A Parallel Simulation Approach to ACAS X Development

Adam Gjersvik

MIT/LL

This material is based upon work supported by the Federal Aviation Administration under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Federal Aviation Administration.

978-1-5386-6854-2/19/S31.00 2019IEEE

Collaborating Organizations

Federal Aviation Administration

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Outline

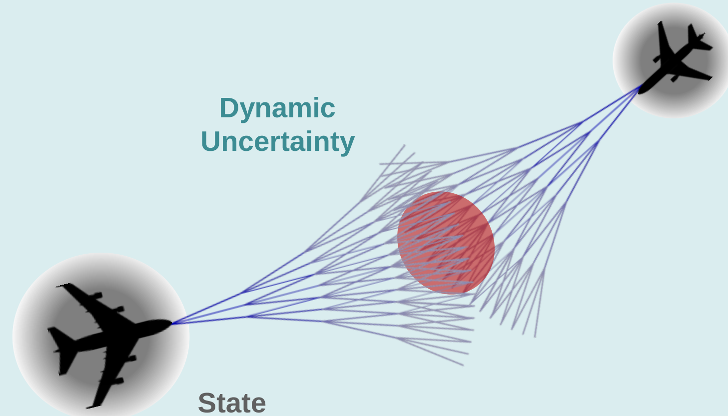
- **Background on Collision Avoidance Systems**
- **System Specification & Algorithm Development**
- **Parallel Simulation Approach**
- **Performance & Simulation Results**
- **System Characterization**
- **Conclusions**

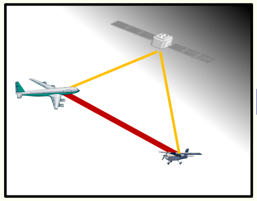
Next Generation Airborne Collision Avoidance System (ACAS X)

- **FAA program since 2009**
  - Improve safety & unnecessary alerts
  - Support reduced separation operations
  - Extend collision avoidance to UAS
  - Streamline development process
- **Lincoln Laboratory leading:**
  - Threat logic
  - Surveillance and tracking
  - Maneuver coordination
  - Flight test planning

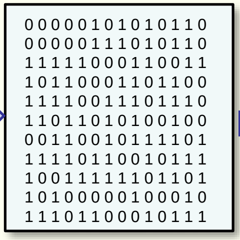
ACAS X supports future airborne collision avoidance requirements

Next Generation System





Additional Surveillance Sources



Improved and flexible threat logic

Standard Interface

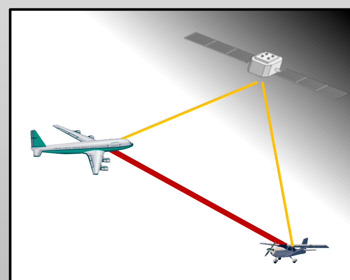
ACAS X Modular Architecture

Potential Surveillance Sources

- Interrogation/Reply
- Automatic Dependent Surveillance – Broadcast (ADS-B)
- Electro-Optical
- Primary Radar

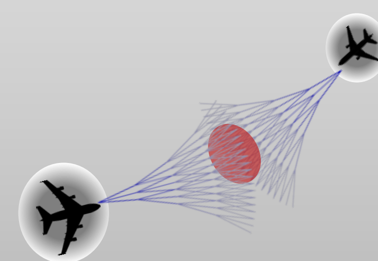
Surveillance and Tracking Module (STM)

