

# Parallel collision avoidance and stereo navigation algorithms

**Pavel Zaykov,**

Zai Jian Jia Li, Lucie Matusova

Honeywell International,  
Brno, Czech Republic

- Company's portfolio
- Avionics applications – overview & parallelization
- Execution time analysis
  - WCET, WCET speedup, and WCET efficiency by Observed, RapiTime, and OTAWA
  - COTS multi-core timing analysis
- Summary



**\$39.1B**

in sales

**55%**

sales outside U.S.

- 1,300 sites, 68 countries
- 131,000 employees
- Morristown, NJ  
headquarters
- Fortune 100



**Aerospace**



**Performance Materials  
and Technologies**



**Automation and  
Control Solutions**



**Transportation Systems**

***Highly Diversified, Technology Driven Industrial Company***

**Honeywell businesses:**

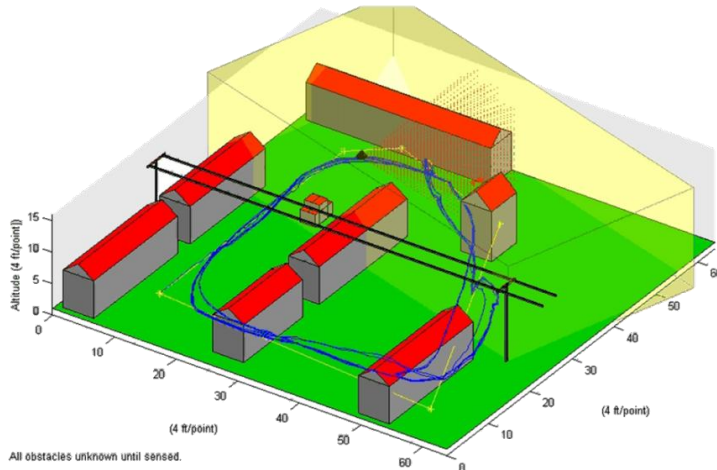
- Air Transport & Regional
- Business & General Aviation
- Defense & Space

**Honeywell solutions:**

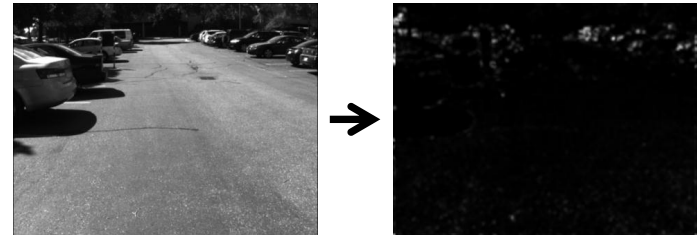
- Runway safety
- Flight safety
- Air Traffic Management
- In-flight connectivity
- Propulsion
- Precision guidance
- Predictive intelligence
- Wheels and brakes
- Logistics services

In parMERASA project, avionics domain is presented by 2 applications

## 3D Path Planning (3DPP) Collision Avoidance



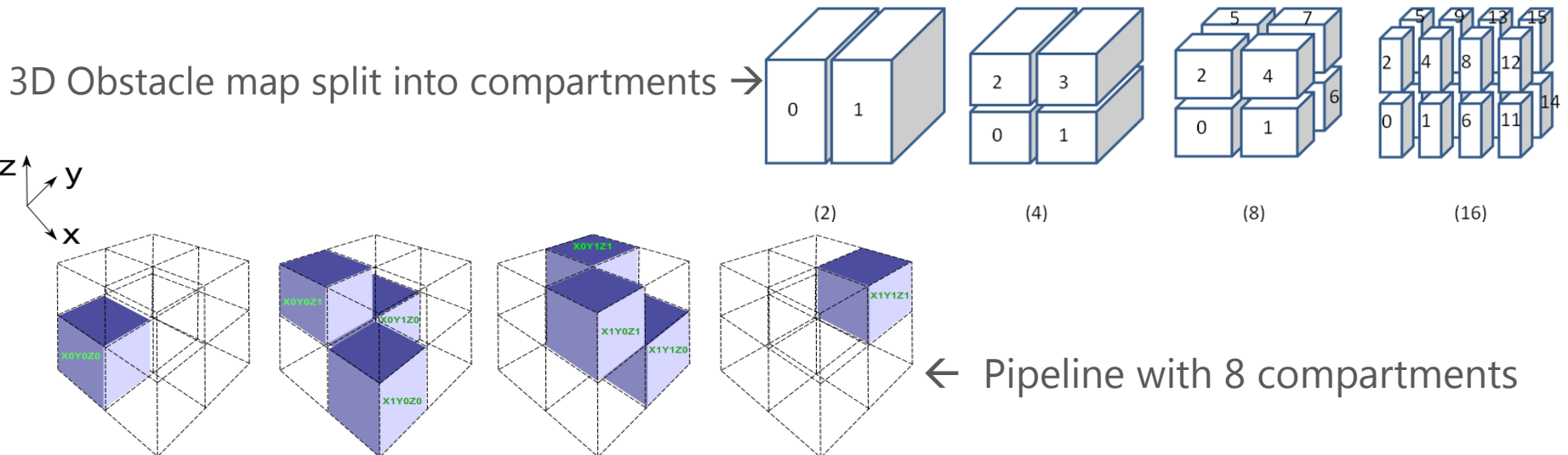
## Stereo Navigation (SN)



