

# A Cluster-Based Implementation of a Fault Tolerant Parallel Reduction Algorithm Using Swarm-Array Computing

Authors:

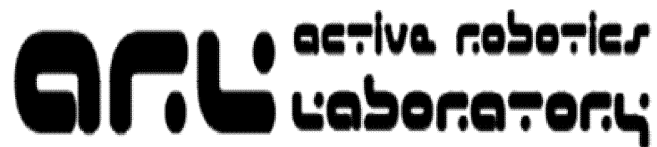
*Blesson Varghese, **Gerard McKee**, Vassil Alexandrov*

School of Systems Engineering

University of Reading, United Kingdom

E-mail: [b.varghese@student.reading.ac.uk](mailto:b.varghese@student.reading.ac.uk),

[g.t.mckee@reading.ac.uk](mailto:g.t.mckee@reading.ac.uk), [v.n.alexandrov@reading.ac.uk](mailto:v.n.alexandrov@reading.ac.uk)



# Table of Contents

- Introduction
- Swarm-Array Computing
- Proof of Concept
- Implementation
- Impact
- Conclusion

## A Cluster-Based Implementation of a Fault Tolerant Parallel Reduction Algorithm Using Swarm-Array Computing

Bhaskar Varghese, Gerard McKee and Vassil Alexandrov  
School of Systems Engineering  
University of Reading, Whiteknights Campus  
Reading, Berkshire, United Kingdom, RG6 6AY  
Email: b.varghese@students.reading.ac.uk, g1.mckee@reading.ac.uk, v.alexandrov@reading.ac.uk

**Abstract**—Recent research in multi-agent systems incorporates fault tolerance concepts. However, the research does not explore the extension and implementation of such ideas for large scale parallel computing systems. The work reported in this paper investigates a swarm array computing approach, namely Intelligent Agents. A task to be executed on a parallel computing system is decomposed to sub-tasks and mapped onto agents that traverse an abstracted hardware level. The agents inter-communicate across processors to share information and for the event of a predicted core/processor failure and for successfully completing the task. The agents hence constitute a swarm array fault tolerance and network building system. The feasibility of the approach is validated by simulation on an FPGA using a multi-agent simulator, and implementation of a parallel reduction algorithm on a computer cluster using the Message Passing Interface.

**Keywords**—swarm-array computing; intelligent agents; fault-tolerant system; cluster-based implementation;

### 1. INTRODUCTION

Fault tolerance is an important area of research in distributed systems. From a high level view, fault tolerance is necessary for reliable business systems, so as to continue operations even after system components have failed. From a low-level view, fault tolerance is necessary for distributed systems, so as to reduce impact of failure when it occurs or when it is predicted and seamlessly continue execution.

Researchers mainly identify two types of fault tolerance applicable to distributed systems [1]. Firstly, reactive fault tolerance that aims to reduce the impact of failure when it occurs in a distributed system [2] [3]. Secondly, proactive fault tolerance that aims to predict failures, and move executing tasks to safe nodes from nodes predicted to fail [3] - [6]. Hence, proactive fault tolerance policies aim for "controlling" a situation by causing something to happen rather than waiting to respond after it happens".

In recent times, multi-agent systems have incorporated concepts of fault tolerance. Research based on multi-agent fault tolerance can be separated into two categories, namely fault tolerance of a multi-agent framework and fault tolerance of an agent within a multi-agent framework.

In [7], fault tolerant multi-agent system characteristics are identified and a potential framework, namely Autonomous

Cooperative System (ACS), is discussed. Key concepts of the framework include reliable communication, fault-tolerant agent platform, fault-tolerant agent architecture, physical distribution and fault-tolerant control system is proposed.

In [8], a fault tolerant networked control system is proposed. Distributed artificial intelligence techniques, namely multi-agents are implemented for safe operations. The number of critical communications needed for safe operations between system components are minimized, hence guaranteeing safe operation and performance even in fault conditions.

In [9], plan-based fault tolerance of a multi-agent system is implemented by the Dynamic Agent Replication Extension (DAEX). In this model, fault tolerance is achieved by replicating those agents that are critical to the system and whose failure could influence other agents in the system.

In [10], fault tolerance of individual agents of a multi-agent system is presented. Fault-tolerance is achieved by exception handling and periodic events that are sent to agents to inspect their state, though an overhead to the system. The agent and broker is integrated in the architecture so as to also achieve fault tolerance in the tolerance system.

