Application-Level Fault Tolerance as a Complement to System-Level Fault Tolerance

J. Haines, V. Lakamraju,
I. Koren and C. M. Krishna

Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003

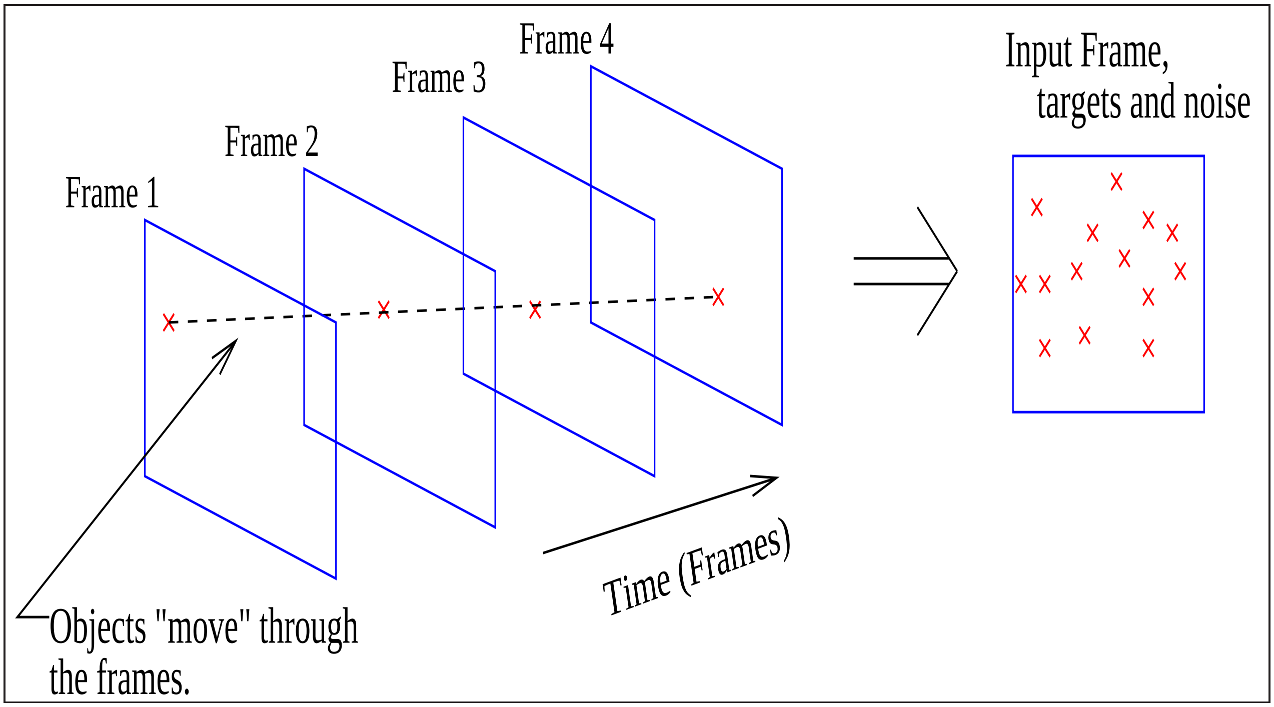
http://www.ecs.umass.edu/ece/realtime

Supported in part by DARPA, and the AFRL
under order number E349.

Introduction

- Fault tolerance in real-time systems may be incorporated at two levels:
  - ⋆ System-Level encompasses all redundancy of system components and recovery actions taken by the system.
  - ⋆ Application-Level encompasses redundancy and recovery actions within the application software itself.
- We have integrated our application-level fault tolerance technique with the Honeywell RTHT Target Tracking and ABF BeamForming Benchmarks.

