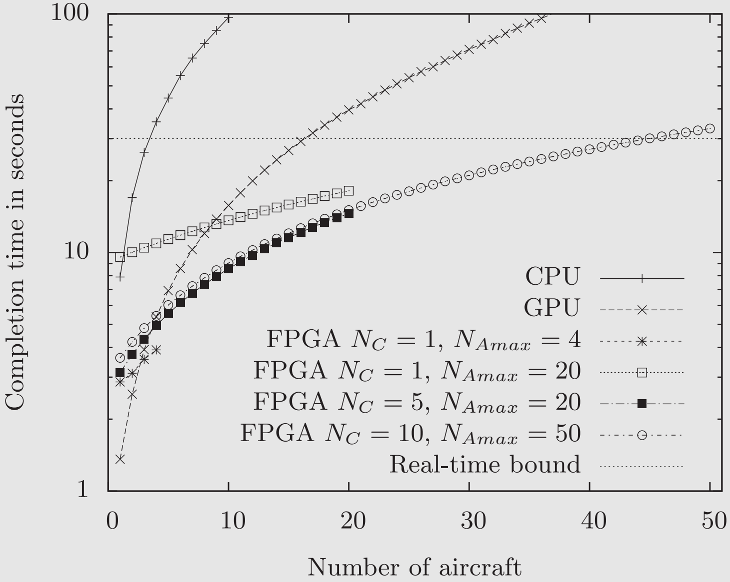
[d] FPGA2: Altera Stratix V 5SGSD8 @150MHz.

* Estimated.

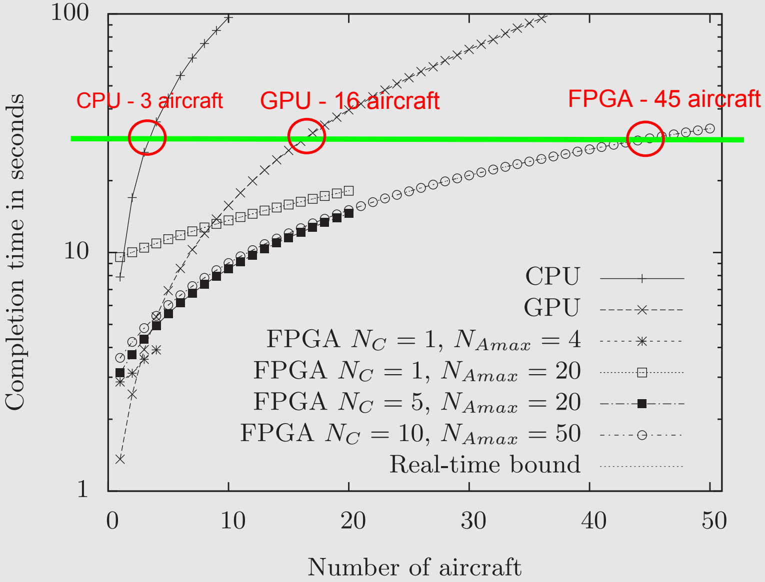
Resource utilisation on Altera Stratix V 5SGSD8



Compute within real-time



Compute within real-time



Current and future work

Extend to support future air traffic management

- More aircraft
- Dynamic change of air traffic
- **More sophisticated model: fuel usage, cruise time**

Extend to support more control applications

- City traffic control: pedestrian, cars, traffic lights

Acceleration of the sequential Monte Carlo method

- Novel particle stream structure for evaluating constraints and weights
- Separate control from data path to promote scalability

Applied to air traffic management

- Altera Stratix V FPGA at 150MHz
- Process 4 aircraft in 3.9s; 45 aircraft in 30s
- First to meet real-time requirement

Speedup

- 35 times faster, 133 times more energy efficient over Intel Core-i7 950 CPUs (1 core)
- 16 times faster, 89 times more energy efficient over Intel Core-i7 950 CPUs (4 cores)
- 2.3 times faster, 13.5 times more energy efficient over NVIDIA Tesla C2070 GPU (448 cores)