In [11], the individual agent fault tolerance is addressed. The prevention of a partial or complete loss of an agent in a system is achieved by employing an algorithm that is similar to the sliding window model [12].

A selected set of research based on multi-agent fault tolerance whose review is reported above, deals with the tolerance of the agent framework and agents within the framework. However, the research does not explore the extension and implementation of such ideas for large scale parallel computing systems. Hence, the question "How can a bridge between fault-tolerance in multi-agent systems and parallel computing systems be built?" arises, and is addressed in this paper.

The work reported in this paper is motivated towards simulating and implementing a "Swarm-Array Computing" approach based on intelligent agents on an FPGA, (Field Programmable Gate Array) and on a computer cluster respectively. A task to be executed on a parallel computing

# Introduction - 1

- Fault Tolerance
  - Important issue in distributed parallel computing systems
  - High level view
    - Reliable business systems – continue operation when system components have failed
  - Low level view
    - Reduce impact of failure when it occurs – seamlessly continue execution of a task

# Introduction - 2

- Two types of fault tolerance:
  - Reactive Fault Tolerance
    - Reduce impact of failure when it occurs
    - “Response after failure occurs”
  - Proactive Fault Tolerance
    - Predicts failures likely to occur
    - “Responding when a failure is likely to occur”
- Research in this paper focuses on *Proactive Fault Tolerance*

# Introduction - 3

- Modern day fault tolerance
  - Technology used – Multi-agent Systems
  - Classification of Multi-agent fault tolerance:
    - Fault tolerance of multi-agent framework
    - Fault tolerance of individual agent in the framework
- Existing research on multi-agent based fault tolerance does not explore the extension and implementation of such ideas for large scale parallel computing systems

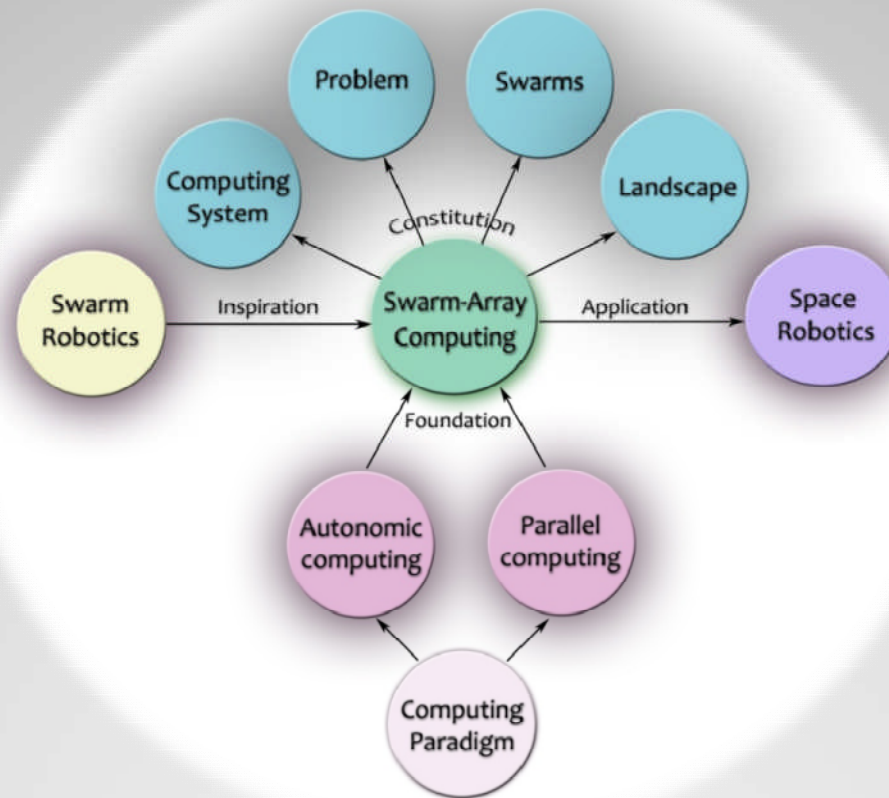
# Introduction - 4

- Question that needs to be addressed:

“How can a bridge between fault-tolerance in multi-agent systems and parallel computing systems be built?”

Hence, Swarm-Array Computing is proposed.