Picture  
[www.mobilerobots.com](http://www.mobilerobots.com)

source:

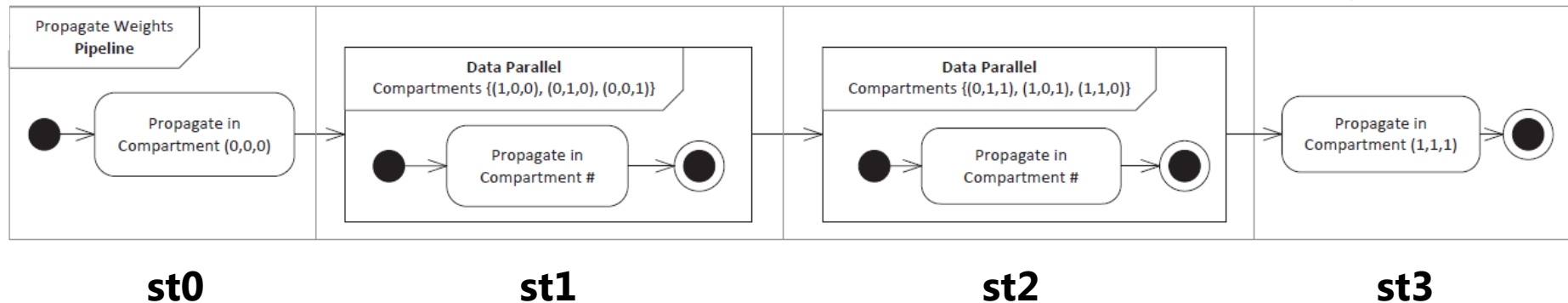
- Update rate: **1 Hz** → Desirable update rate: **4 Hz**



- Parallel Design Patterns:
  - Data parallelism
  - Task parallelism (Pipelining)
- Sync with **barriers**

- 3DPP configurations:
  - 1) 1 thread (sequential)
  - 2) 2 threads
  - 3) 4 threads
  - 4) 8 threads
  - 5) 16 threads

