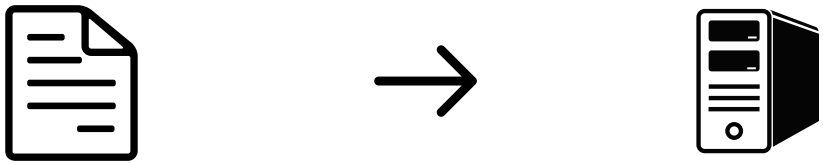
Toward an autonomic approach of workflows distribution on cloud

Hadrien Croubois

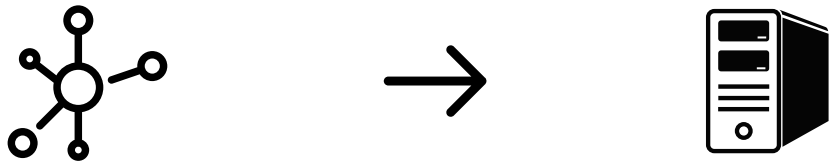
Phd Student at Avalon, Laboratoire de l'informatique du Parallélisme
École Normale Supérieure de Lyon, France

Introduction

Scheduling: Matching jobs and resources

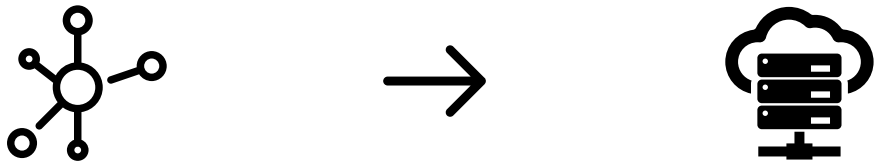


Scheduling: Matching jobs and resources



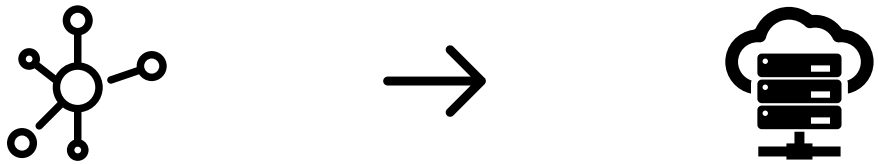
- Jobs definition is changing,

Scheduling: Matching jobs and resources



- Jobs definition is changing,
- Resources are changing.

Scheduling: Matching jobs and resources



Challenge

The matching logic need to consider chose changes.

Challenge: Accounting for three factors

- Complex jobs (workflows),
- Multi-Agents (collaborative),
- Dynamic platform (IaaS cloud).

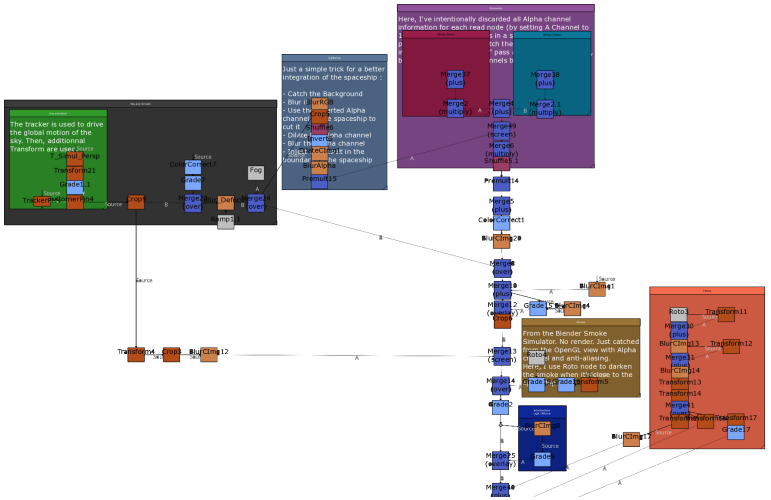


Figure 1: Rendering workflow in Natron

Challenge: Accounting for three factors

- Complex jobs (workflows),
- Multi-Agents (collaborative),
- Dynamic platform (IaaS cloud).