# Swarm-Array Computing - 1



# Swarm-Array Computing - 2

- Constitution
  - Computing Systems
    - Field Processing Gate Arrays (FPGAs) and Computer Clusters used in this study
    - Cores can be considered as 'intelligent cores'
  - Problem/Task
    - A Task to be executed can be considered as a swarm of autonomous agents
    - Tasks can be considered as 'intelligent agents'



# Swarm-Array Computing - 3

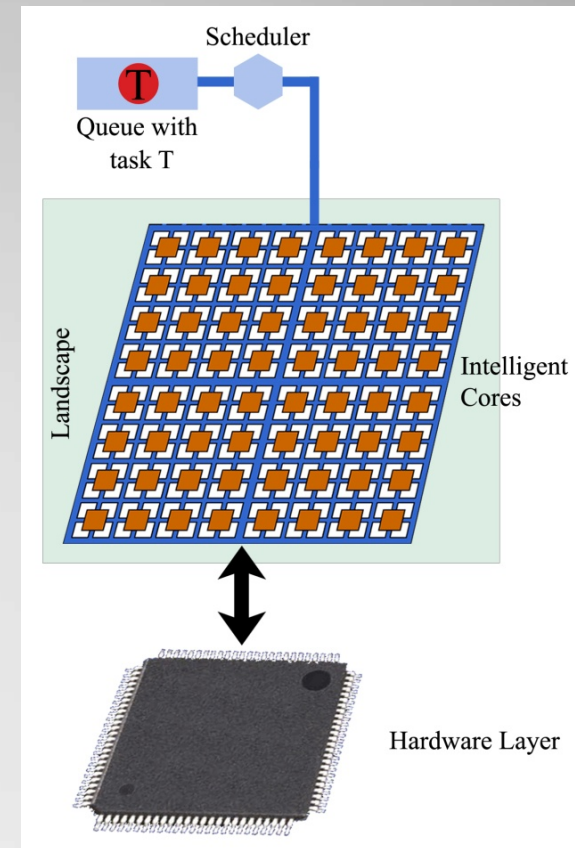
- Constitution (contd.)
  - Swarms
    - Combination of Intelligent Cores and Intelligent Agents
  - Landscape
    - Arena in which cores and agents interact with each other
    - Defines the state of the computing system and the task being executed

# Swarm-Array Computing - 4

- Approaches
  - Fits the Swarm-Array computing constituents together
  - Three approaches
    - First Approach - Intelligent Cores
    - Second Approach - Intelligent Agents
    - Third Approach - combinative approach considering both Intelligent Cores and Intelligent Agents

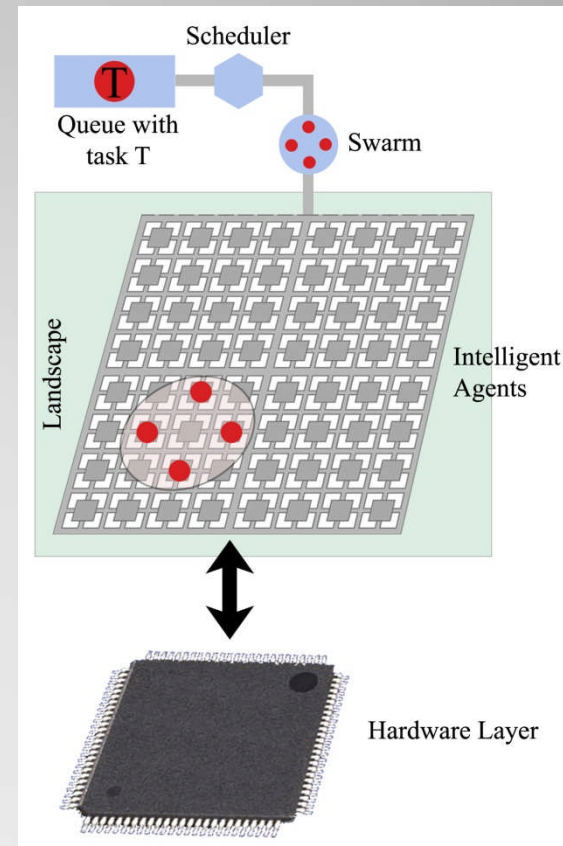
# Swarm-Array Computing - 5

- Approaches (contd.)
  - First Approach - Intelligent Cores
    - Hardware abstracted to intelligent cores
    - On the event of a failure, tasks can get transferred from one core to another
    - Landscape – the arena on which the task gets executed