Surveils nearby aircraft and provides TRM with tracked target data

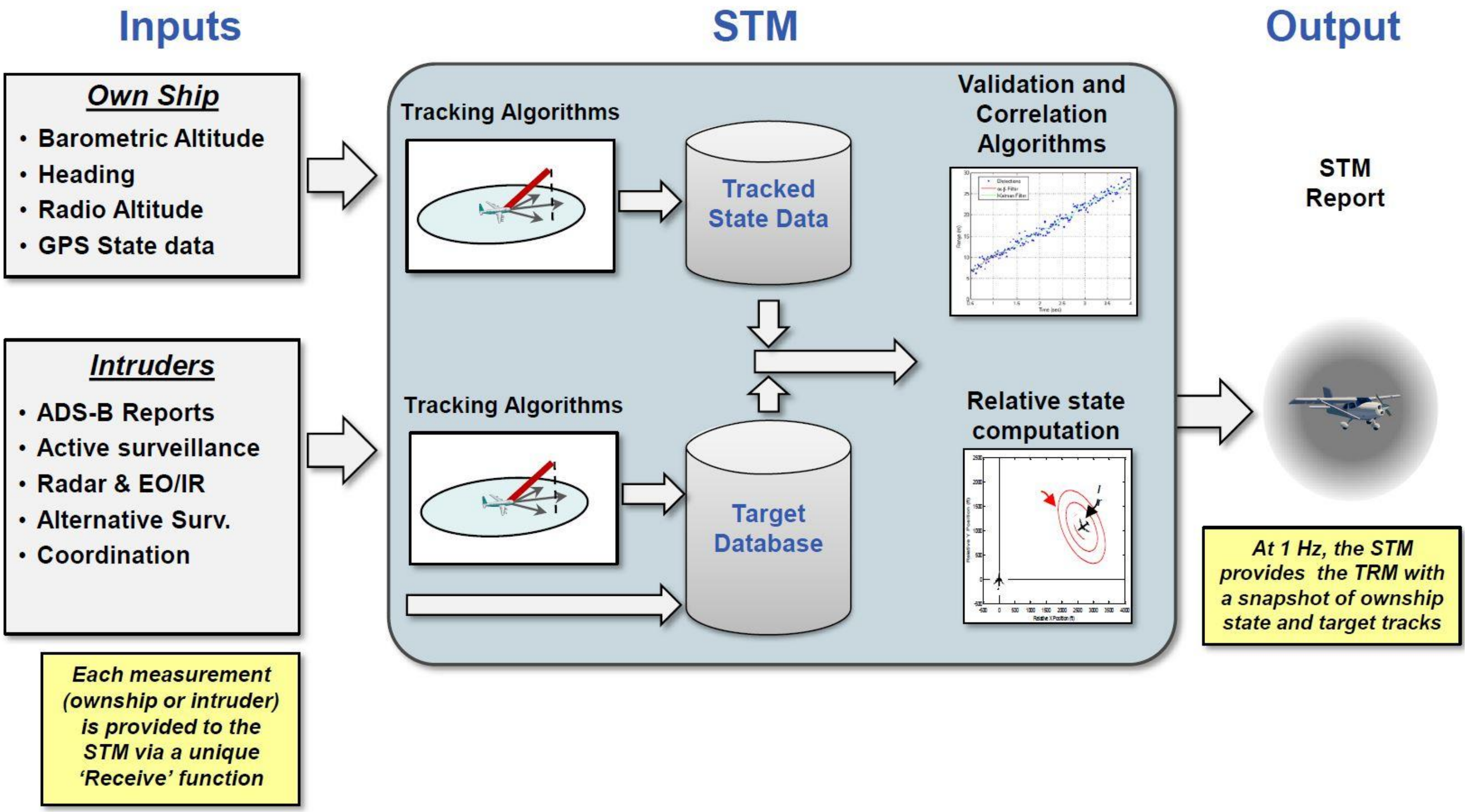


Threat Resolution Module (TRM)

Processes target information from STM to provide evasive maneuver command to pilot if necessary



Surveillance and Tracking Module (STM)

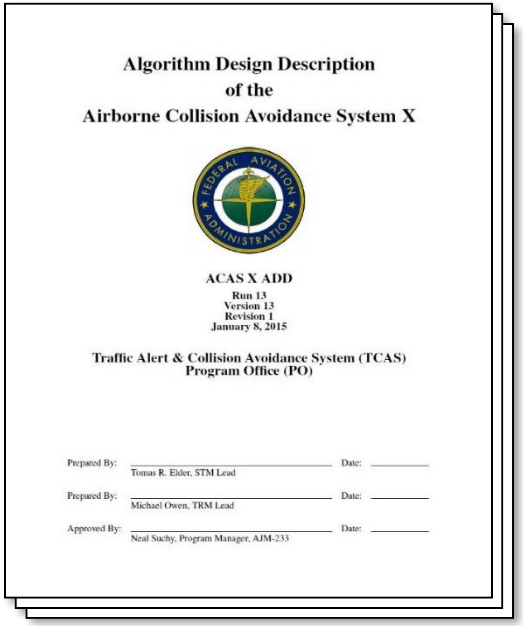


Outline

- Background on Collision Avoidance Systems
- System Specification & Algorithm Development
- Parallel Simulation Approach
- Performance & Simulation Results
- System Characterization
- Conclusions

System Specification with julia

- FAA need for rapid development of ACAS X system algorithms
- A high-performance executable specification language allows rapid testing and verification
- Julia provides specification and rapid prototyping in one language


Algorithm Design Description
of the
Airborne Collision Avoidance System X

ACAS X ADD
Run 13
Version 13
Revision 1
January 8, 2015

Traffic Alert & Collision Avoidance System (TCAS)
Program Office (PO)

Prepared By: ______ Date: ______
Tomas R. Elder, STM Lead

Prepared By: ______ Date: ______
Michael Owen, TRM Lead

Approved By: ______ Date: ______
Neal Suchy, Program Manager, AJM-233

Julia Characteristics

✓ Executable	✓ Readable and concise syntax	✓ High performance
✓ Open source	✓ Useful abstractions	✓ Platform independent results
✓ Extensive type system	✓ Interactive command line	

Julia meets the requirements of an improved specification language

Outline

- Background on Collision Avoidance Systems
- System Specification & Algorithm Development
- Parallel Simulation Approach
- Performance & Simulation Results
- System Characterization
- Conclusions