An exemplary configuration with 8 threads



- Update rate: **1 Hz** → Desirable update rate: **4 Hz**



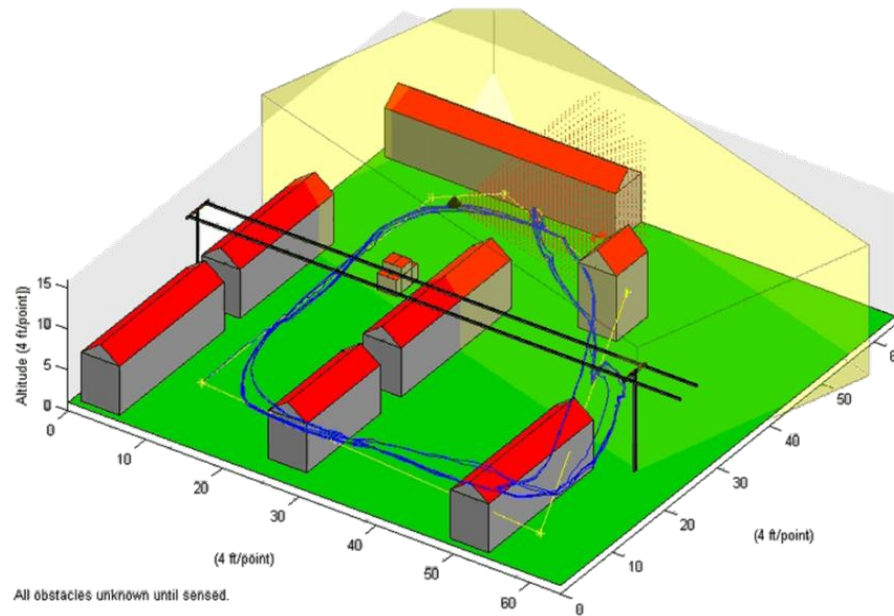
- Parallel Design Patterns:
  - Data parallelism - L/R images, tiles, filtering...
  - Task parallelism (Pipelining)
- Sync with **barriers**

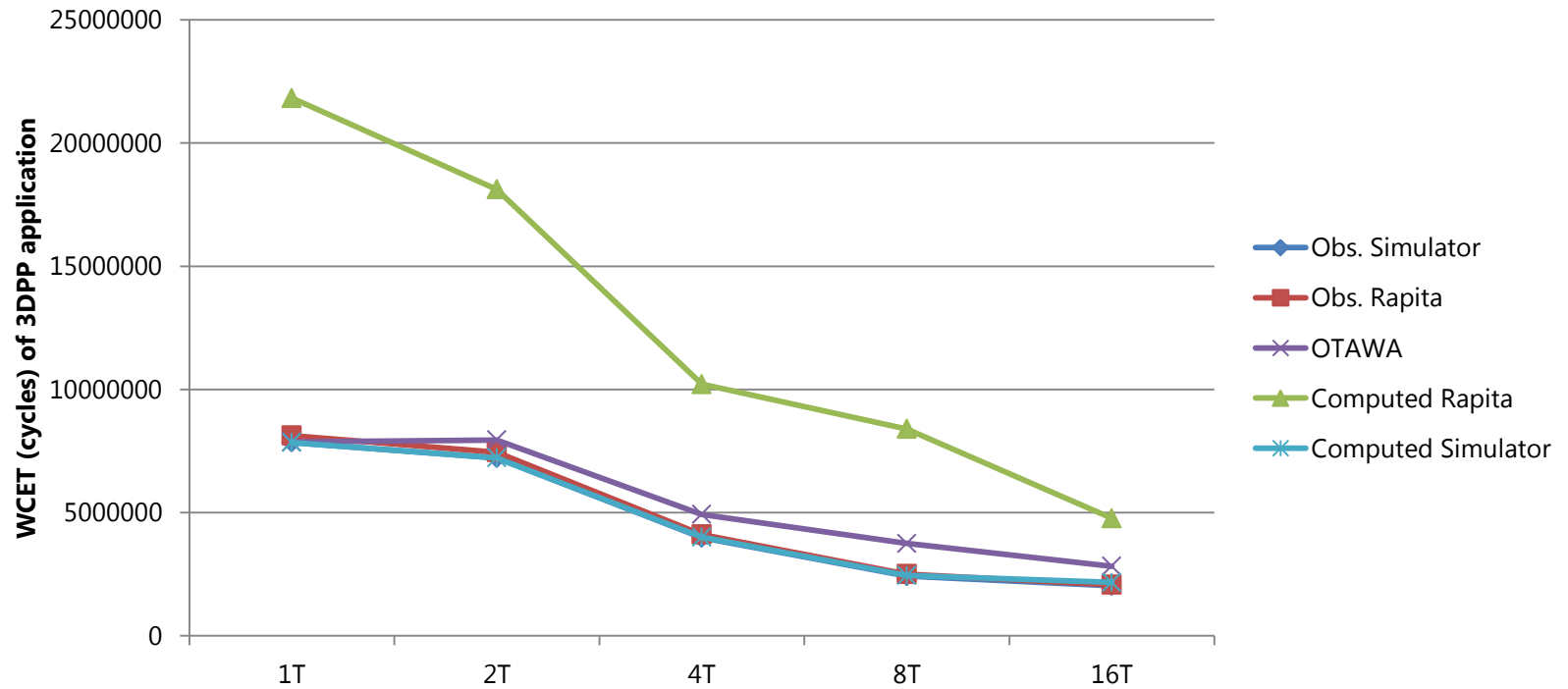
- Application configurations:
  - 1) 1 thread (sequential)
  - 2) 7 threads
  - 3) 12 threads (filtering)
  - 4) 12 threads (cameras)
  - 5) 14 threads
  - 6) 16 threads

