A Genetic Algorithm with Communiation Costs to Schedule Workflows on a SOA-Grid

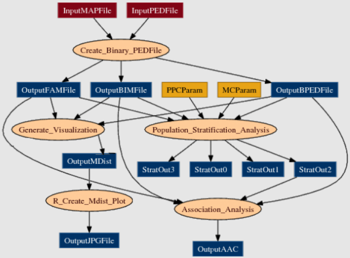
Laurent PHILIPPE

Co-authors: **Lamiel Toch** and **Jean-Marc Nicod**
Laboratoire d'Informatique de Franche-Comté
Université de Franche-Comté
Besançon

HETEROPAR - August 2011

Workflow applications

- Combine several applications or application modules
- Precedence constraints (Files)
- Application domaine : Astronomy, Bioinformatics, Chemistry, Climate Modeling, Computer Science, Image Processing, etc.
- Batch processing
- Collection of workflows



SOA Grids

- Provides applications access
- Execution on clusters
- Simple acess for scientists
- Tools : DIET or NINF-G

Contents

Framework model

Applicative framework

- Collection $\mathcal{B} = \{\mathcal{J}^j, 1 \le j \le N\}$ of N workflows to schedule
- Workflow $\mathcal{J}^j$ is represented by a DAG $\mathcal{J}^j = (\mathcal{T}^j, \mathcal{D}^j)$
  - $\mathcal{T}^j = \{T_1^j, \dots, T_{n_j}^j\}$: the tasks
  - $\mathcal{D}^j$: the precedence constraints
  - $F_{k,i}^j$ is the file sent between T_k^j and T_i^j when $(T_k^j, T_i^j) \in \mathcal{D}^j$
- $\mathcal{T} = \cup_{j=1}^N \mathcal{T}^j = \{T_{i_j}^j, 1 \le i_j \le n_j \text{ and } 1 \le j \le N\}$: set to schedule
- Typed tasks : $t(i,j)$ as the type of task T_i^j.

Framework model - 2

Target platform

- Platform PF : n machines modeled by an undirected graph $\mathcal{PF} = (\mathcal{P}, \mathcal{L})$
  - The vertices in $\mathcal{P} = \{p_1, \ldots, p_n\}$ represent the machines
  - The edges of $\mathcal{L}$ are the communication links
  - Each link (p_i, p_j) has a bandwidth $bw(p_i, p_j)$
- τ : set of task types available
  - Each machine p_i is able to perform a subset of τ.
  - $t \in \tau$ is available on the machine p_i, $w(t, p_i)$ is the time to perform a task of type t on p_i.
- $a(i, j)$ is the machine on which T_i^j is assigned.

Framework model - 3

Communication model

- one-port model
  - one data transmitted / communication link
  - one reception and one transmission / node
- $\mathcal{R}(p_k, p_i) = \{(p_j, p_{j'}) \in \mathcal{L}\}$ is a route from p_k to p_i.

Problem definition

- Static scheduling
- Makespan optimization for the collection of worflows

Related works

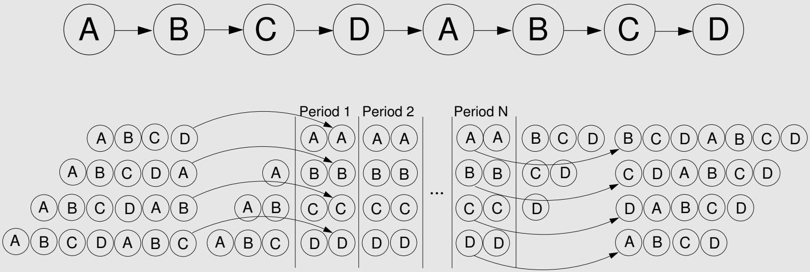
Workflow Scheduling

- Makespan optimization : NP-Hard Problem
- List based heuristics : HEFT, Critical Path, etc.
- Difficult in heterogeneous contexts

Advanced algorithms

- GA for scheduling
  - GA give good results on complex systems
  - But still a heuristic, distance to optimal ?
- Steady State :
  - flow optimization
  - identical intrees
  - optimal results

Steady-state Scheduling

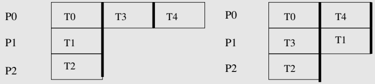


Contents

GA without communication costs

Classical GA for workflow :

- gene = task
- chromosome one row per processor
- phenotype = schedule
- $fitness = 1/makespan$
- population, generation, crossover, mutation ...