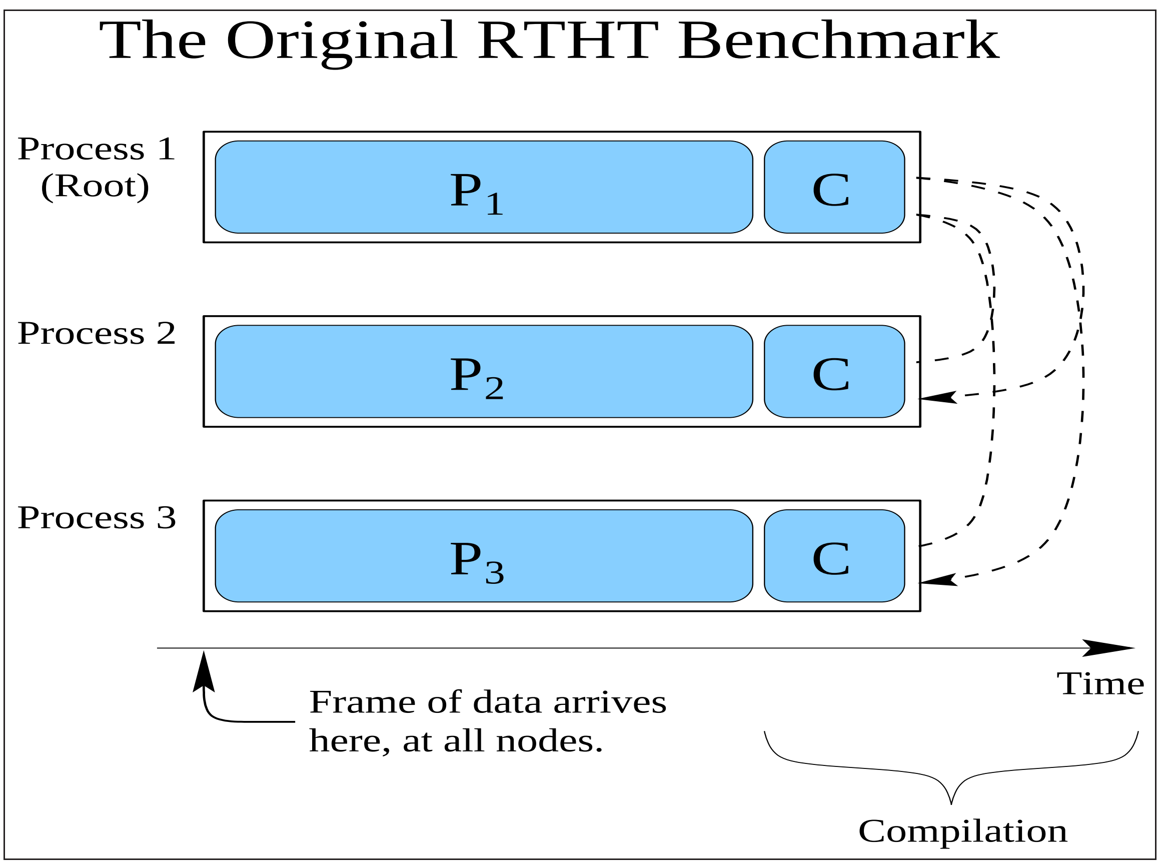
The RTHT Benchmark



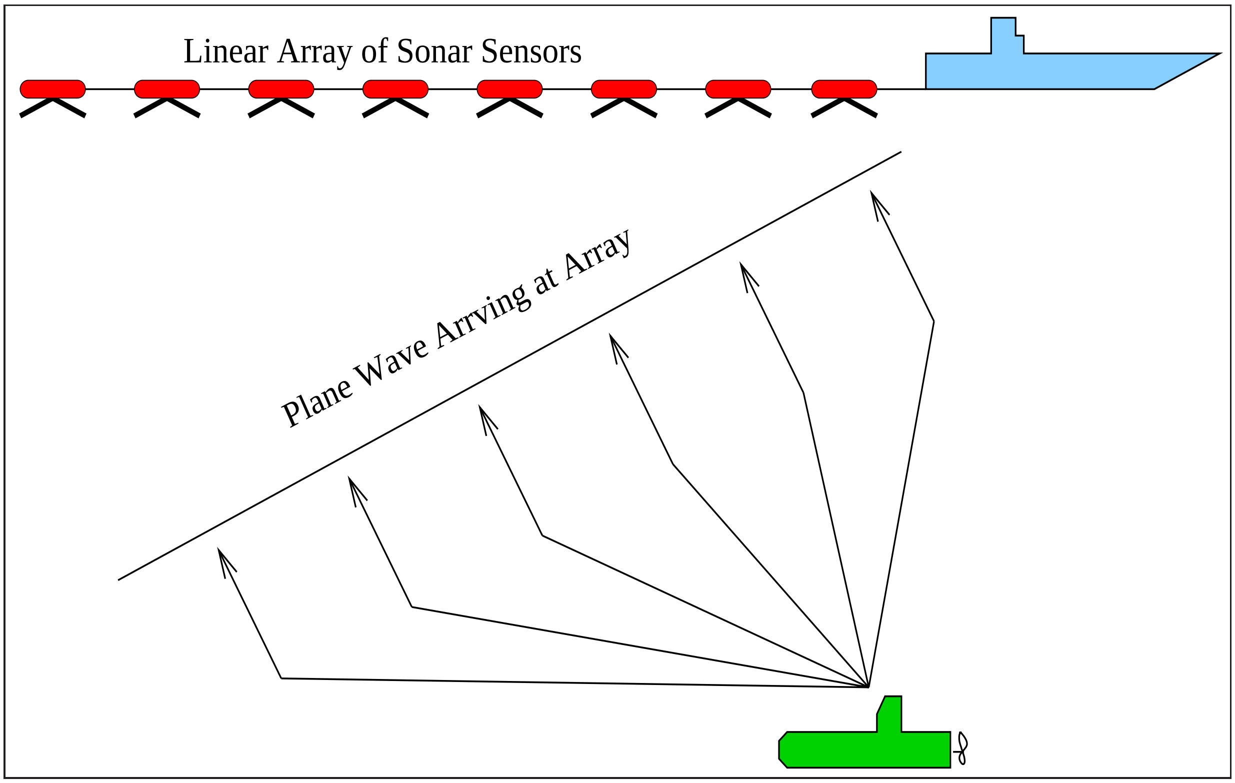
- The **R**eal-**T**ime **H**ypothesis **T**racking benchmark
- The **purpose** of the RTHT benchmark is to track a number of objects moving about in a 2d coordinate plane, as though the data is collected by radar.

