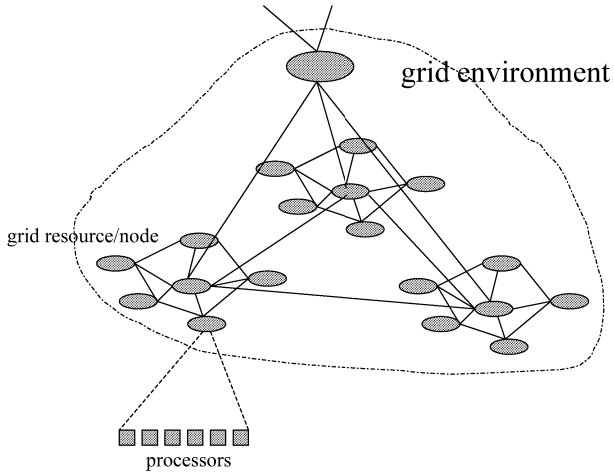
Agent-Based Grid Load Balancing Using Performance-Driven Task Scheduling

Presented by: Xiaolong Jin

Based on: J. Cao, *et al.*, Agent-Based Grid Load Balancing Using Performance-Driven Task Scheduling, in Proceedings of 17th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2003), Nice, France, April 2003,

Grid Environment Considered



Performance-Driven Task Scheduler

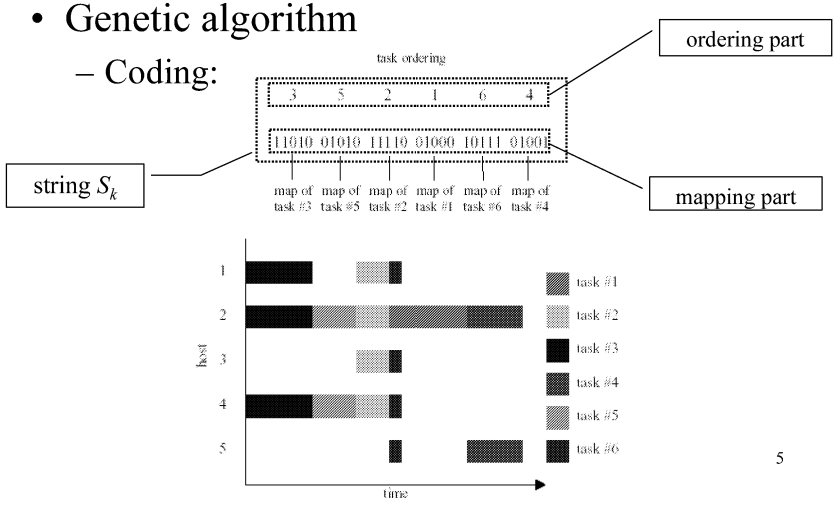
- Resources
  - A grid resource (node) P with n processor
  - Resource model ρ_i describes perf. info. of processor p_i
- Tasks
  - m parallel tasks T to be run on P
  - Application model σ_i describes perf. related info. of tasks T_i
  - δ_i is the deadline requirement of task T_i

Performance-Driven Task Scheduler (Cont.)

- Schedule
  - A schedule is a set of $\overline{p}_j \subseteq P$ ($\overline{\rho}_j \subseteq \rho$) allocated to task T_j, and a set of start time τ_j
  - The execution time for task T_j is a function $t_x(\overline{\rho}_j , \sigma_j)$
  - The completion time is: $\eta_j = \tau_j + t_x(\overline{\rho}_j , \sigma_j)$
  - Makespan ω of a schedule: $\omega = max_j \{\eta_j\}$, i.e., the latest completion time of any task
  - Scheduler goal:
    - Minimize ω, and
    - $\forall$j, $\eta_j \leq \delta_j$

Performance-Driven Task Scheduler (Cont.)

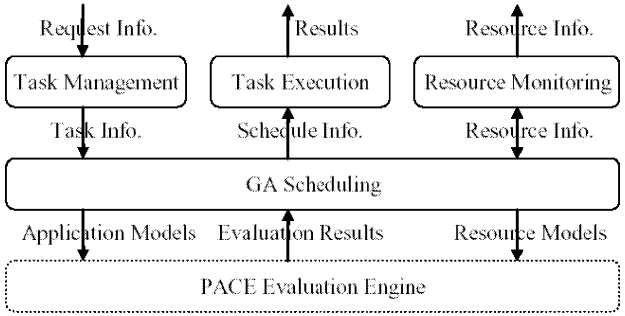
- Genetic algorithm
  - Coding:



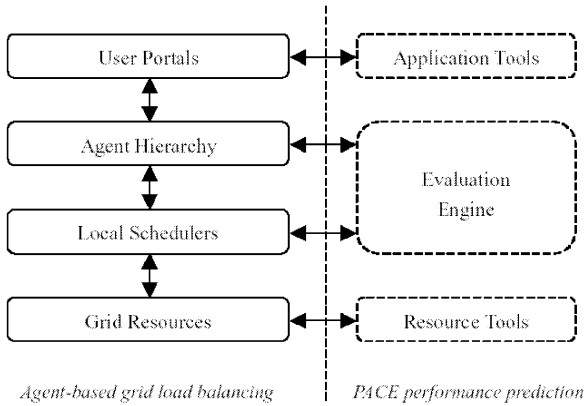
Performance-Driven Task Scheduler (Cont.)

- Fitness function
  - Cost value: $f_c^k = \frac{W^m \omega_k + W^i \varphi_k + W^c \theta_k}{W^m + W^i + W^c}$
    where ω_k is makespan, φ_k is the penalty for idle time, θ_k is the contract penalty.
  - Fitness: $f_y^k = \frac{f_c^{max} - f_c^k}{f_c^{max} - f_c^{min}}$
    where f_c^{max} and f_c^{min} are the best and worst cost value in the GA population
- Crossover: crossover the ordering part, then the mapping part
- Mutation: similar

Scheduler Implementation



Agent-Based Grid Load Balancing System



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.

Agent-Based Grid Load Balancing System (Cont.)

- A task tends to be dispatched to a grid resource that has less workload and can meet the deadline requirement.
- The discovery process does not aim to find the best service for each task, but endeavors to find an available service provided by a neighboring grid resource

System Performance Metrics

- Average advance time $\varepsilon = \dfrac{\sum_{j=1}^{M}(\delta_j - \eta_j)}{M}$
- Average resource utilization rate

$$\upsilon = \frac{\sum_{i=1}^{N} \upsilon_i}{N}, \quad \upsilon_i = \frac{\sum_{\forall j, P_j \in P_i} (\eta_j - \tau_j)}{t}$$

Where v_i is the resource utilization rate of processor P_i

- Load balancing level

$$\beta = 1 - \frac{d}{\upsilon}, \quad d = \sqrt{\frac{\sum_{i=1}^{N}(\upsilon - \upsilon_i)^2}{N}}$$

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.