Do not take communication into account

With Communication Costs

Communications in the chromosome

- Communication task
- One row per communication link
- Dependencies to the source and target node -> inconsistent communications
- Poor efficiency

Evaluation function

- Communications depends upon tasks placement
- Fitness evaluation with comunication costs
- Used solution

Algorithm : fitness of a chromosome

Data : $\mathcal{T}_{ToSched}$: remaining tasks, $C(T_i^j)$: completion time of T_i^j, $\sigma(T_i^j)$: start time of T_i^j on $p_{a(i,j)}$, $\delta(p_u)$: next time p_u is idle, $w(t, p_i)$: the time to perform a task of type t on p_i, $CT(F_{k,i}^j)$: the communication time to send $F_{k,i}^j$ along route $\mathcal{R}(p_{a(k,j)}, p_{a(i,j)})$

$\mathcal{T}_{ToSched} \leftarrow \mathcal{T}$
while $\mathcal{T}_{ToSched} \neq \emptyset$ **do**
  choose a free task $T_i^j \in \mathcal{T}_{ToSched}$ (EFT heuristic)
  $\mathcal{T}_{pred} \leftarrow \{T_k^j | (T_k^j, T_i^j) \in \mathcal{D}^j\}$ and $\sigma(T_i^j) \leftarrow 0$
  foreach *task* $T_k^j \in \mathcal{T}_{pred}$ **do**
    $\sigma(T_i^j) \leftarrow \max(\sigma(T_i^j), C(T_k^j) + CT(F_{k,i}^j))$
  $\sigma(T_i^j) \leftarrow \max(\delta(p_{a(i,j)}), \sigma(T_i^j))$
  $C(T_i^j) \leftarrow \sigma(T_i^j) + w(t(i,j), p_{a(i,j)})$
  $\delta(p_{a(i,j)}) \leftarrow C(T_i^j)$ and $\mathcal{T}_{ToSched} \leftarrow \mathcal{T}_{ToSched} \setminus \{T_i^j\}$
return $fitness(ch) = 1/C_{max} = 1/max_{T_i^j \in \mathcal{T}}(C(T_i^k))$

Contents

Experimental settings

Simulations

- SimGrid-MSG
- GA = 200 individuals

Platforms

- Random platform generation : uniform distribution
- Platform size : 4 to 10 nodes
- Homogeneous
- Heterogeneous
- CCR : communication to computation ratio

Experimental settings - 2

Applications

- Batch sizes from 1 to 10.000
- Applications : 4 to 12 tasks
- 1900 simulations of platform/application
- Heterogeneity :
  - Execution from 1 to 10
  - Communications from 1 to 4

Contents

Communication Model

- No cost
- Static
- 1-route Bellman-Ford
- 3-route Bellman-Ford

Communication Model - Results

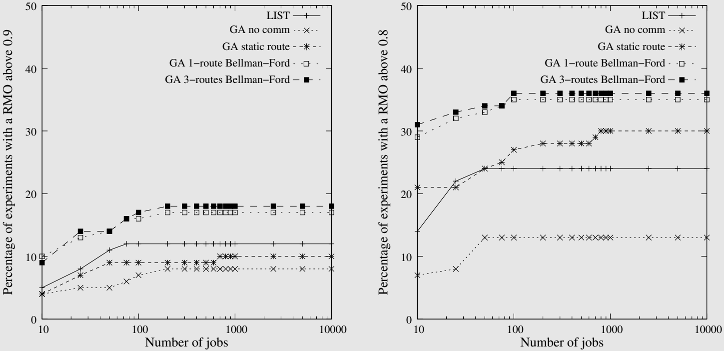
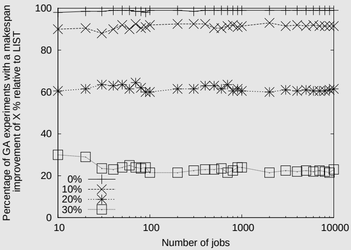
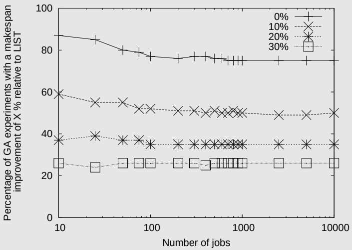