RTHT Benchmark Cont.

- Each process works on a distinct subset of data from the radar, and tracks targets through the creation and extension of hypotheses which include a figure of likelihood.



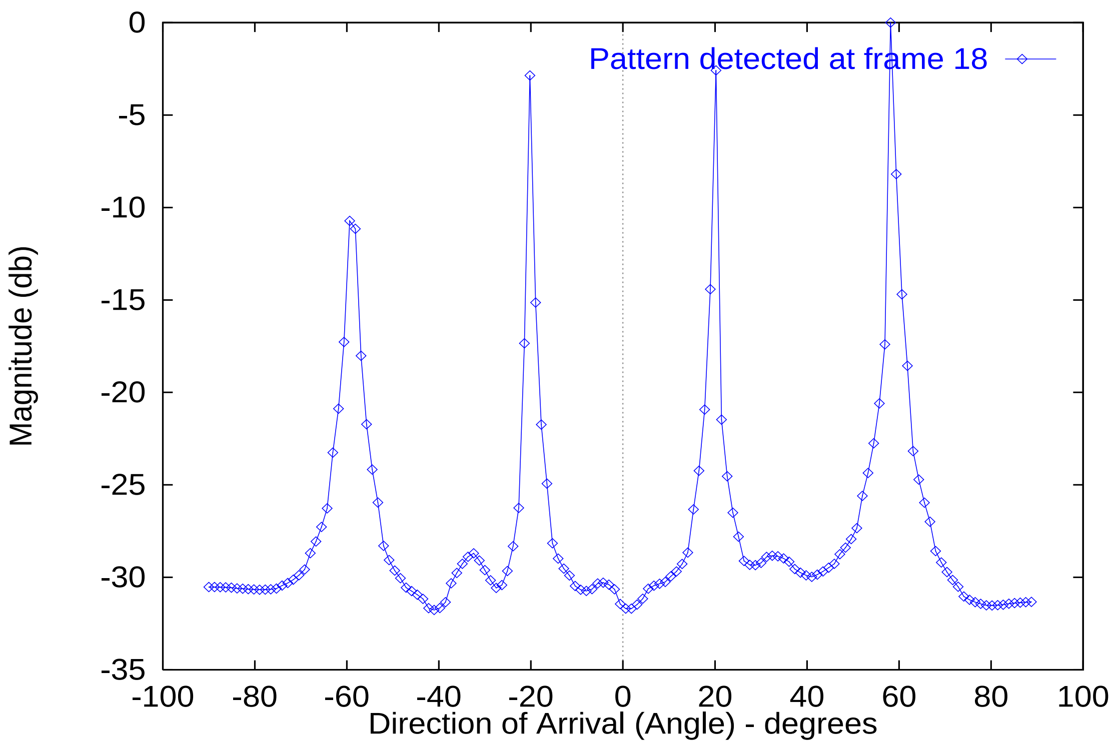
The ABF Benchmark



- The **A**daptive **B**eam**F**orming benchmark.
- The purpose of the ABF benchmark is to detect sound as it impinges on a linear array of sonar sensors.

ABF Benchmark Cont.

- Each process works on a distinct subset of the frequency range, in each frame it updates a set of weights.
- Applying these weights to the input samples has the effect of forming a beam which emphasizes the sound arriving from each direction.