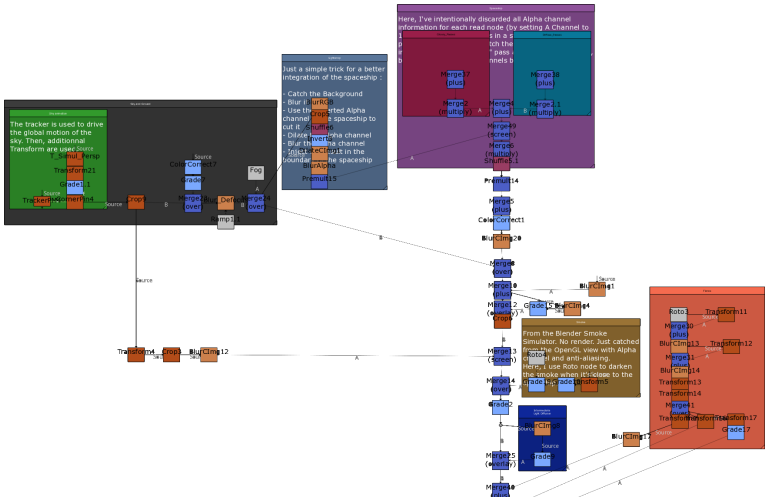


Figure 1: Rendering workflow in Natron

State of the art

Previous work considers at most 2 of those 3 factors.

Our approach

Dataflows: managing data locality

- VMs can migrates,
- Unpredictable network topology and bandwidth,
- Shared DaaS (NFS, Dropbox, Amazon S3, . . .)

Reconsidering communications

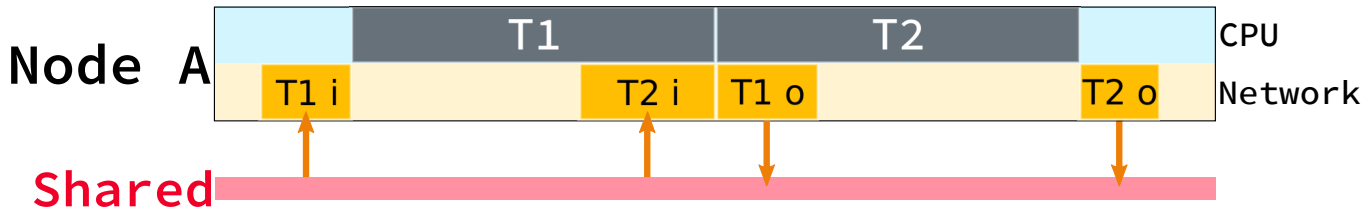


Figure 2: Each nodes is divided into two units: the CPU where tasks are computed, and the network interface which handles communications.

Cutting the problem into smaller ones

Cutting the problem into smaller ones

- Offline task clustering using edge zeroing heuristics,
- Méta-tasks prioritisation,

Cutting the problem into smaller ones

- Offline task clustering using edge zeroing heuristics,
- Méta-tasks prioritisation,

- Méta-tasks deployment,

Cutting the problem into smaller ones

- Offline task clustering using edge zeroing heuristics,
- Méta-tasks prioritisation,

↓

- Méta-tasks deployment,

↓

- Load estimation,
- Platform adaptation.

Cutting the problem into smaller ones

- Offline task clustering using edge zeroing heuristics,
- Méta-tasks prioritisation,

↓

- Méta-tasks deployment,

↓

- Load estimation,
- Platform adaptation.

Challenge

Build specific solutions for each item.