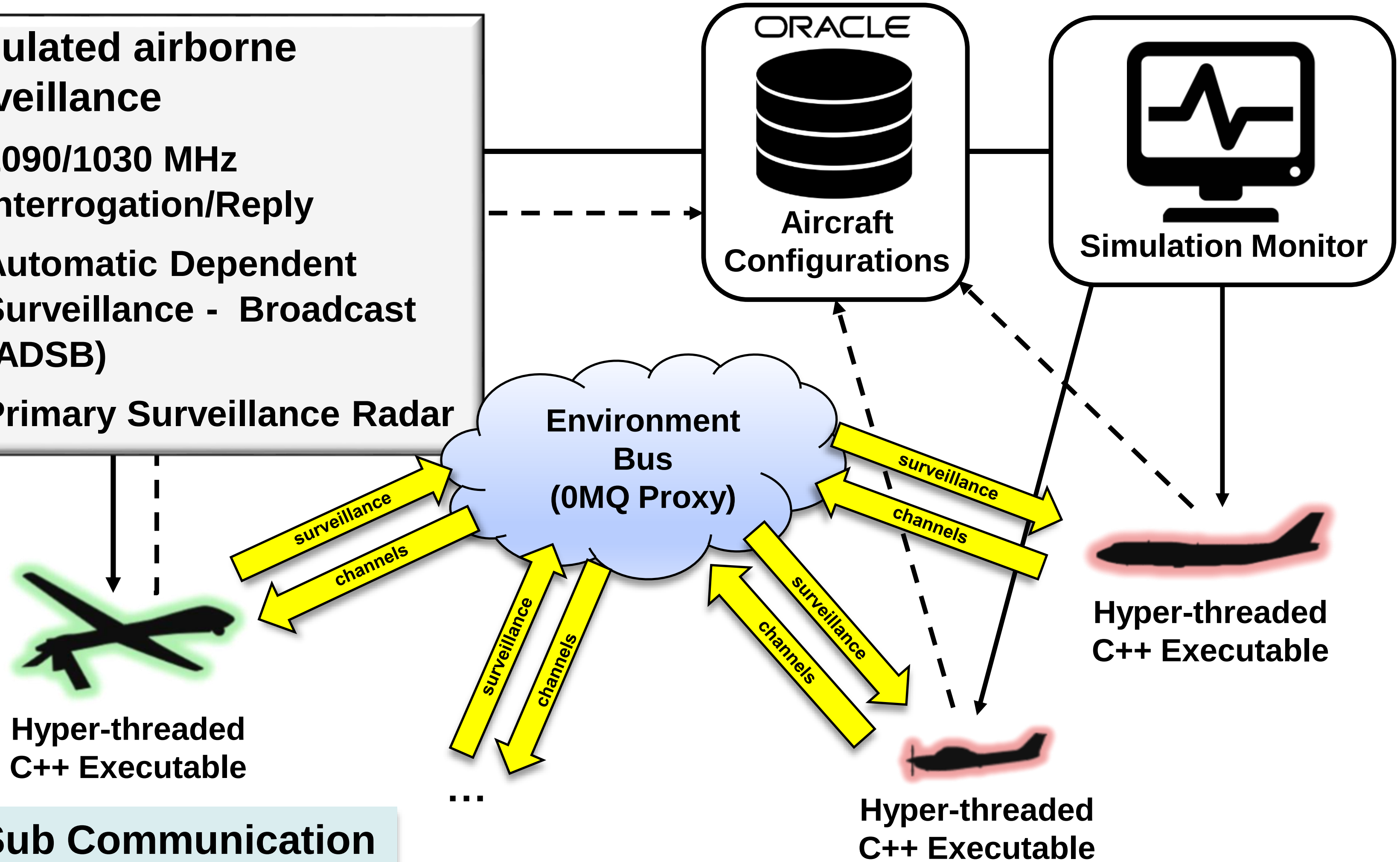
Computational Resources

- **Lincoln Laboratory Supercomputing Center (LLSC) TX-Green interactive, on-demand cluster**

	Intel Xeon CPU E5-2683 v3	**AMD Opteron Processor 6274**	**Intel Xeon 64 Core**
Cores per Node	28	32	64
Clock Speed	2.0 GHz	2.2 GHz	1.3 GHz
RAM per Node	256 GB	128 GB	192 GB
User Allocation	1024 cores	1024 cores	8072 cores