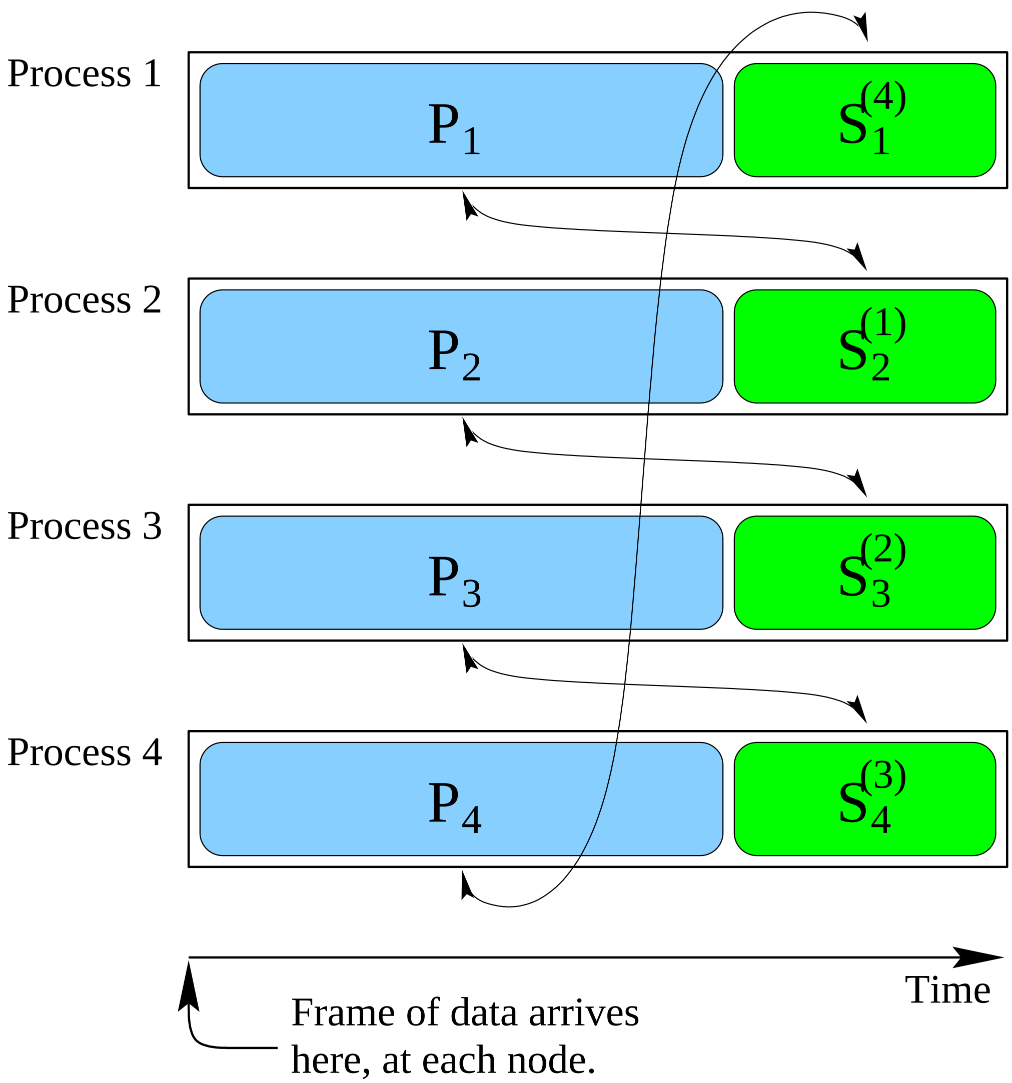
The Problem

- System-level fault tolerance alone does not provide the required reliability.
  - ⋆ Overhead associated with movement of large process checkpoints increases chances of missing a deadline.
  - ⋆ Significant recovery period could be required if checkpoint data can not be relocated in timely fashion.
  - ⋆ (RTHT) Cold-start is required if failure to participate in compilation occurs.
- Reliability is measured in terms of the number of targets (or beams) successfully tracked vs. the number of targets (or beams) that should have been tracked.