- Observed Simulator (Obs. Simulator)
  - Application observed execution time (non-intrusive)
- Observed RapiTime (Obs. Rapita)
  - Application observed execution time + RapiTime I-points
- Static WCET analysis by OTAWA (OTAWA)
  - Application Control-Flow Graph + platform model
- Measurement-based WCET by RapiTime (Computed Rapita)
  - Application Control-Flow Graph + RapiTime “magic”
- Computed Simulator
  - $\sum \mathbf{max}$  (observed function execution time)



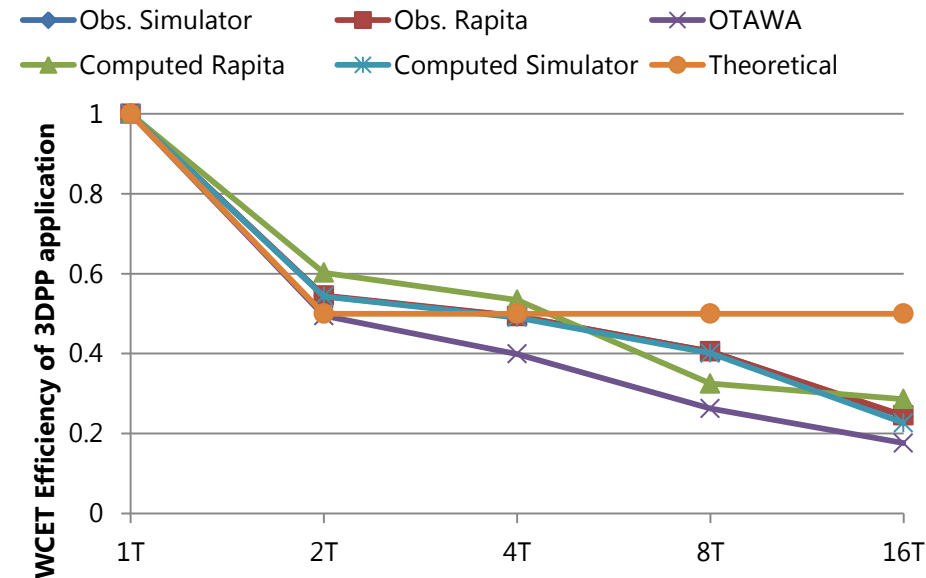
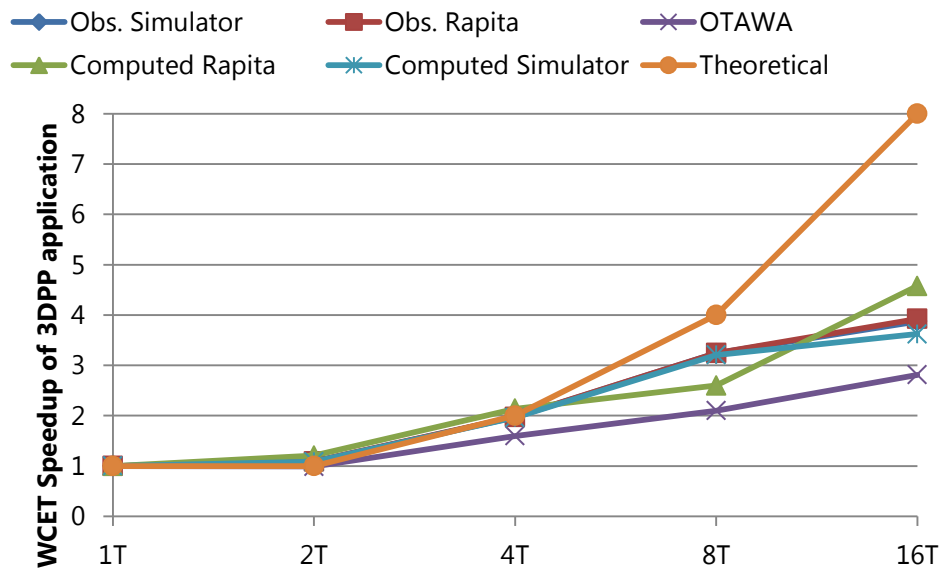
# 3DPP execution time analysis