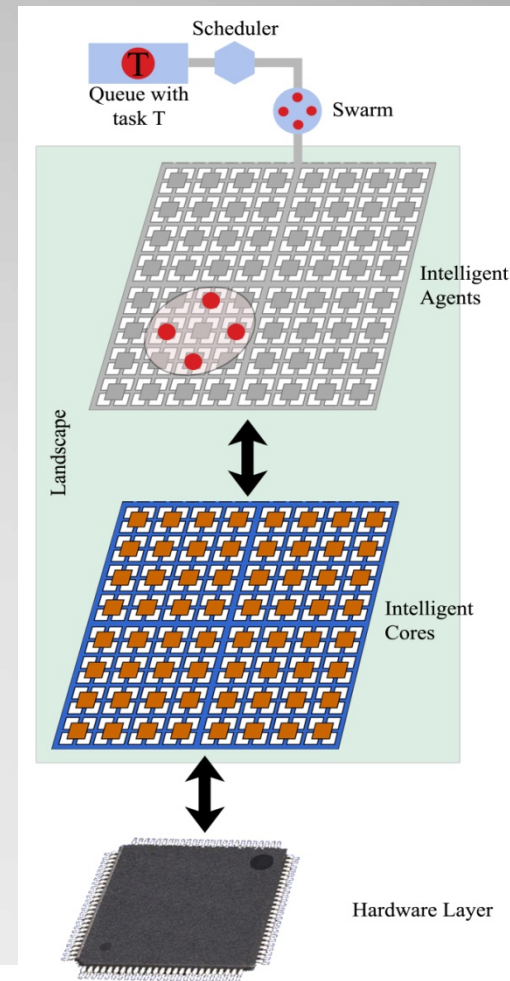
# Swarm-Array Computing - 6

- Approaches (contd.)
  - Second Approach - Intelligent Agents
    - Hardware layer abstracted
    - Tasks mapped onto autonomous swarm agents
    - On the event of a node failure, agents move from one core to another
    - Landscape – the arena on which the agents traverse
- Intelligent Agent based approach considered in this paper



# Swarm-Array Computing - 7

- Approaches (contd.)
  - Third Approach - Intelligent Cores and Intelligent Agents
    - Combination of the first and second approach



