Hierarchical Deadlock Detection

→ sites are logically arranged in hierarchy.

→ a site is responsible for detecting deadlocks involving only its children sites.

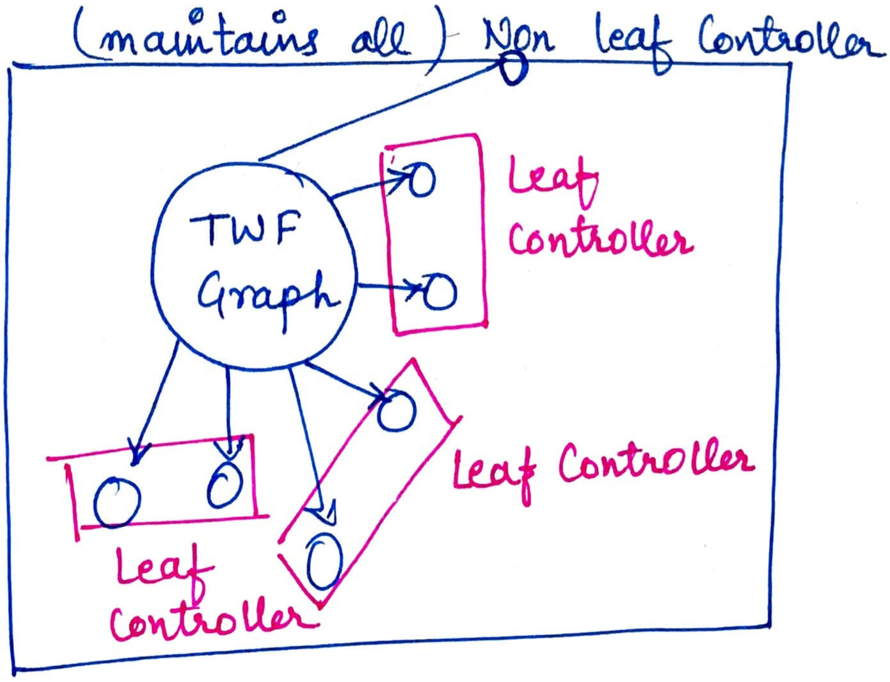
1. The Menasce – Muntz Algorithm

→ Controllers are arranged in tree fashion.

What is a Controller? It manages a resource or is responsible for deadlock detection.

Controller → Bottom most level → called leaf Controller → manage resources & other → deadlock detection

Controller → others → called non leaf controller



→ Leaf controller maintains a part of the global TWF graph concerned with the allocation of the resources at that controller

→ Non leaf controller - maintains all TWF graph & responsible for deadlock detection including leaf controllers.

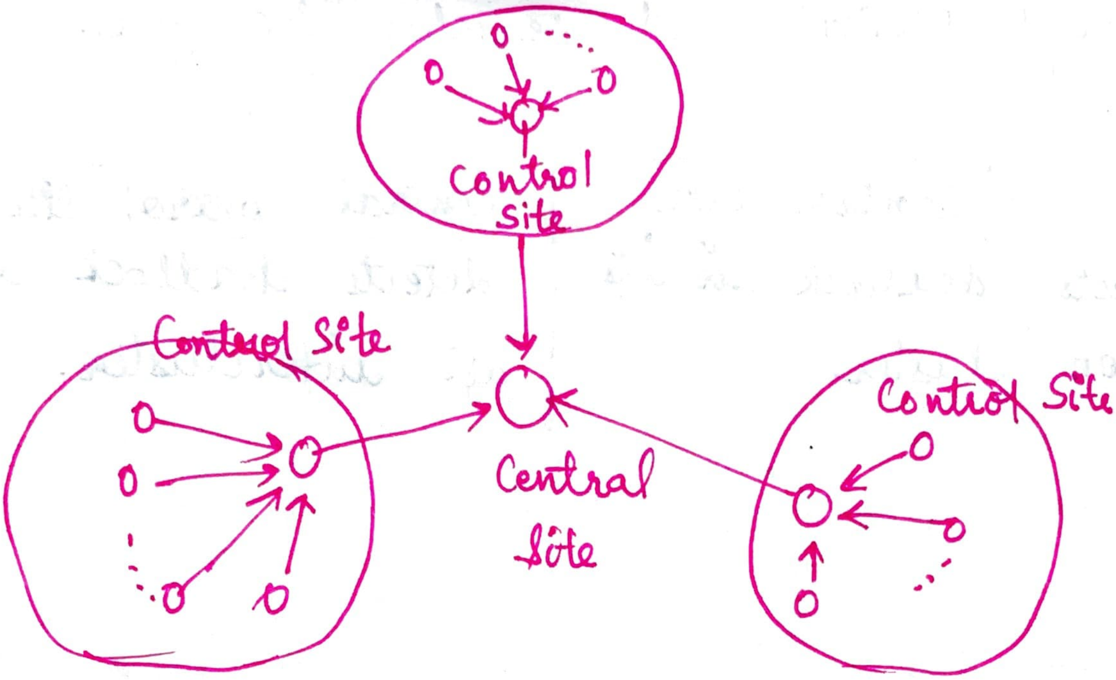
→ If any change occurs in TWF graph = due to resource allocation, wait, release

⇒ propagated to parent controller

→ Parent controller searches for cycle & forward information upward as well as non leaf controller is updated continously for its children.

Hierarchical Deadlock

2. The Ho- Ramamoorthy Algorithm



→ sites are grouped into several disjoint clusters

→ Periodically, a site is chosen as Central Control Site, which dynamically chooses a control site for each cluster.

→ Central control site requests from every control site their a) intercluster transaction status information
b) wait for relations

→ Control Site ⇒ collects status tables from all sites in its cluster & applies one phase deadlock detection algorithm to detect all deadlocks involving only intracluster transacti

→ It sends this information with wait for relations data to Central Control Site.

→ Central control site ⇒ constructs Wait for Graph (WFG) and searches for cycle.

Control Site	Central Control Site
detects deadlock in its own cluster.	detects deadlock in all intercluster.