3DPP WCET with various number of threads

- OTAWA has a “tight” WCET estimation due to:
  - accurate platform model
  - correctly set cache preconditions
- $WCET_{Computed\ RapiTime} > WCET_{OTAWA}$



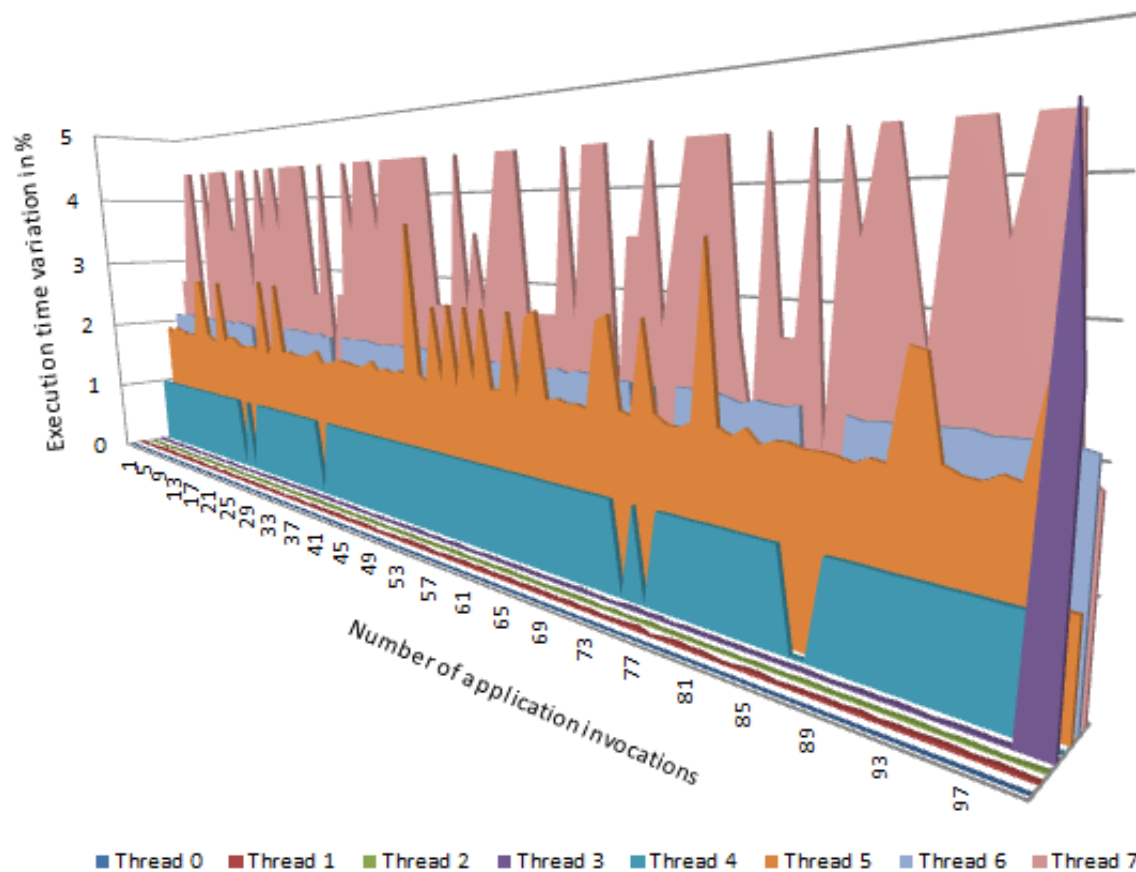
3DPP WCET speedup and efficiency with various number of threads

$$WCET_{speedup} = \frac{WCET_{sequential}}{WCET_{parallel}};$$

$$WCET_{efficiency} = \frac{WCET_{speedup}}{num\ of\ cores};$$

- Obs. Speedup > Theoretical speedup (Wrong estimation of theor. due to a fine-grain parallelism)
- WCET speedup goes up to **4x** with 16T implementation.
- The application "sweet spot" is with 4T-8T:
  - stress on shared resources
  - input dataset variation
  - low ratio between computation and communication