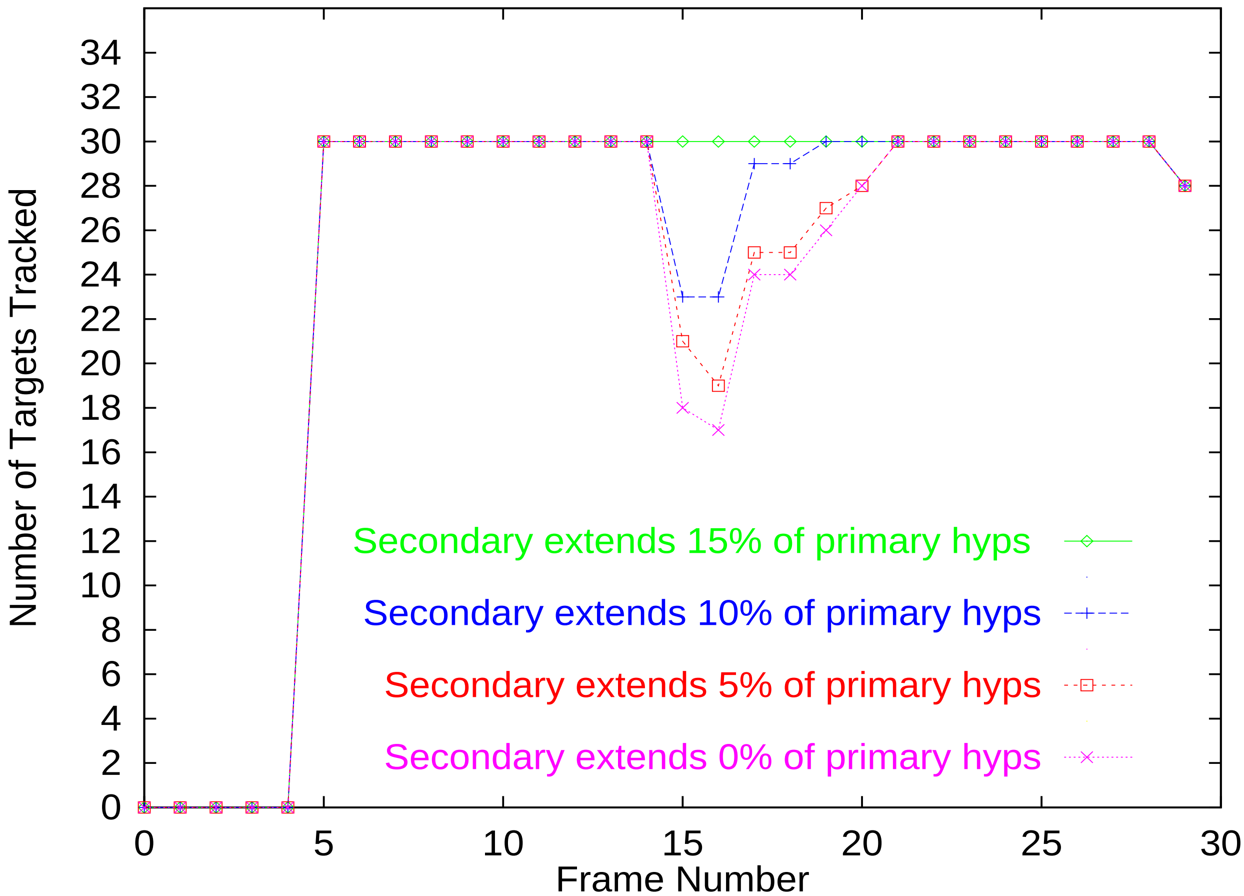
Our Technique

- We aim to bridge the gap with application-level redundancy when system-level fault tolerance is not able to provide the required reliability.
- Each process performs, in addition to its regular work, a piece of its neighbor's primary work.
- Upon detecting a faulty neighbor, a node makes use of both its primary and secondary data at the point in the frame when results are due.
- When rescheduled, the interrupted process begins calculations with data which its secondary has computed on its behalf.

