IAS - Group - 7 (Team - 3)
Design Document

Monitering & Fault Tolerance and Node Management Module

Aditya Rathi (2020201041), Punit Sharma (2020201029), Yagyesh Purohit(2020201013)

May 2021

Contents

1 Nodes and Node Manager 1

1.1 Description . . . 1

1.2 Flow chart . . . 1

1.3 Subsystems . . . 1

1.4 Functional Requirements . . . 1

1.5 Interaction with other modules . . . 2

1.6 Registry . . . 2

1.7 Test cases . . . 2

2 Monitoring and Fault Tolerance 3

2.1 Description . . . 3

2.2 Flow chart . . . 3

2.3 Subsystems . . . 3

2.4 Functional Requirements . . . 4

2.5 Interaction with other modules . . . 4

2.6 Registry . . . 5

2.7 Test cases . . . 5

List of Figures

1 Node Manager module . . . 1

2 Monitoring and Fault Tolerance . . . 3

1 Nodes and Node Manager

1.1 Description

This module is used to manage and allocate new node (virtual machine) as and when required. It provides a node details along with node id, when a node request is made.If available is unable to satisfy the request a new node is created.

1.2 Flow chart

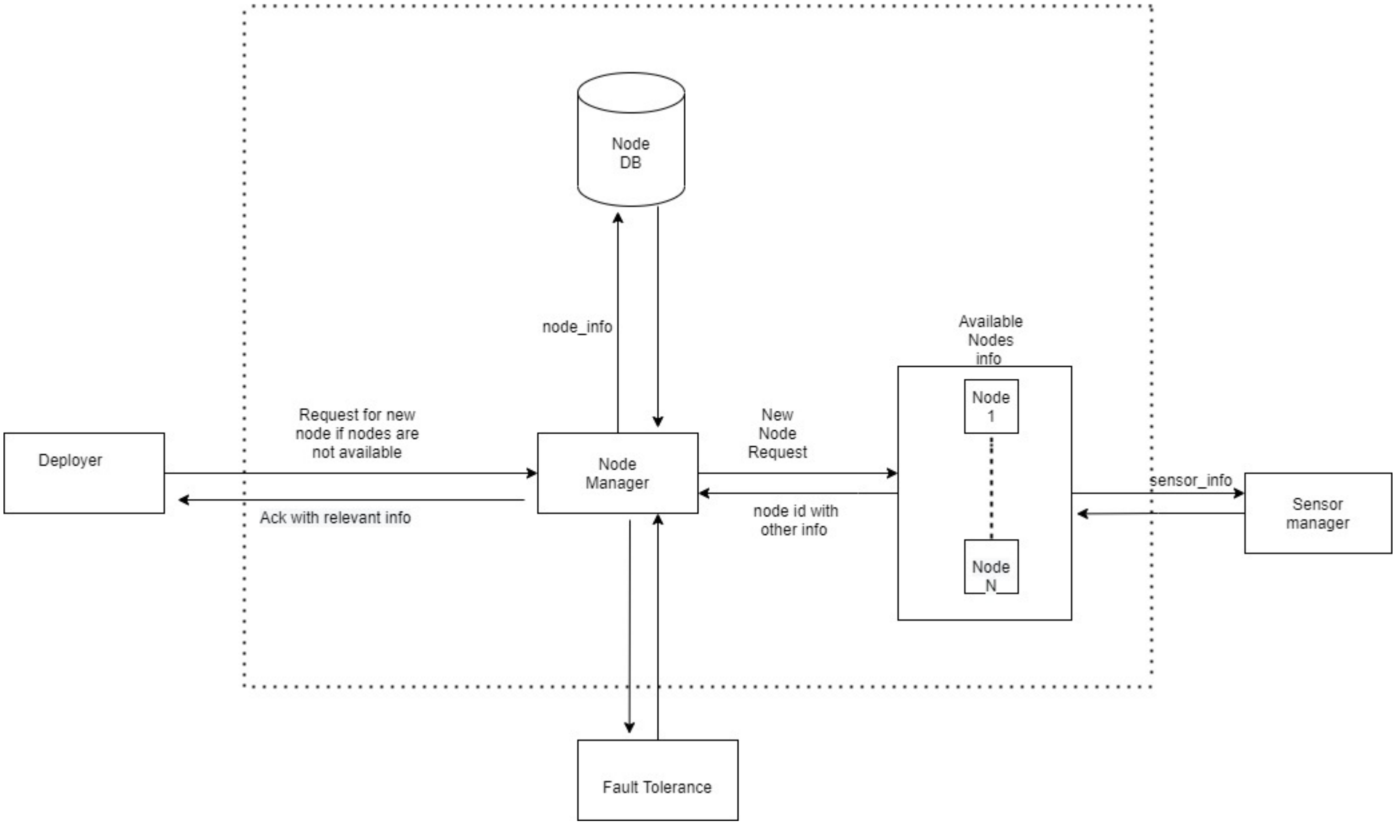

Figure 1: Node Manager module

1.3 Subsystems

- **Nodes -** Nodes are the entities where execution of an application instance or a platform component takes place. This may be a physical server or a virtual machine. The nodes may also communicate with the sensor manager for sending/receiving data to/from sensor(s).
- **Node Manager -** Node Manager module is responsible for managing the node information and providing the deployer a new node if needed.

1.4 Functional Requirements

- Providing a new node to the deployer when required.

- Providing a list of active nodes if required.
- Maintaining the node registry.
- Shutdown a node.

1.5 Interaction with other modules