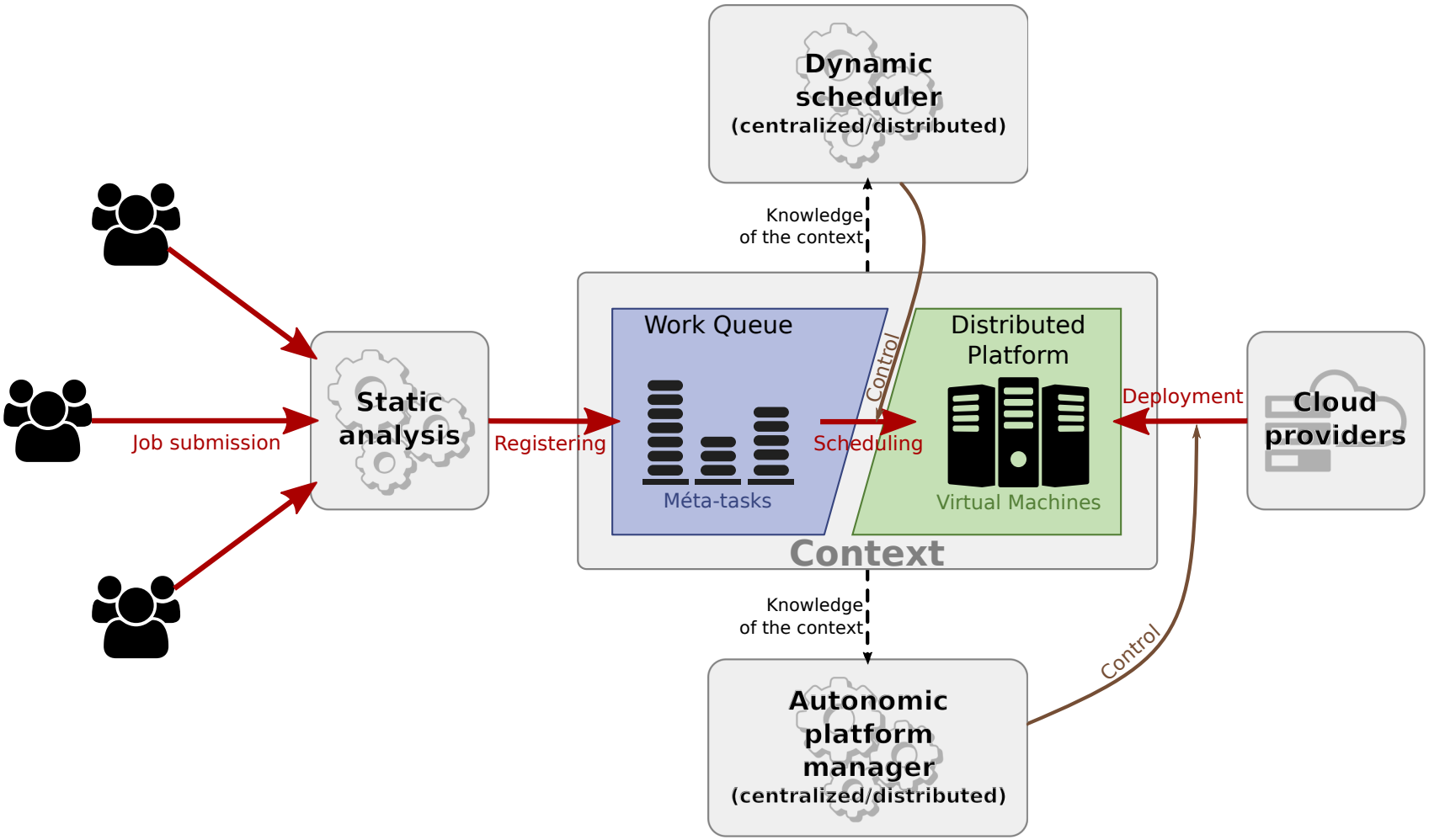


Figure 3: Framework architecture

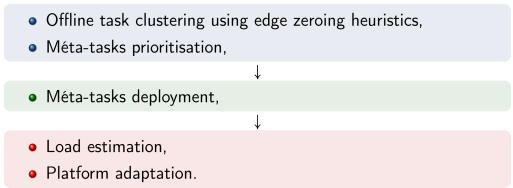
Conclusion

Where are we ?

- We have identified the problem,
- We divided it into sub-problems,
- We have a framework for building a solution.

What is next ?

- 3 steps (5 problems) to solve,
- Open to collaboration on any of those,
- Implementing and validation the model,
- Scalability ?



Thank you for your attention.