Benchmarks with Application-Level FT



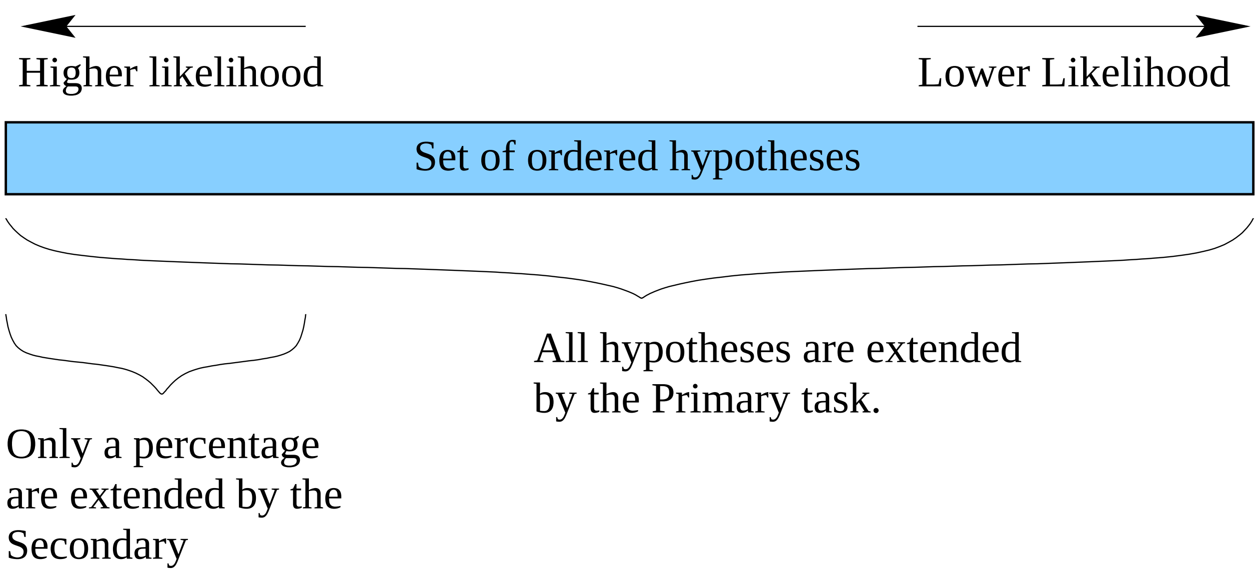
RTHT Results



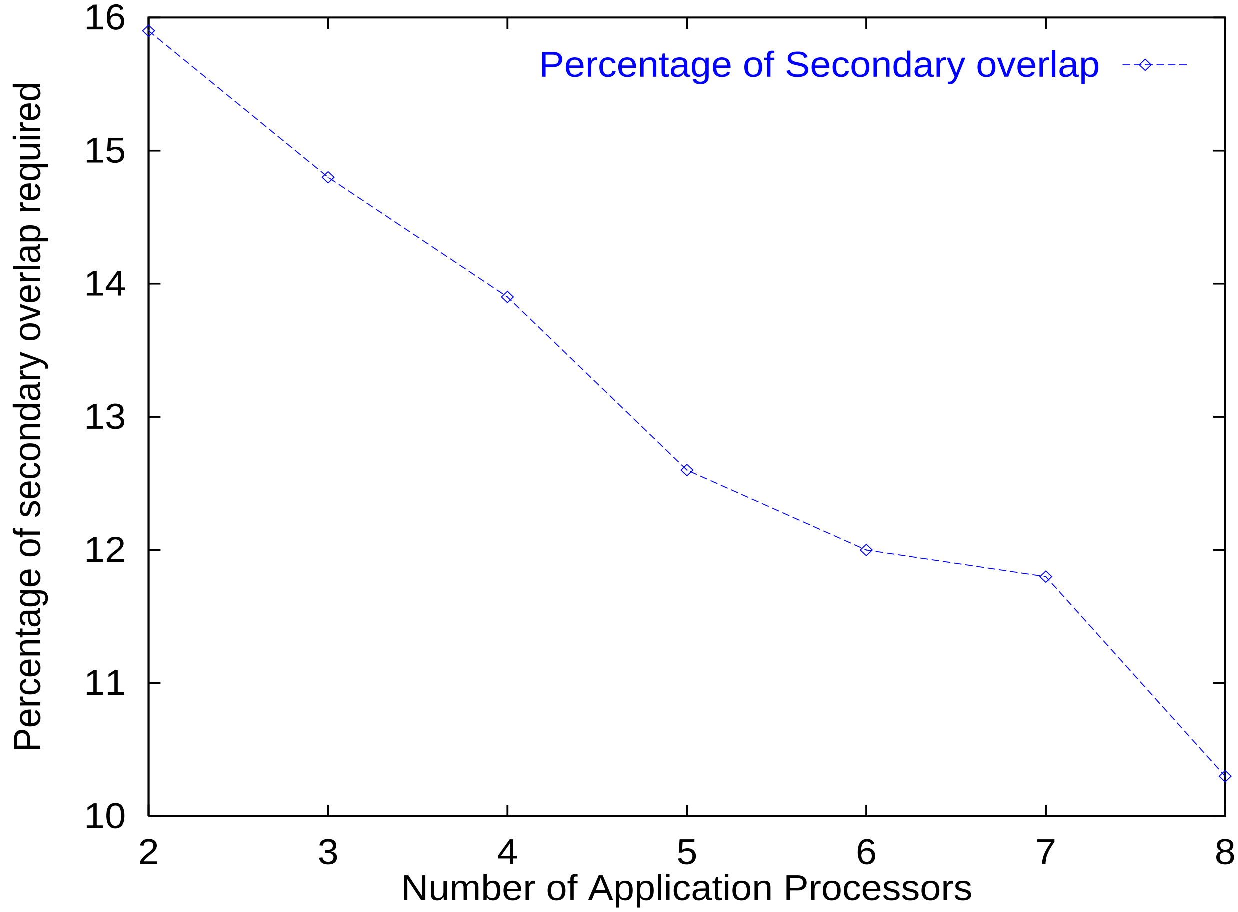
- **Results have been obtained with 30 real targets, roughly 80 false alarms and 2 application processes. A single fault, lasting one frame, occurs at frame 15.**
- **We see that a redundancy of only 15% is satisfactory to track all targets despite a fault at one node.**

RTHT Results Cont.

- A small amount of redundancy has a great effect on tracking stability in the presence of a failure.
- This is partly due to the fact that the hypotheses extended by the secondary are the ones most likely to be real targets, as the hypotheses are sorted in order of likelihood at compilation.

