

Load prediction in a workflow system

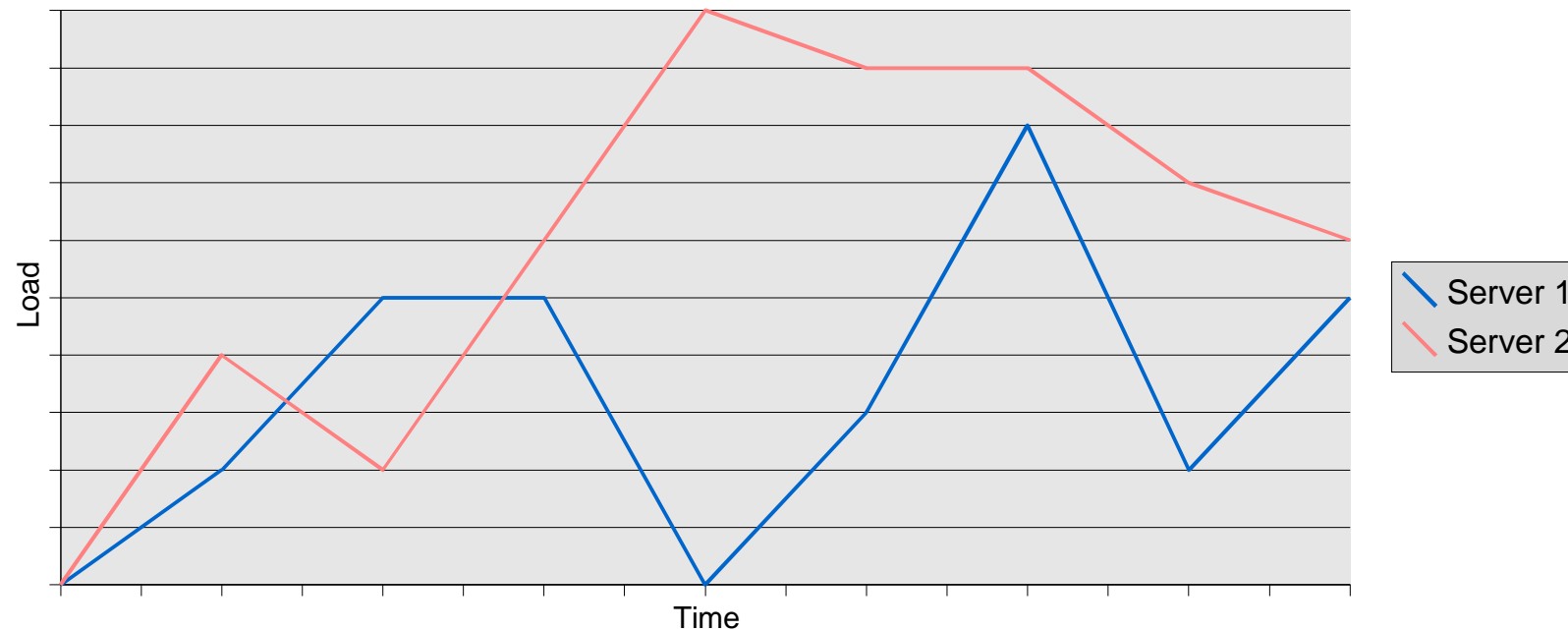
Biörn Biörnstad
ETH Zürich

Agenda

- Load Prediction
- Motivation
- Workflow model
- Method Overview
- Simple Case
- Advanced Case
- Assumptions

