Multi-workflow scheduling

Raphaël Bolze
LIP ENS Lyon
GRAAL project
http://graal.ens-lyon.fr
Introduction

• State of the problem
  - Workflow
  - Grid resources
  - Target applications
    ▪ Pipealign
    ▪ Docking
    ▪ Robinson
    ▪ Cosmology
  - Related work

• Heuristics for solve the problem
  - HEFT heuristic
  - Several other heuristics
  - Simulation
    ▪ first observations

• DIET integration

• Next
Problem

- Several users share grid computing resources (heterogeneous)
- Each user can launch an application (expressed with a workflow)
- Questions:
  - How can we schedule (ordering and allocation) tasks?
  - Can we be fair?
Definition: grid resources

- Grid resources are computing nodes fully interconnected
  - Interconnections are heterogeneous
  - Characteristics of resources are heterogeneous
Definition: workflow

_workflow definition:

- Direct Acyclic Graph (DAG)
  - Each vertex is a task
  - Each directed edge represents a communication between tasks

_Questions:

- Ordering problem?
- Mapping problem?
Targets applications
Docking Application

- Detection of protein-protein and protein-DNA interactions.
- Screening a database containing thousands of proteins for functional sites involved in binding to other proteins, DNA or ligand targets.
PipeAlign Application

- The sequence-to-function relationship can be understood through the analysis of conserved patterns and evolution of protein organization mainly based on amino acid sequence comparisons in the context of the multiple alignments.

![PipeAlign diagram](image)
This application annotate human genes according to their expression in neurological or muscular tissues, but also to the expression of their homolog other species.
Cosmology application

- Simulate the evolution of dark matter particles during time to compare it to the real observation.

Centre de Recherche en Astronomie de Lyon
Related work

- List algorithms
- Clustering algorithms
- Duplication based algorithms
- Metaheuristics

None for multi-workflows online
List Scheduling HEFT

- **List scheduling : HEFT**
  - **Ordering**
    - Set the weights of the tasks
    - Set the weights of the edges
    - Compute the rank (critical path, b-level) of each task
    - Sort the tasks into a list $L$ by non-increasing order of their rank
  
  - **Mapping**
    - While the list $L$ of tasks is not empty
      - Select the first task $t$ of the list $L$
      - Select the resource $r$ that have the earliest finish time for the task $t$
      - Allocate task $t$ on resource $r$
      - Remove $t$ from list $L$. 
Online extension for multiple DAGs
HEFT multi-dags online
Other heuristic

Grid
Framework of heuristics

Each time a new DAG is submitted{
- compute rank (critical path) of each DAG’s tasks
- Sort the DAG’s tasks by non increasing order of their rank
- put the DAG in a list D
while there are unscheduled task {
- select a DAG $d$ from the list D
- select the first unscheduled task $t$ from DAG $d$
- choose the EFT server $s$ for task $t$
- allocate $t$ on $s$
}
Implemented and tested heuristics

- Online Heuristics
  - F1
  - F1 oldness
  - F2
  - F2 oldness
  - Round Robin
  - Random
  - FIFO
  - SRPT
  - LRPT
  - HEFT
  - HEFT oldness

Each time a new DAG is submitted{
  - compute rank (critical path) of each DAG’s tasks
  - Sort the DAG’s tasks by non increasing order of their rank
  - put the DAG in a list D
  while there are unscheduled task {
    - select a DAG $d$ from the list D
    - select the first unscheduled task $t$ from DAG $d$
    - choose the EFT server $s$ for task $t$
    - allocate $t$ on $s$
  }
}
HEFT oldness multi-dags online

Grid
Simulation :

- Need simulation to simply compare heuristics
- Write a simulation software for testing heuristics

- Explore the space of possible variations
  - Submission time
  - Resources: number, homogeneous, heterogeneous
  - DAG (shape, number and size of the tasks)

demo
First observations

• Oldness :
  ▪ Improve average flow with comparable makespan
• Round Robin heuristics
  ▪ RR1 & RR4
  ▪ RR2 & RR3
• Fairness :
  ▪ F1, F2 do not have expected behaviours.
  ▪ FIFO, SRPT, HEFT oldness performed well.
  ▪ Fairness vs. makespan optimization
• General remarks :
  ▪ SRPT advantage small DAGs (in terms of CP)
• Important parameters :
  ▪ Arrival time and type of DAGs

Need a table to summarize observations
Multi-workflows and DIET
MultiWorkflow in DIET

- Meta scheduler distributed in the client and in the MA-DAG (Abdelkader)
• Make a complete survey of the behavior of the heuristics

• Try to make a classification of heuristics against dags types
  ♦ Criteria:
    • Makespan : i.e. length of the critical path
    • Total amount of work ( sum of \(w_i\) )
    • DAG’s shapes
    • Some others

• Add heuristics based on other ordering and resource selection criteria
  ▪ Example : SDC, DLS

• Test heuristics on a real environment : DIET + applications

• Dynamic DAGs ?
Questions ?