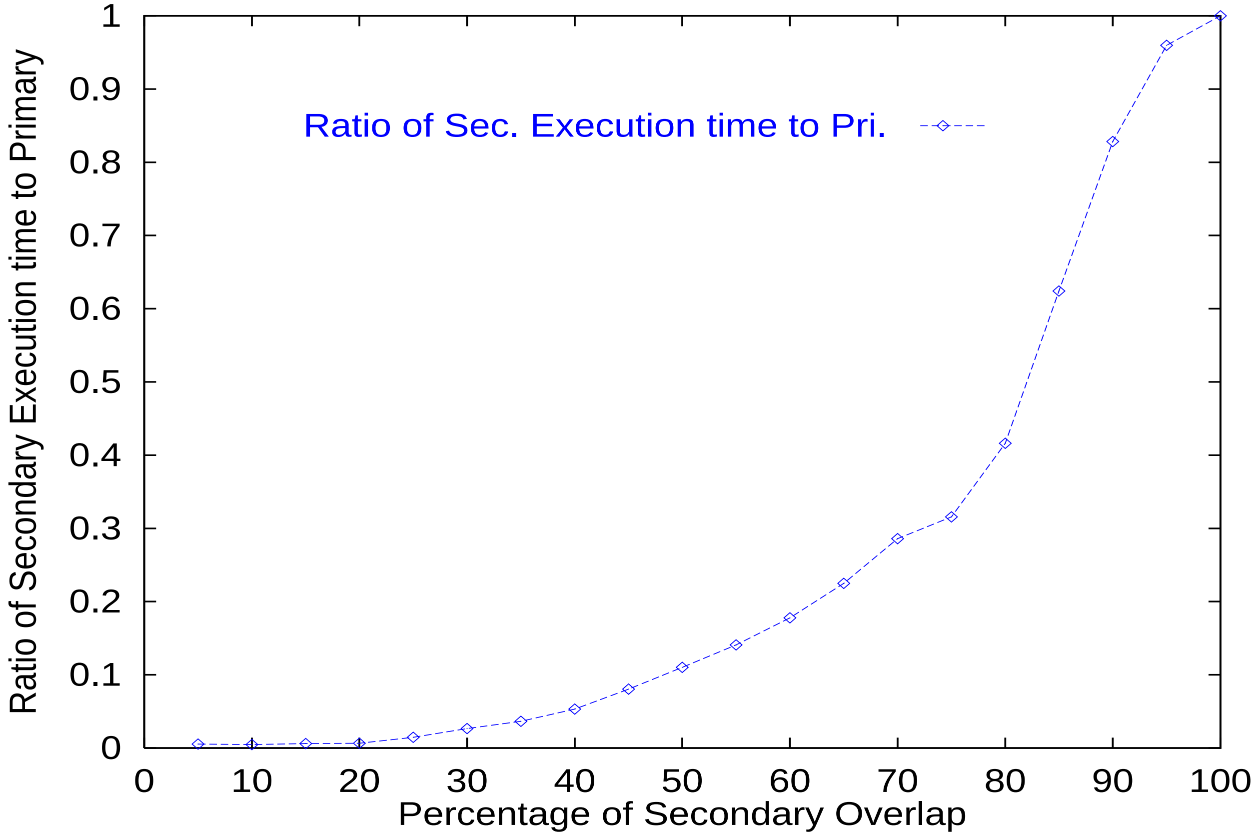


RTHT Results Cont.



- **The amount of redundancy required to achieve a given level of reliability decreases, almost linearly, as the number of application processes increases.**

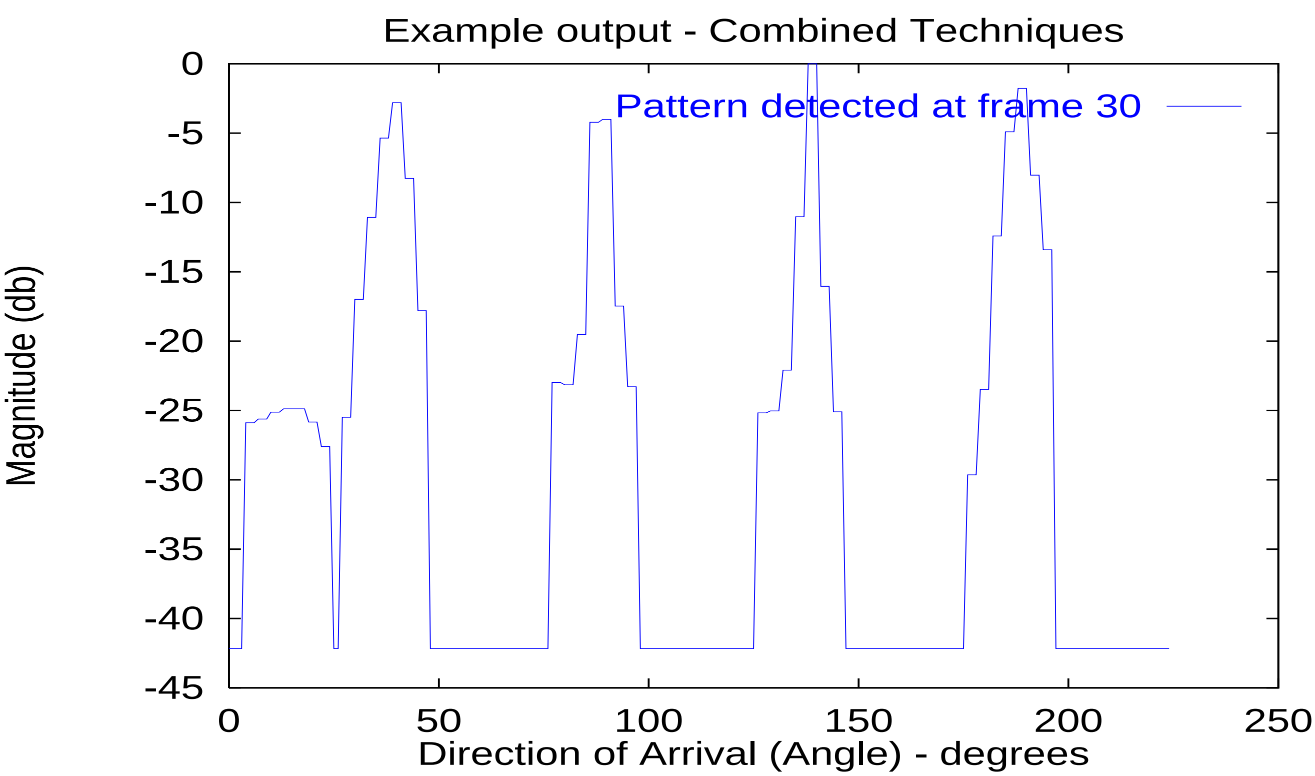
RTHT Results Cont.



- A proportionately small load is imposed on the processor by the computation of the secondary task set.
- This is due to the fact that the extension of hypotheses that are more likely to be real tracks, takes less time.