- Sensor Manager - An application instance running on a particular node may wish to communicate with the sensor manager to send/receive data stream to/from a particular sensor or multiple sensors.
- Deployer - The deployer would request the node manager for a new node in case the load on present active nodes increases.
- Monitoring and Fault tolerance - Sends the heart beat message to the monitoring and Fault tolerance module.

1.6 Registry

Registry will be used to store several run-time related information for applications and platform modules.

- Number of nodes available
- Unoccupied node id list
- Per node information
  - Node id
  - Node status - occupied/unoccupied.
  - Type of service running - platform module/ application.
  - Applications
    - Application id
    - Application status
    - Sensor ids
      - sensor id1
      - sensor id2

1.7 Test cases

- **Input**: Deployer asking for execution of an algorithm when the available nodes are occupied.
  Output: Return the new node id to the deployer
- **Input**: Application instance requiring data from 4 sensors.
  Output: Pass the application id and corresponding sensor ids from the node (on which that application instance is running) to the sensor manager. Then receive the data from the sensor manager and pass it to that application.

2 Monitoring and Fault Tolerance

2.1 Description

This module continuously monitors the components for failures and allows the system to operate properly, even in case of failure of one or more of its components.

It monitors the platform modules/application instances and nodes using heart beat messages. It continuously checks if the module/application and node is working properly or node. If it finds that some node is not working properly, then it initializes a new node along with the applications running on it, else it reinitializes the application on the same node.

