Agent-Based Grid Load Balancing Using Performance-Driven Task Scheduling

Presented by: Xiaolong Jin


Performance-Driven Task Scheduler

• Resources
  – A grid resource (node) $P$ with $n$ processor
  – Resource model $\rho_i$ describes perf. info. of processor $p_i$

• Tasks
  – $m$ parallel tasks $T_j$ to be run on $P$
  – Application model $\sigma_i$ describes perf. related info. of tasks $T_i$
  – $\delta_j$ is the deadline requirement of task $T_j$

Performance-Driven Task Scheduler (Cont.)

• Schedule
  – A schedule is a set of $\bar{p}_j \subseteq P$ ($\bar{f}_j \subseteq \rho$) allocated to task $T_j$, and a set of start time $\tau_j$
  – The execution time for task $T_j$ is a function $t_e(\bar{p}_j, \sigma_j)$
  – The completion time is: $\eta_j = \tau_j + t_e(\bar{p}_j, \sigma_j)$
  – Makespan $\omega$ of a schedule: $\omega = \max_j \{ \eta_j \}$, i.e., the latest completion time of any task

  Scheduler goal:
  • Minimize $\omega$, and
  • $\forall j, \eta_j \leq \delta_j$
Performance-Driven Task Scheduler (Cont.)

- Genetic algorithm
  - Coding:
    - string $S_k$
    - task ordering
    - ordering part
    - mapping part

�

Scheduler Implementation

Request Info. → Task Management → Task Execution → GA Scheduling → Application Models → PACE Evaluation Engine

Results → Resource Info. → Resource Monitoring → Resource Info. → Evaluation Results → Resource Models

Agent-Based Grid Load Balancing System

User Portals → Application Tools

Agent Hierarchy → Evaluation Engine

Local Schedulers → Resource Tools

Agent-based grid load balancing → PACE performance prediction
Agent-Based Grid Load Balancing System (Cont.)

- Each resource is managed by an agent coupling a GA-based scheduler
- All agents are organized into a hierarchical structure
- Service discovery: After a task is submitted to a resource
  - If the service can match the task requirement, the discovery ends successfully. Otherwise,
    - The corresponding agent evaluates the service information of upper and lower agents, and passes the task to the one that provides the best requirement/service match.
    - If no service can match, the task is submitted to the upper agent.
    - If the task reaches the head of the agent hierarchy and does still not find a matched service, the discovery terminates unsuccessfully.

System Performance Metrics

- Average advance time \( \bar{e} = \frac{\sum_{i=1}^{M} (\delta_j - \eta_j)}{M} \)
- Average resource utilization rate \( v = \frac{\sum_{j=1}^{N} v_j}{N}, \quad u_i = \frac{\sum_{j \in F_i} (\eta_j - \tau_j)}{t} \)

Where \( v_i \) is the resource utilization rate of processor \( P_i \)
- Load balancing level \( \beta = 1 - \frac{d}{v}, \quad d = \sqrt{\frac{\sum_{i=1}^{N} (v - u_i)^2}{N}} \)

Remarks

- Load balancing is addressed at a lower/local level. The proposed mechanism cannot guarantee global load balancing.
- In a grid resource, tasks are scheduled in a batch. (Say, a batch contains 6 tasks)
  - The scheduler has to wait until 6 tasks are submitted?
  - During the process of scheduling, processors are idle?
- An agent at a higher level must know the service information of all its offspring agents.