ABF Implementation Details

- Two methods of secondary overlap:
  - ⋆ **Limited field of view** - Secondary only searches in certain directions (windows).
  - ⋆ **Reduced granularity** - Secondary searches full field, with reduced granularity.
- Combination of both illustrated here.

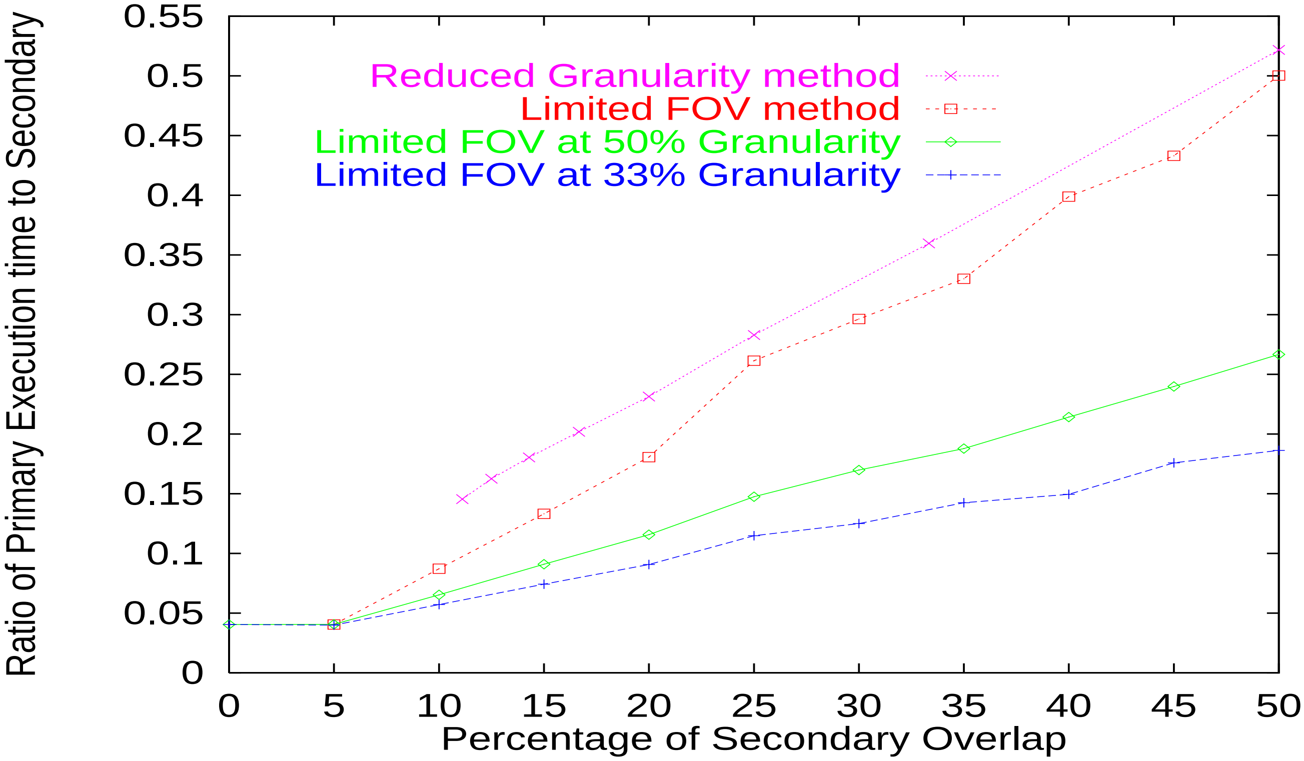


ABF Results

- Results were obtained with 4 beams of sound at each of 32 frequency ranges, and 2 application processes. A single frame fault occurs at frame 20.
- Combining the two greatly reduces the computational overhead, while maintaining similar results.
- Table shows minimum redundancy required to not lose track of any beams despite a fault in one process.

Redundancy Technique	Secondary Overlap	Computational Overhead
Reduced Granularity	33%	35%
Limited FOV	30%	30%
Combined - 30%FOV,50% Granularity	15%	17%

ABF Results



- **The ABF Benchmark's computational load curves are linear, unlike with the RTHT Benchmark, due to the ABF's uniform dataset.**
- **Still, a proportionally small amount of extra computation is necessary to mask the fault.**

Advantages of Our Technique

- The system's job during recovery is made easier: when a process has to be restarted on another node, the process' data segment no longer needs to be moved.
- In the case of transient failures, our technique provides leeway in the transfer of a task, in hopes that the fault will disappear.
- Provides a substantial improvement over complete system duplication.
- Secondary need not be cold-started: it is ready to go when the failure of the primary is detected.

Conclusions

- A high degree of fault tolerance may be obtained with a minimal investment of system resources, particularly in applications exhibiting data parallelism and a prioritized ordering within the data set
- This is achieved through a combination of application-level and system-level fault tolerance.

Future Work

- Analysis of use of other system resources, such as memory requirements.
- Development of a general set of guidelines to help application designers take better advantage of this technique.