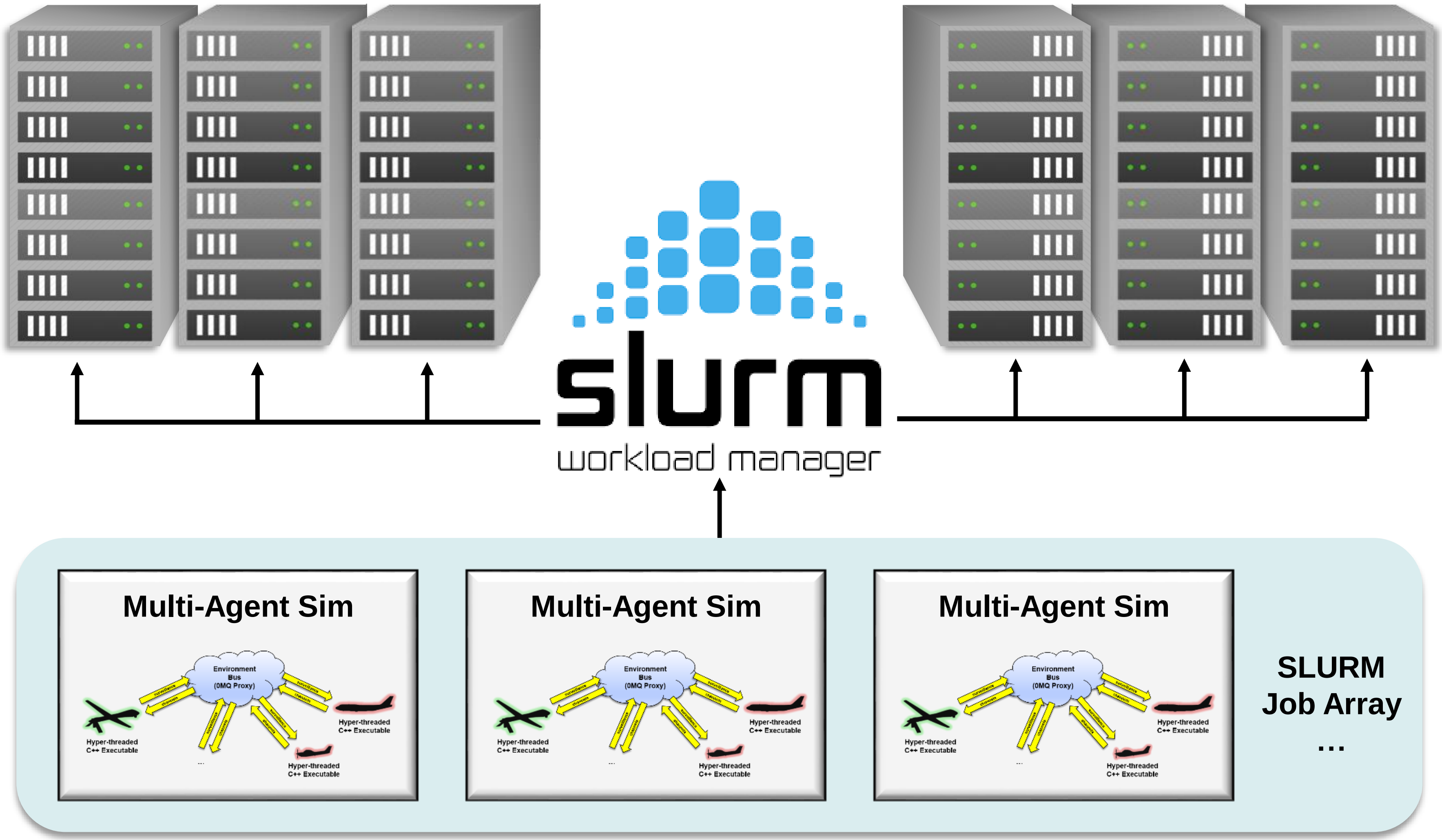
Real-Time Multi-Agent Aircraft Simulation

- **Simulated airborne surveillance**
  - **1090/1030 MHz Interrogation/Reply**
  - **Automatic Dependent Surveillance - Broadcast (ADSB)**
  - **Primary Surveillance Radar**