## 3DPP threads execution time



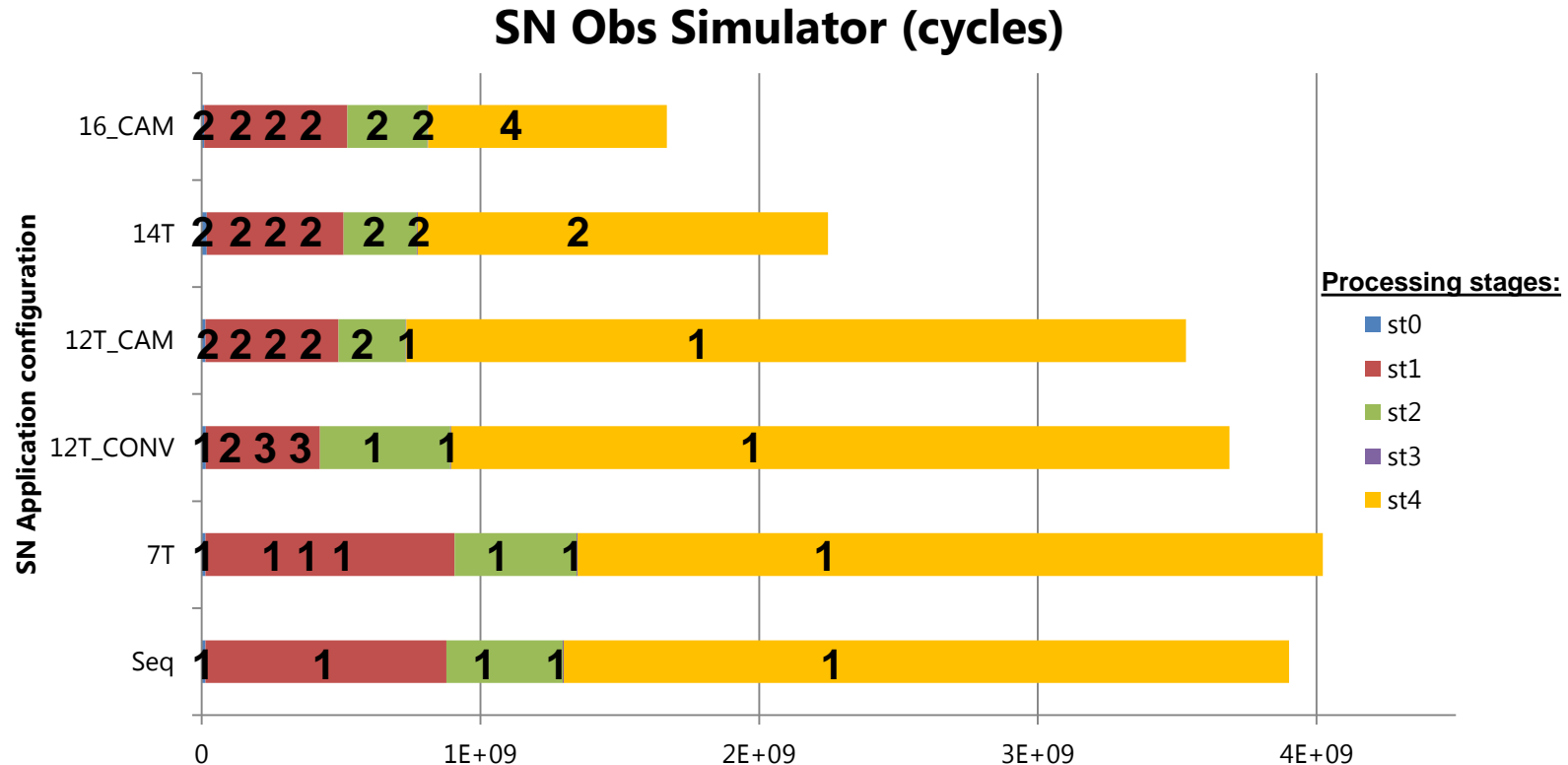
- **8** threads configuration
- **100** 3DPP iterations
- Preemptive RTOS
- 2-core COTS platform
  - core 0: thread 0
  - core 1: thread 1-7