2.2 Flow chart

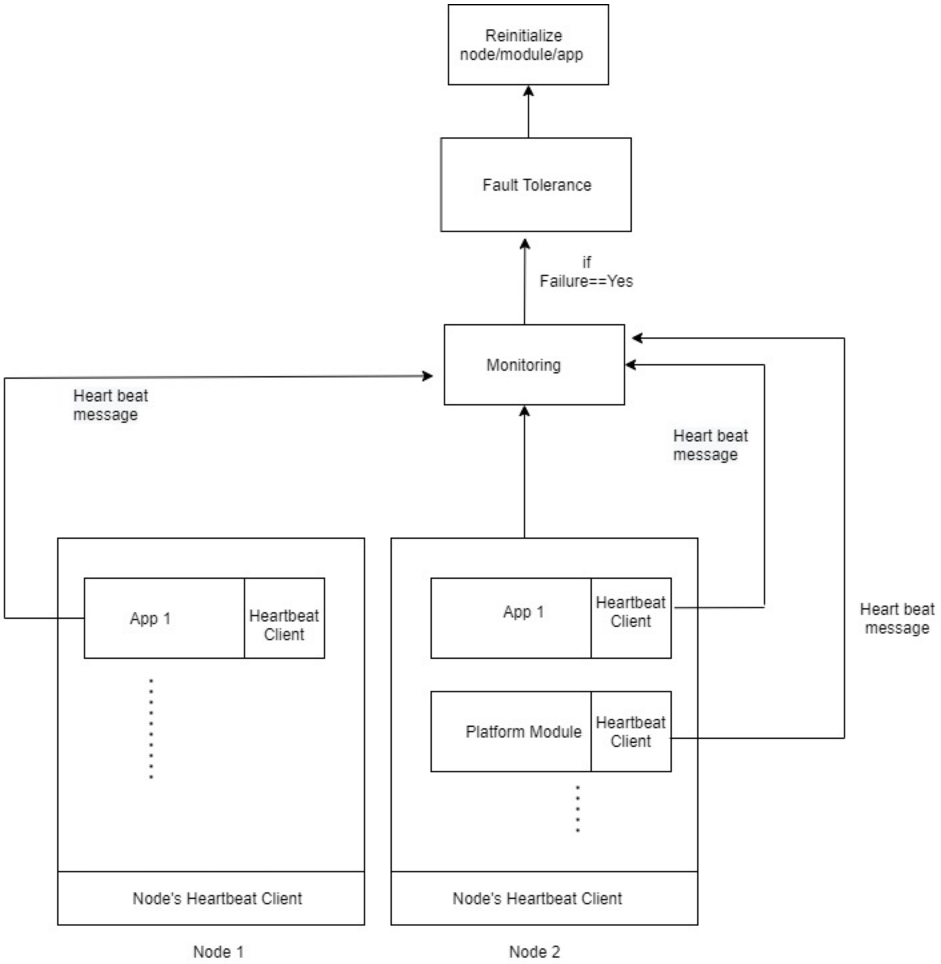

Figure 2: Monitoring and Fault Tolerance

2.3 Subsystems

This module includes the following sub-systems:

- **Monitoring** - It continuously receives the heart-beat message from the applications and node. If it doesn't receive a heart beat from a module or node for some time (say t - threshold) then it assumes that the application/ node is down, and conveys this message to the fault tolerance sub-system.
- **Fault Tolerance** - When it receives information of a module/node being down, it reinitializes it using its init configuration details.
- **Heartbeat Client** - It is associated with every module and node. It sends a heart - beat message to the monitoring service.

Heart-beat Format

- Type - 1 From the application/ module.
  1 node id application id
- Type 2 - From the node
  2 node id

2.4 Functional Requirements

- **Platform** - To check whether all the components (like scheduler, deployer, application manager etc) of the platform are working properly, and in case of failure reinitialize the component.
- **Application** - To check if the application instances are working properly or not. If they are not working then re-initialize another app instance.

  If some active node gets down, then it reinitializes all the applications/modules running on the node on some other node.

2.5 Interaction with other modules

The monitoring and fault tolerance module interacts with the following modules:

- Scheduler
- Sensor Manager
- Load balancer
- Deployer
- Application Manager
- Node Manager
- Nodes and application instances

For all the aforementioned modules, periodic monitoring for their proper functioning will be done. If any discrepancy or fault is detected in the working of these modules, the fault tolerance module's instance related to that module gets triggered, and an alternate node is allocated to run that module.

2.6 Registry

The registry contains the following information:

- Node info
  - Node id
  - Node status - 0 (failure), 1 (running)
  - Node failure count
  - Application details
    - App/module id
    - App status - 0 (failure), 1 (running).
    - Failure count

2.7 Test cases

- **Input** : node failure (did not received node's heartbeat message)
  Output : A new node with all the previous applications running.
- **Input** : Application/module failure (did not receive application/module heartbeat message)
  Output : A new instance of the application running on the same node.