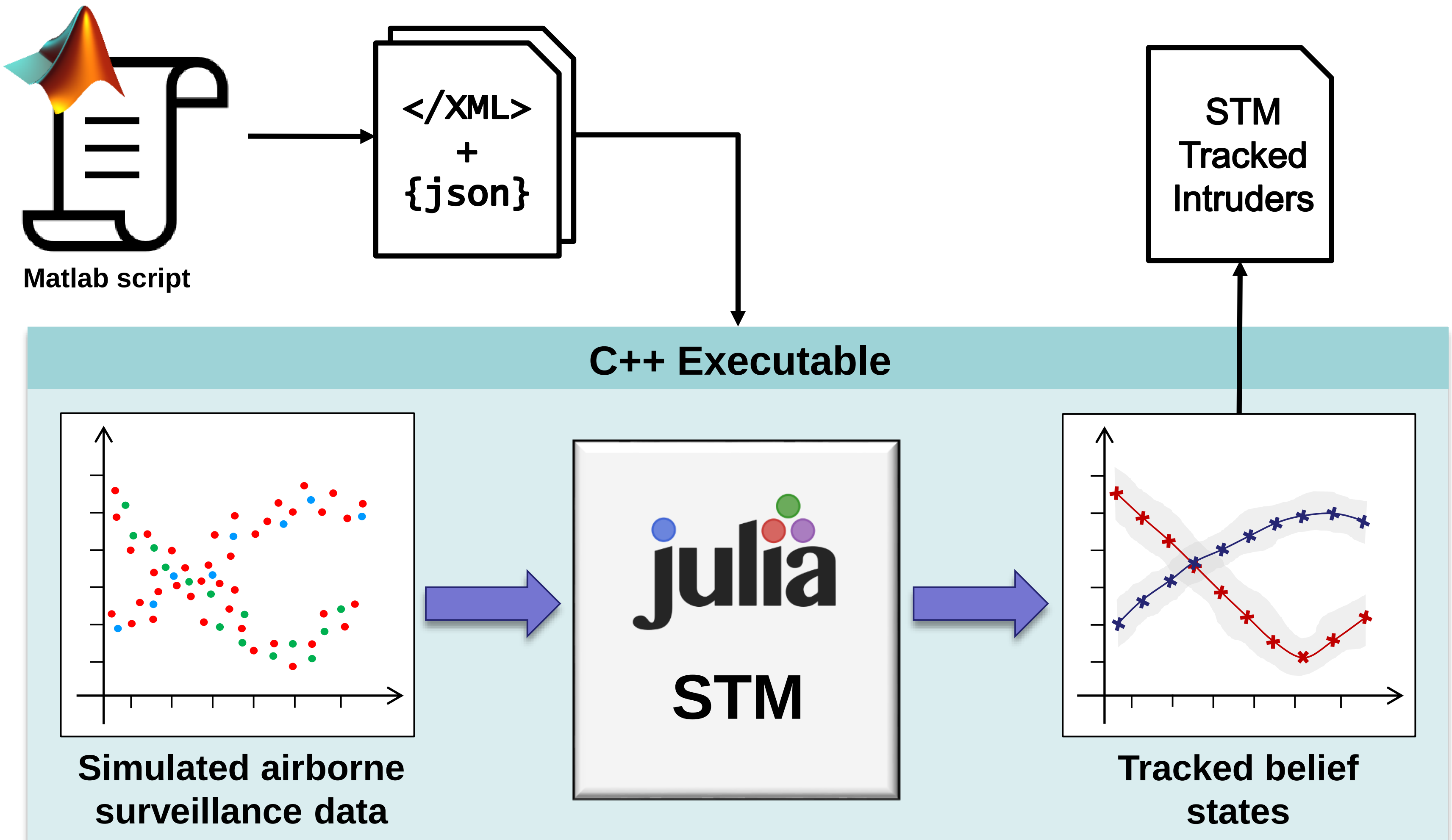


Pub/Sub Communication

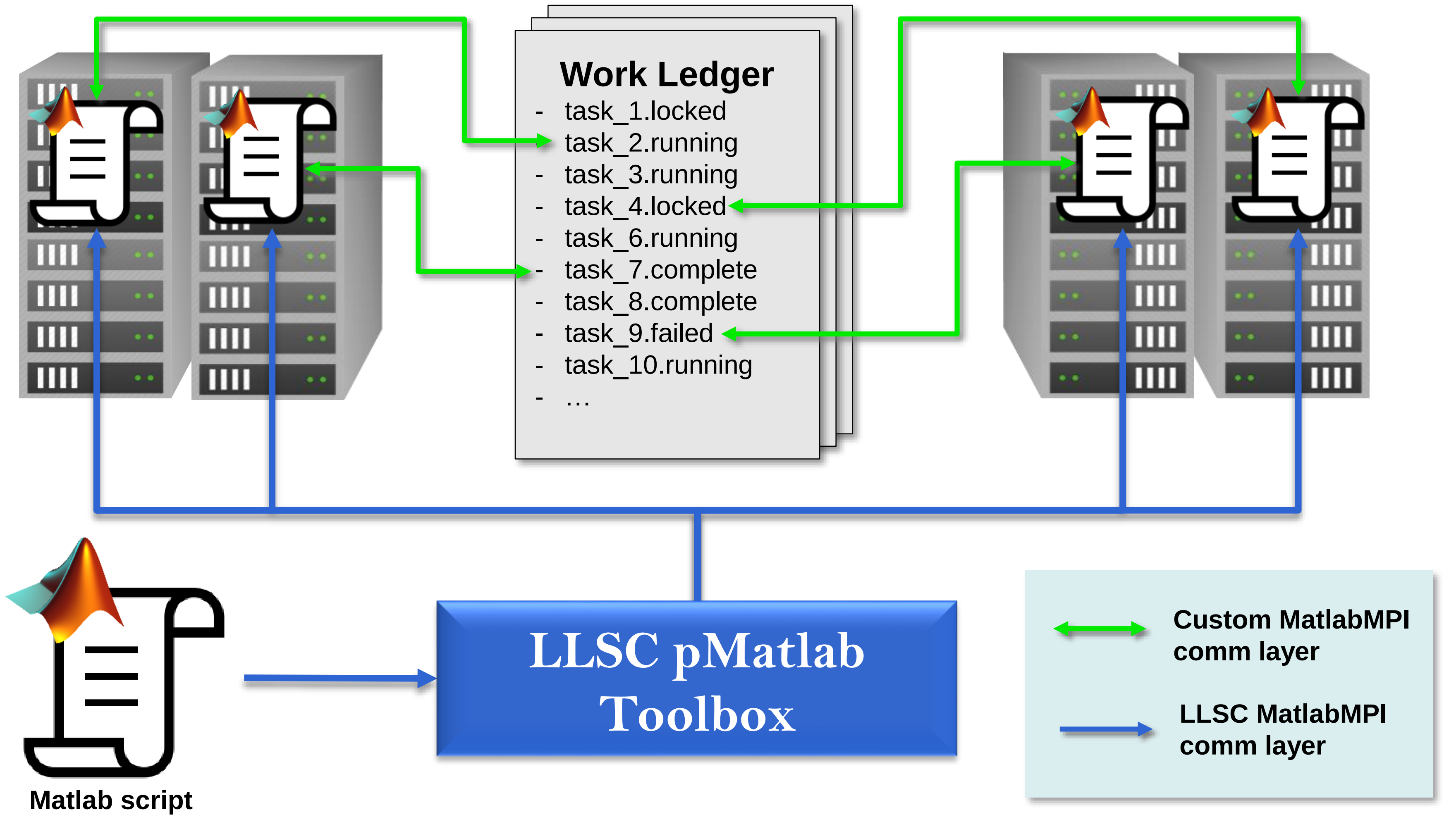
Parallel Multi-Agent Simulations



Fast-Time System Playback



Parallel Fast-Time Playback Simulations

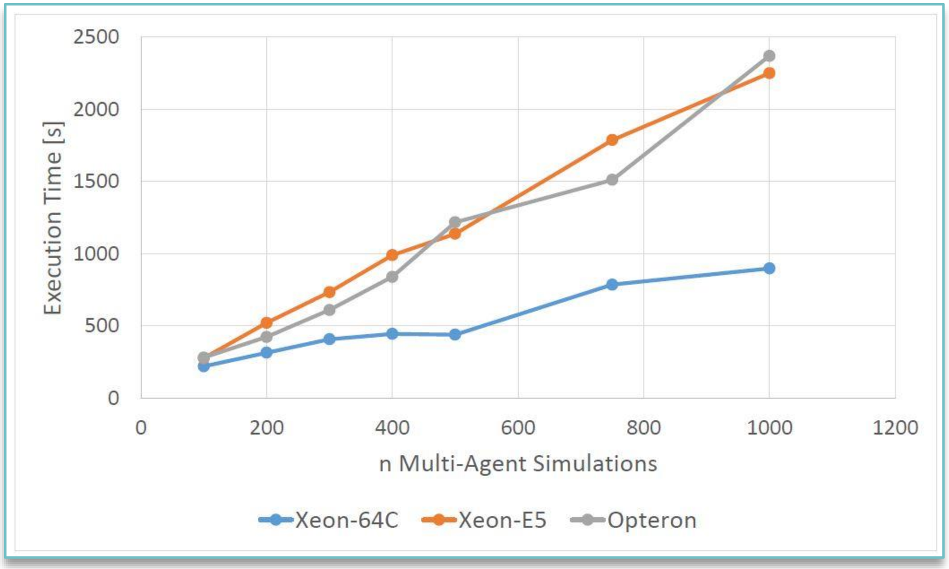


Outline

- **Background on Collision Avoidance Systems**
- **System Specification & Algorithm Development**
- **Parallel Simulation Approach**
- **Performance & Simulation Results**
- **System Characterization**
- **Conclusions**