Load Prediction

- Prediction of the load on each server in the system
- Load = # tasks

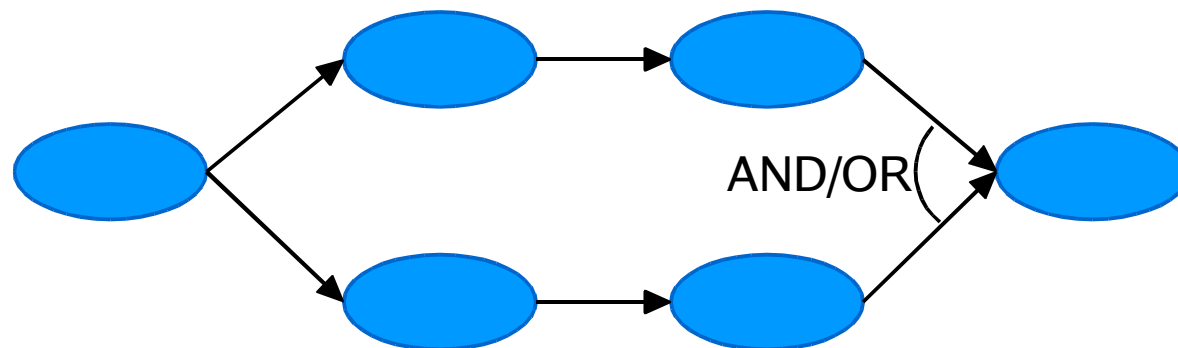


Motivation

- Predict time windows suitable for maintenance
- Predict overload and avoid it (QoS)
 - Reschedule
 - Admission control
 - Resource allocation
- Web services \leftrightarrow Workflow?
 - Composite service is modeled as a workflow

Workflow model

- Directed graph of tasks
- AND/OR conditions
- Several different servers (categories)
- A task is executed on a certain server (category)

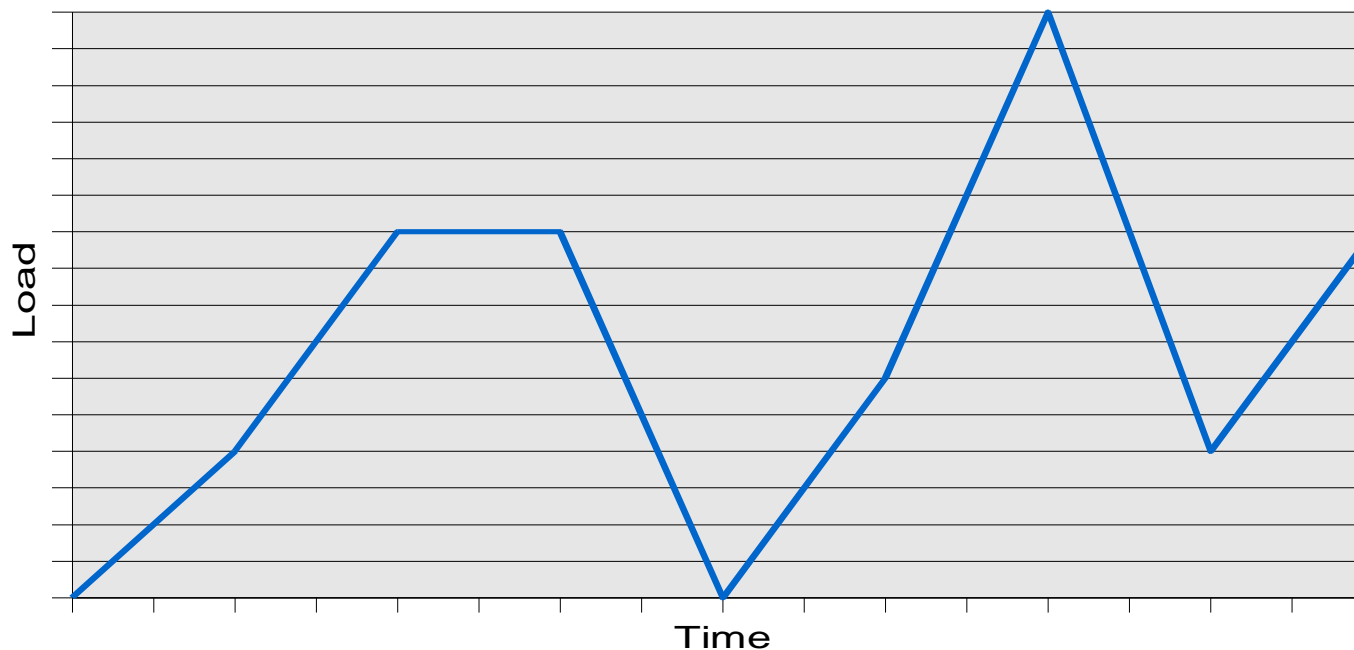


Method Overview

- Collect statistics from WFMS
 - Observation
 - Analysis of logs
- Extract course of each process
- Calculate load of each server

