SUMMARY – JICOS : A JAVA-CENTRIC NETWORK COMPUTING SERVICE

Introduction:

JICOS is a Java-centric network computing service that supports high-performance parallel computing. It supports scalable and adaptively parallel computation, tolerates basic faults and hides communication latencies. It provides an API well suited for solving divide and conquer problems and also for problems falling under branch and bound strategy. JICOS API for Branch and bound is different from Master- Slave Pattern in 3 ways:

(i) No fixed depth of task Decomposition

(ii) Unbalanced Task Decomposition tree

(iii) Intelligent pruning to reduce unwanted computations

Architecture:

JICOS comprises of 3 service components namely

(i) **Hosting Service Provider (HSP) :**

An interface for clients to login, submits tasks, requests results and logout.

(ii) **Task Server:**

It stores and distributes the ready tasks evenly to the hosts and includes fault-tolerant mechanism to reassign tasks to different hosts during failures.

(iii) Host:

Host is a compute server node that dynamically joins a task server and request for tasks.

Hiding Communication Latency:

JICOS provides application controlled directives to improve performance by hiding or overlapping computation time with communication time. Three techniques are as follows:

(i) Task caching

Caches first constructed subtask to reduce communication latency.

(ii) Task pre-fetching

Requests another task in a separate thread when atomic task is being executed.

(iii) Task server computation

Computing smaller tasks in task server rather than sending it to hosts.

Observations:

All the above optimizations will reduce delay associated with Host – TaskServer communications. Placing Task server on the same machine as compute server substantially reduces communication time and improves performance because of its simple compose tasks. However overhead is involved in reassigning tasks from faulty hosts, recognizing machine faults etc;

Experiments and Future Work:

Taskserver and compute server in same machine gives 3114.8 sec average time compared to 31115 sec. Both shows 99.7 % ideal speedup. Every host is given a lease and whenever a host's lease expires, all its tasks are given back to space thus tolerating faulty hosts. It uses Cilk like mechanisms and supports immutable shared input object and mutable shared object which serves as lower bound. Future work includes dynamically figuring out how much hosts a task server can serve by finding average task execution time overheads. Experiments show that JICOS is efficient due to space based architecture, explicit continuation passing making it an efficient distributed system.