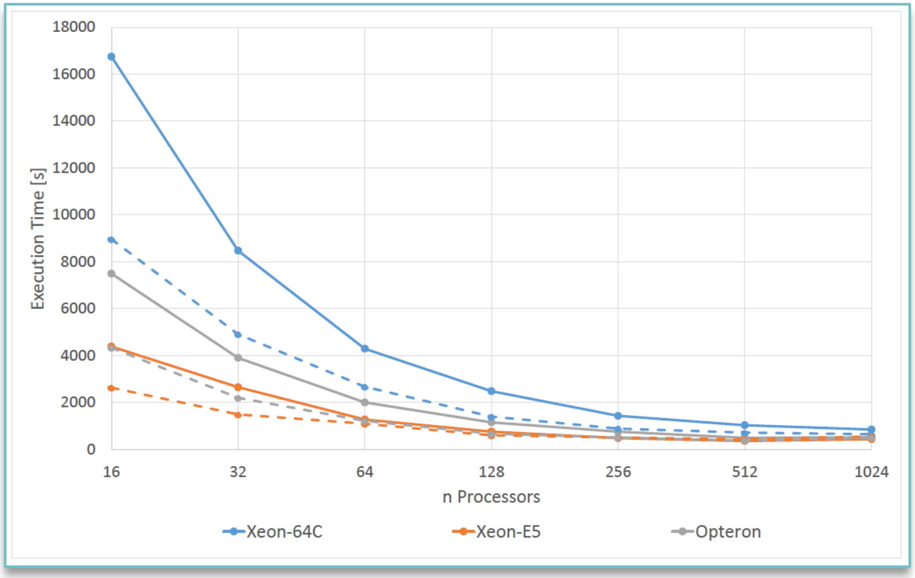
Real-Time Multi-Agent Simulation Speed

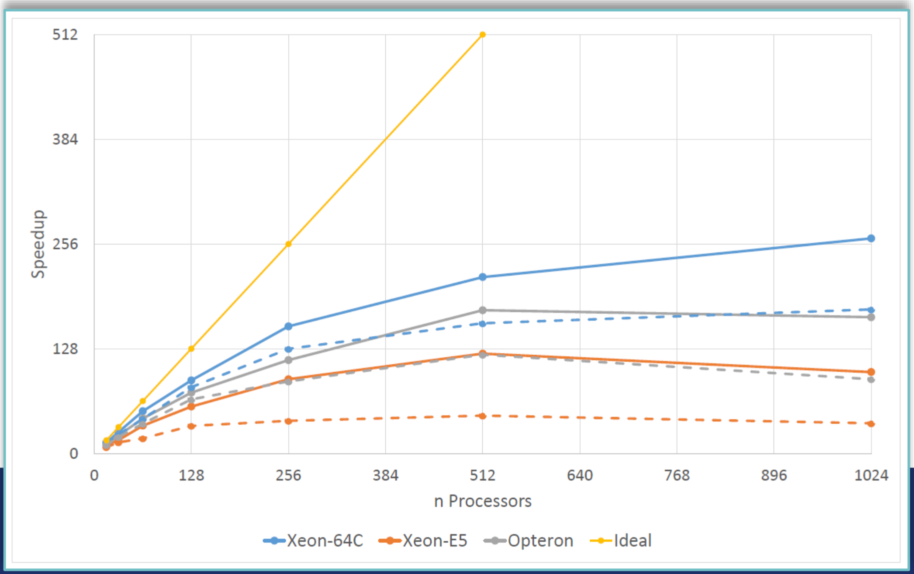
- **Individual multi-agent simulations executed on single exclusive cluster nodes**
- **Number of parallel multi-agent simulations limited by user node allocation**
- **Approximately linear increase in execution time due to static parallel batch sizes**
- **Higher Xeon64c user node allocation allows faster execution of parallel multi-agent simulations**



	Xeon-64c	Xeon-e5	Opteron
User Node Allocation	128	28	32

Fast-Time Playback Simulation Speed





- **Execution of 1500 fast-playback simulations on increasing numbers of processors**
- **Execution from Julia Image is nearly twice as fast as execution from source**
- **Execution on Xeon 64-core nodes is most benefited by high parallelization**

Execution from Julia Image

Execution from Julia Source

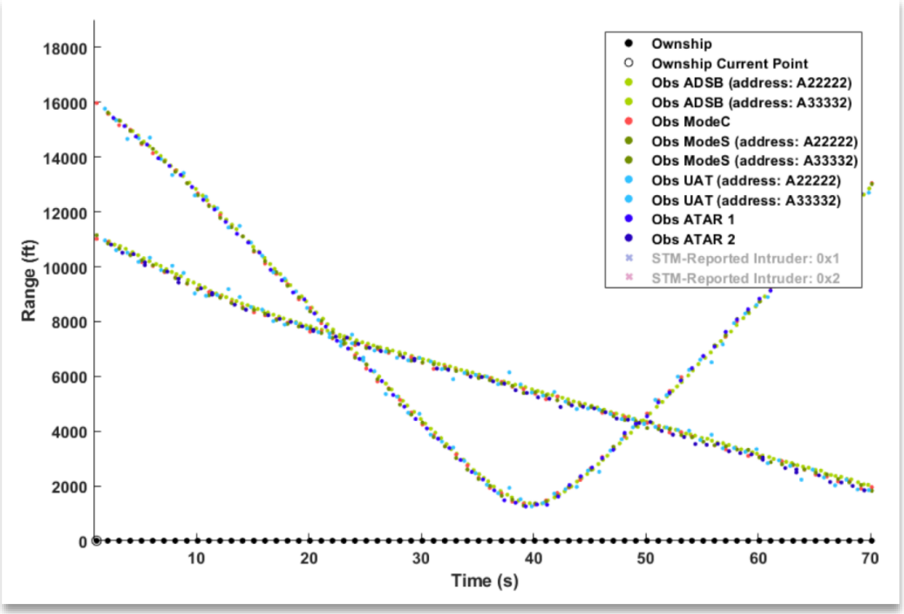
Individual Simulation Output Data

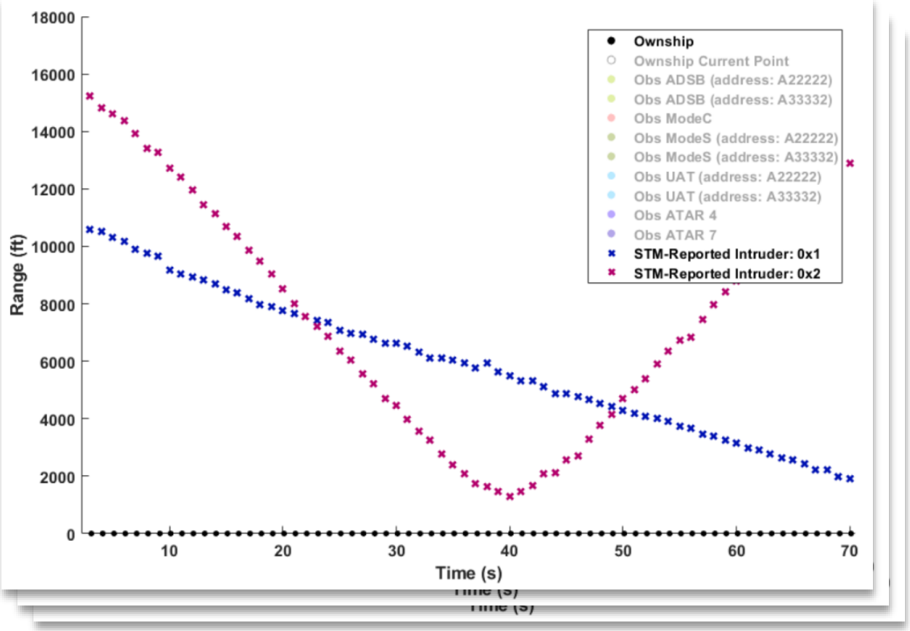
Multi-Agent Simulation Data

- Simulated airborne surveillance data on all surveillance channels (~800Kb)
- One multi-agent simulation per aircraft encounter geometry