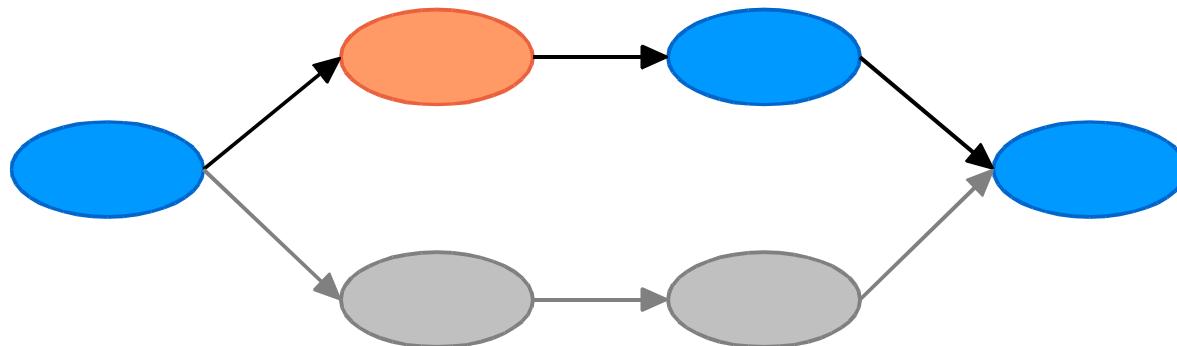
Simple Method

- Extract data from the logs for each process
- Draw graph of load per process/server



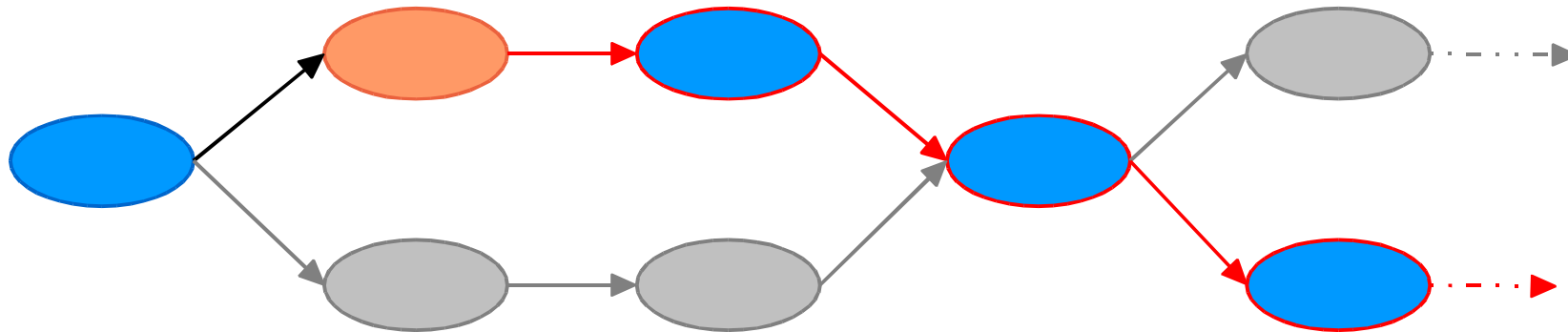
Advanced Method

- Consider structure of process (tasks)
- Consider possible execution paths
- When in the middle of a process, course of the process can be predicted more accurately



Advanced Method (cont.)

- Identify common paths
- eliminate impossible ones



Advanced Method (cont.)

- Measure the execution time for each task (probability function)
- Compute the start times of tasks by adding the execution times
- Compute a graph for the path/server
- Time-shift and add paths for each server

Assumptions

- Response time of a server is constant

Summary

- Load prediction can help to
 - react to changes in the environment
 - guarantee QoS
- CS can be executed as Workflows
- Simple and more advanced methods depending on complexity of workflow

Future Work

- Develop more accurate Methods
- Efficient computation of load
 - incremental: maintain a total load, add differences
- Load prediction for distributed systems