~Swarm-Array Computing~  
Presented by: B. Varghese

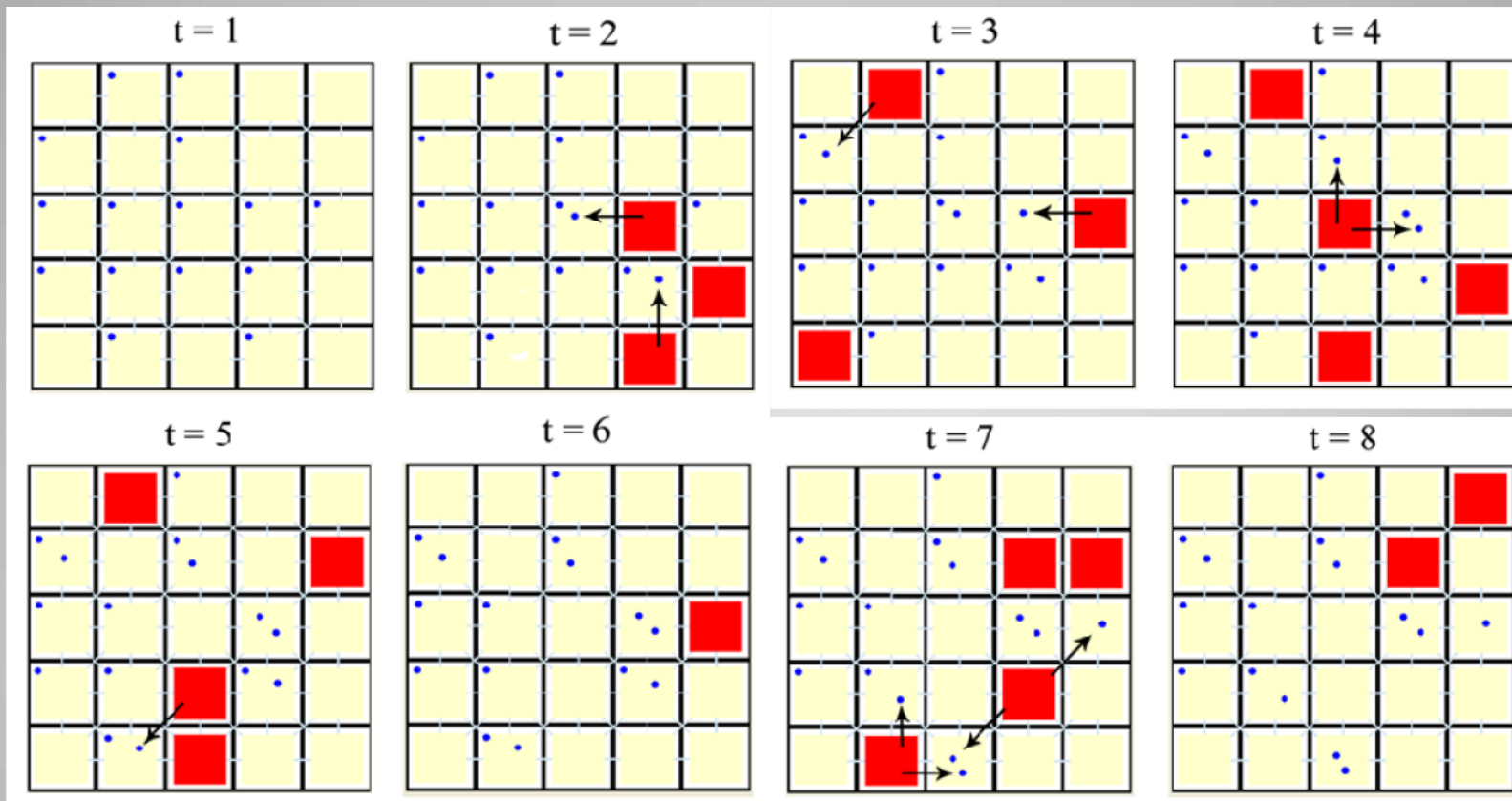
# Proof of Concept - 1

- Experimental Environment
  - Multi-agent simulator the best option
  - SeSAM (Shell for Simulated Agent systems) simulator
    - Provision for modelling agents, world and simulation runs
- Modelling
  - The cores of the FPGA modelled as agents
  - 5 X 5 regular grid FPGAs considered

# Proof of Concept - 2

- Modelling (contd.)
  - Core temperature simulated
  - Approach 2 – Intelligent Agents
    - When core temperature increases beyond a threshold, the **agent executing on a core** moves to another core

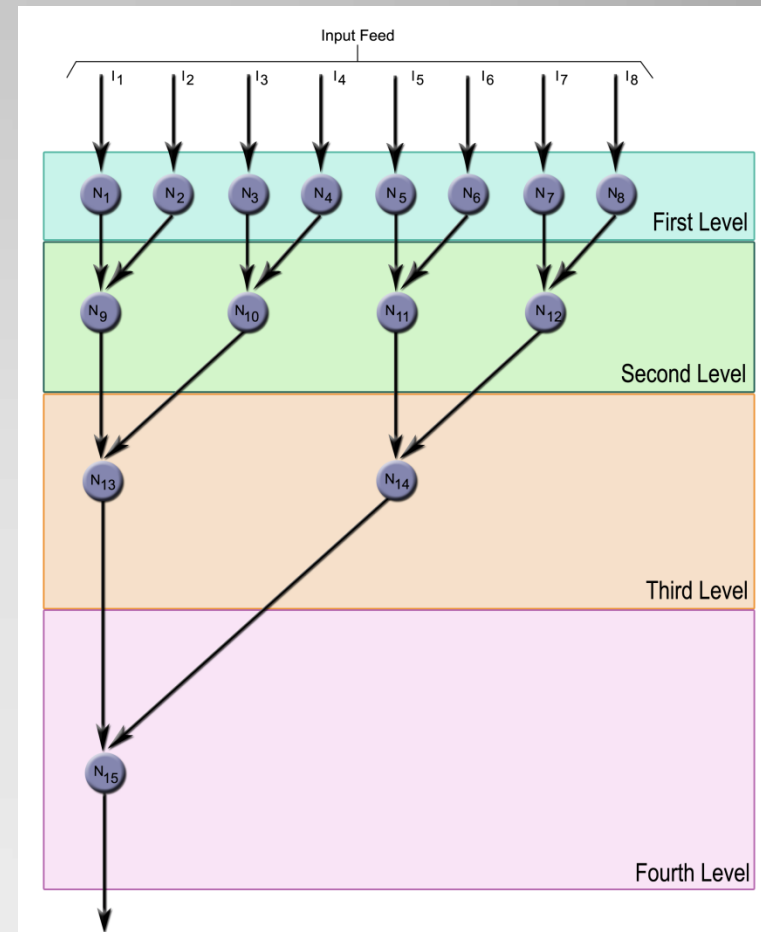
# Proof of Concept - 3





# Implementation - 1

- What tasks can benefit from Swarm-Array Computing?
  - Parallel Reduction Algorithms
    - the computing nodes of a parallel reduction algorithm tend to be critical
    - employed in critical applications such as space applications