Fast-Playback Simulation Data

- Data from combinations of surveillance source pairs fed to the STM for tracking (~250Kb)
- Many fast-playback simulations per aircraft encounter geometry

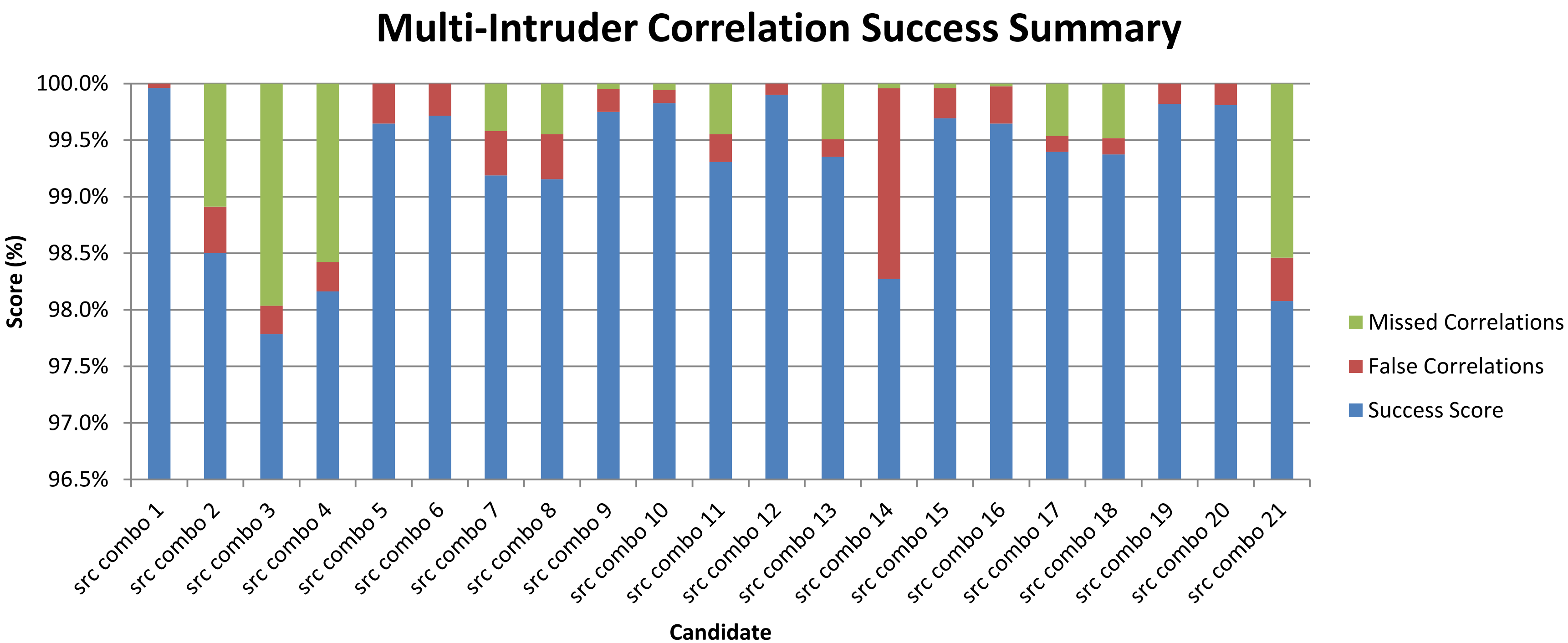




Outline

- Background on Collision Avoidance Systems
- System Specification & Algorithm Development
- Parallel Simulation Approach
- Performance & Simulation Results
- System Characterization ⇦
- Conclusions

Collective Performance Metrics



- ***Missed Correlation*** **= percentage of tracked seconds where more intruders were tracked in the airspace than were actually present (i.e. 3 intruders tracked when there were only 2)**
- ***False Correlation*** **= percentage of tracked seconds where less intruders were tracked in the airspace than were actually there (i.e. 1 intruder tracked when there were 2)**

- **Collective performance metrics enable statistical analysis of STM correlation performance**
- **Ability to simulate more encounters with parallelization adds statistical validity to collective performance metrics**

***Figure not indicative of current correlation performance**

Outline

- Background on Collision Avoidance Systems
- System Specification & Algorithm Development
- Parallel Simulation Approach
- Performance & Simulation Results
- System Characterization
- Conclusions

Conclusion

- **Multi-Agent simulations 90x faster**
- **Fast-Playback simulations 30x faster**
- **End-to-end simulations 33x faster**
- **ACAS X algorithm specification with the Julia Programming Language allows rapid system prototyping and verification**
- **Simulation parallelization on the LLSC TX-Green cluster has enhanced system testing and characterization**