FIGURE: Comparing different algorithms to choose the route

GA Improvement (3-Bellman-Ford)



a. Improvement for different DAGs



b. Improvement for identical DAGs

Contents

Relative Measure to Optimal

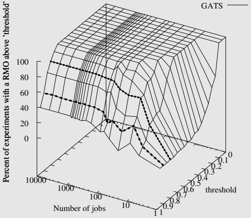
Distance to optimal ?

- Algorithm improves the quality of the results
- Case of collection of intrees : Steady state algorithm gives optimal flow
- Lower bound

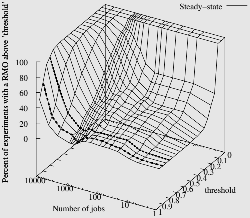
Relative measure to Optimal (RMO)

- Optimal throughput ρ
- Lower bound $L_0 = \frac{N}{\rho}$, N number of intrees
- $RMO = \frac{L_o}{makespan_r}$, $makespan_r$ result of the algorithm

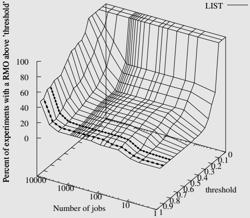
Fully homogeneous platforms, $CCR \approx 0.01$



a. GA algorithm

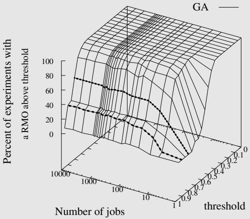


b. Steady-State algorithm

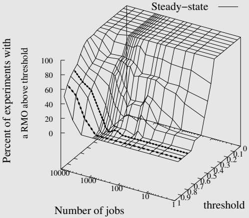


c. LIST algorithm

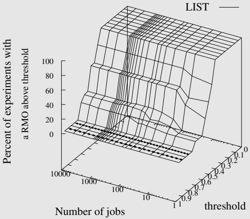
Fully homogeneous platforms, $CCR \approx 1$



a. GA algorithm

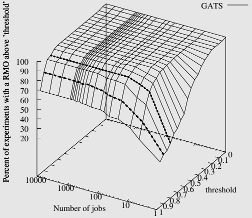


b. Steady-State algorithm

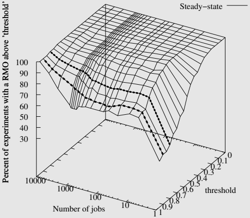


c. LIST algorithm

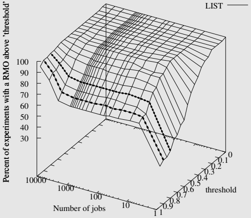
Fully heterogeneous platforms, $CCR \approx 0.01$



a. GA algorithm

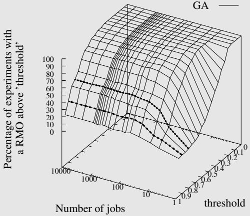


b. Steady-State algorithm

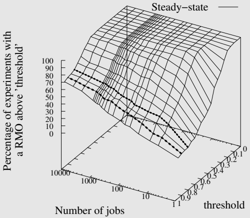


c. LIST algorithm

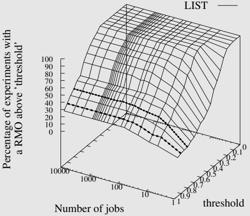
Fully heterogeneous platforms, $CCR \approx 1$



a. GA algorithm



b. Steady-State algorithm



c. LIST algorithm

Conclusion and future works

Algorithm's performance :

- GA Scheduling for batches of workflows on SOA Grids with communication costs
- Collection of different workflows
- Identical intrees, comparison to optimal
- Complex implementation

Future Works

- Other communication models
- Other Genetic representation, network driven

Thank you !