- Minor Obs. ET (exec. Time) variation (<5%) =  $\frac{ET - ET_{min}}{ET_{max}} * 100$

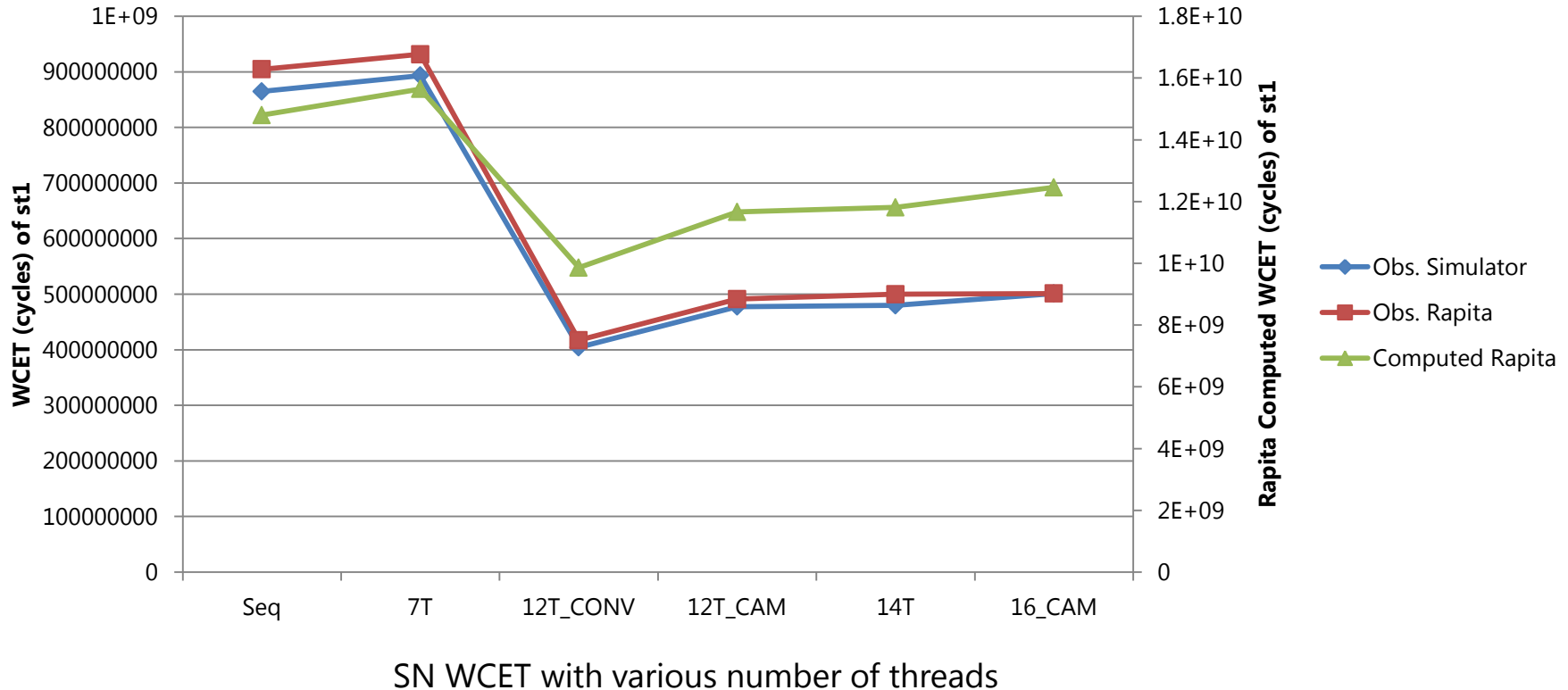
## SN execution time analysis



Picture source: [www.mobilerobots.com](http://www.mobilerobots.com)