# Implementation - 2

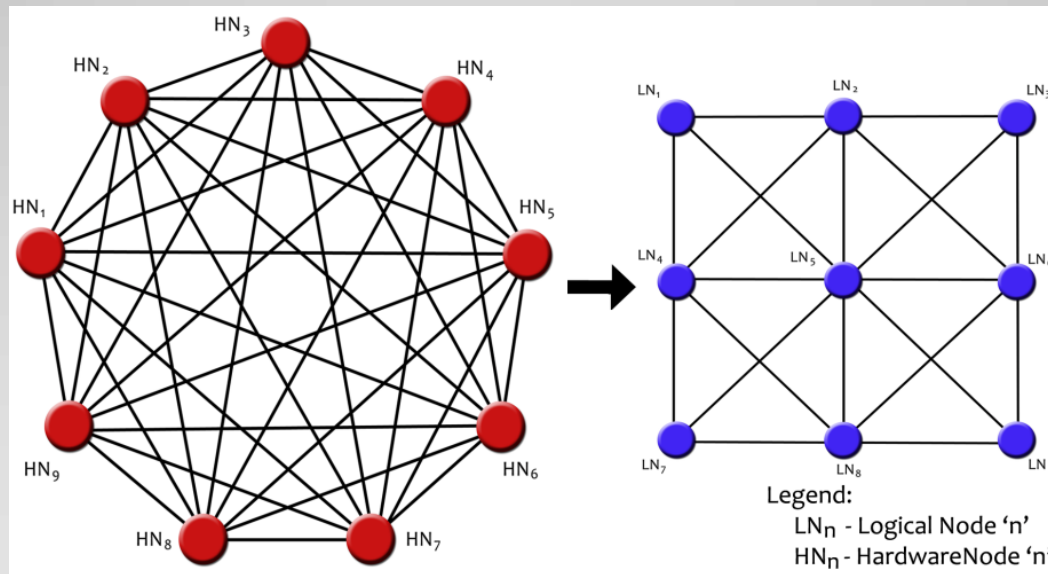
- Resources:
  - ACET Teaching Cluster used as computing platform
  - 1 head node and 33 compute nodes
  - Connected via the Gigabit Ethernet switch
  - All communications through TCP
- Middleware
  - Open MPI 1.3.3, open source implementation of MPI (Message Passing Interface) version 2.0
  - Supports Dynamic Process Creation and management

# Implementation - 3

- Two Parallel Reduction Algorithm Implementations:
  - Classic Version
    - No fault tolerant concepts
    - If used in critical versions would stall the algorithm
  - Fault Tolerant Version
    - Implemented using 'Intelligent Agents' in Swarm-Array Computing

# Implementation - 4

- Landscape:
  - Rules / Policies for abstraction
  - Hardware nodes abstracted to logical nodes



# Implementation - 5

- Each process executing on a node gathers some sensory information
  - Prediction on whether a node is likely to fail
  - Similar to proactive fault tolerance.
- Node temperatures simulated
  - When the temperature of a node rises beyond a threshold, the process executing on that node predicts a failure
  - Spawn a new process on an adjacent core in the abstracted layer.
- The agent on the abstracted core expected to fail shifts to the adjacent core on which the new
- Dependency information carried by the agent that was shifted to the new core is employed to reinstate the state of execution of the algorithm.
- Ensures that information is not lost and does not affect the final solution in critical applications.

# Impact

- Useful for space applications
  - Space crafts employ FPGAs
  - When space craft leaves the atmosphere, Single Event Upsets (SEUs) likely to occur due to radiations
  - Hardware reconfiguration or software uploading from earth extremely impossible
  - Hence self-managing approach required
  - Swarm-Array Computing can come to play

# Conclusion

- `Intelligent Agent' approach in Swarm-Array Computing considered
- Proof of concept validated on a multi-agent simulator
- Implementation on the ACET teaching cluster using Open MPI
- Two implementations – classic vs fault tolerant
- Traditional Fault Tolerant methods can be replaced

**Thank you for your  
undivided attention**