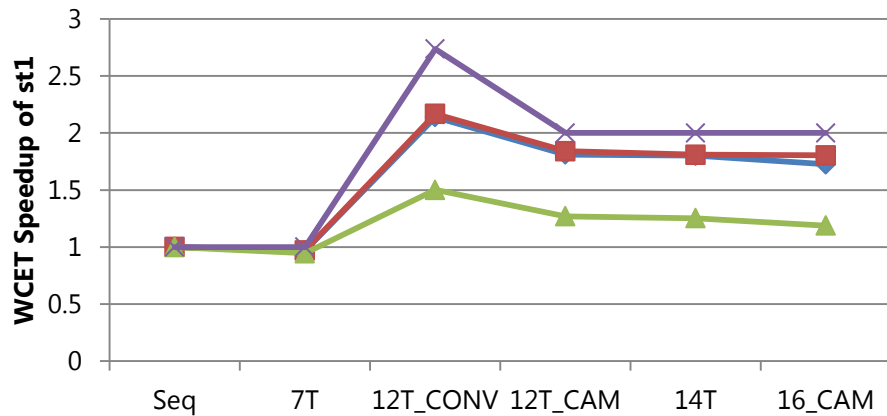


- We analyze SN **st1** (data filtering) stage in term of WCET
- SN st4 (pose estimation) relies on probabilistic analysis & does **not** require RT
- 7T app config has longer Obs. Simulator due to **no** parallelization + comm. overhead

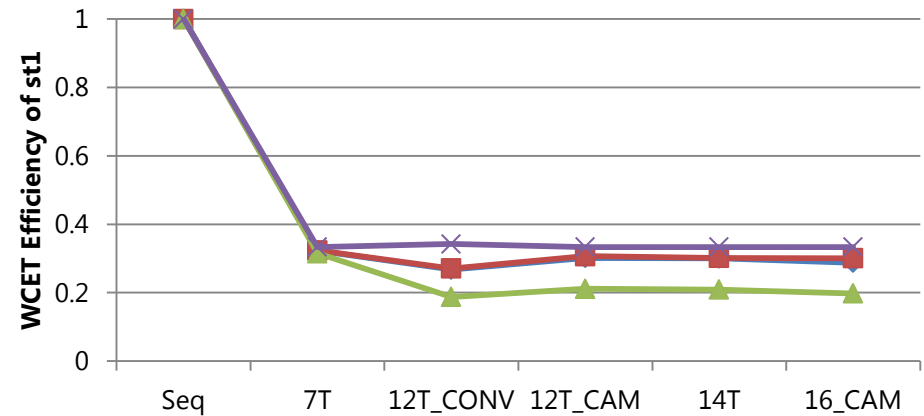


- The WCET trend is observed in all WCET analysis
- $WCET_{RapiTime} \gg WCET_{Obs}$  due to analysis of barrier implem. & nested loops
- 12T\_CAM, 14T, 16\_CAM  $\equiv$  exploit the same parallelism in st1, but WCET  $\uparrow$ 
  - caused by shared resources contentions

Obs. Simulator Obs. Rapita Computed Rapita Theoretical



Obs. Simulator Obs. Rapita Computed Rapita Theoretical



SN WCET speedup and efficiency with various number of threads

$$WCETspeedup = \frac{WCET_{sequential}}{WCET_{parallel}};$$

$$WCETefficiency = \frac{WCETspeedup}{num\ of\ cores};$$

- WCET speedup goes up to **2.6x** with 12T\_CONV application configuration
- The application "sweet spot" is with 12T
- WCET efficiency for >7T is the same among all app. configurations (0.2 – 0.3)



- Summary
  - Avionics domain is presented by 2 applications
  - WCET-aware parallelization is performed
  - WCET analysis based on Obs., RapiTime, and OTAWA
  - WCET, WCET speedup, WCET efficiency
  - COTS analysis on the Execution Time variations is performed
  
- We learnt
  - Static WCET analysis is tight with an accurate platform model
  - to identify app “sweet spots” for optimal WCET speedup
  - high number of cores causes an increase of ET variations
  - COTS platforms with preemptive RTOS have minor ET variations



*The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement FP7-ICT-287519*

Thank you

