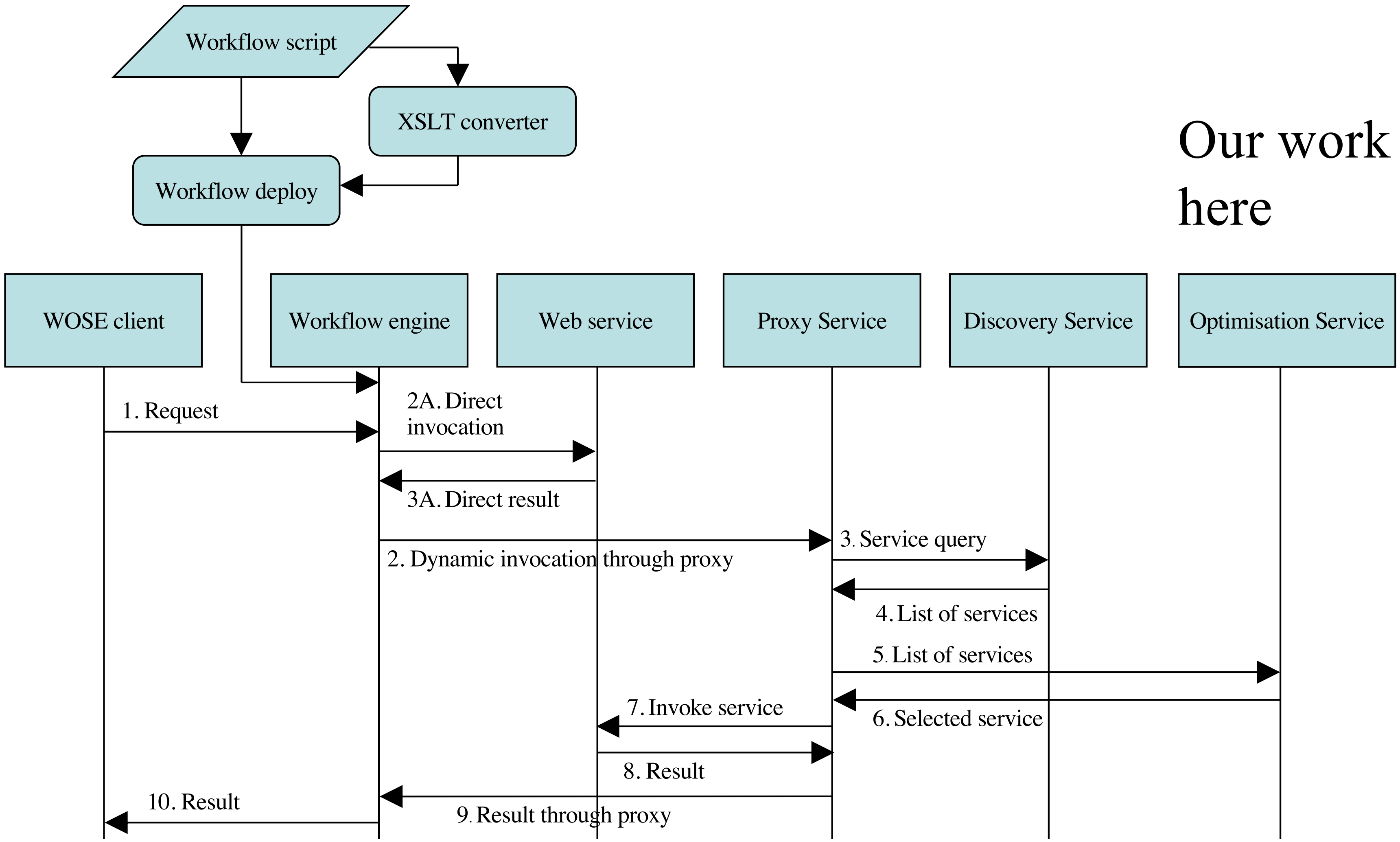
- Title: Grid Workflow Scheduling in WOSE (Workflow Optimisation Services for e-Science Applications)

- Authors: Yash Patel, Andrew Stephen McGough and John Darlington

Overview

- Multiple copies of a service with different performance or other user defined set of criteria; and these services cannot be selected at design time because their performance is not known at that time.
- workflow optimisation by selecting optimal web services at run-time and integrating dynamic selection of web service into workflow

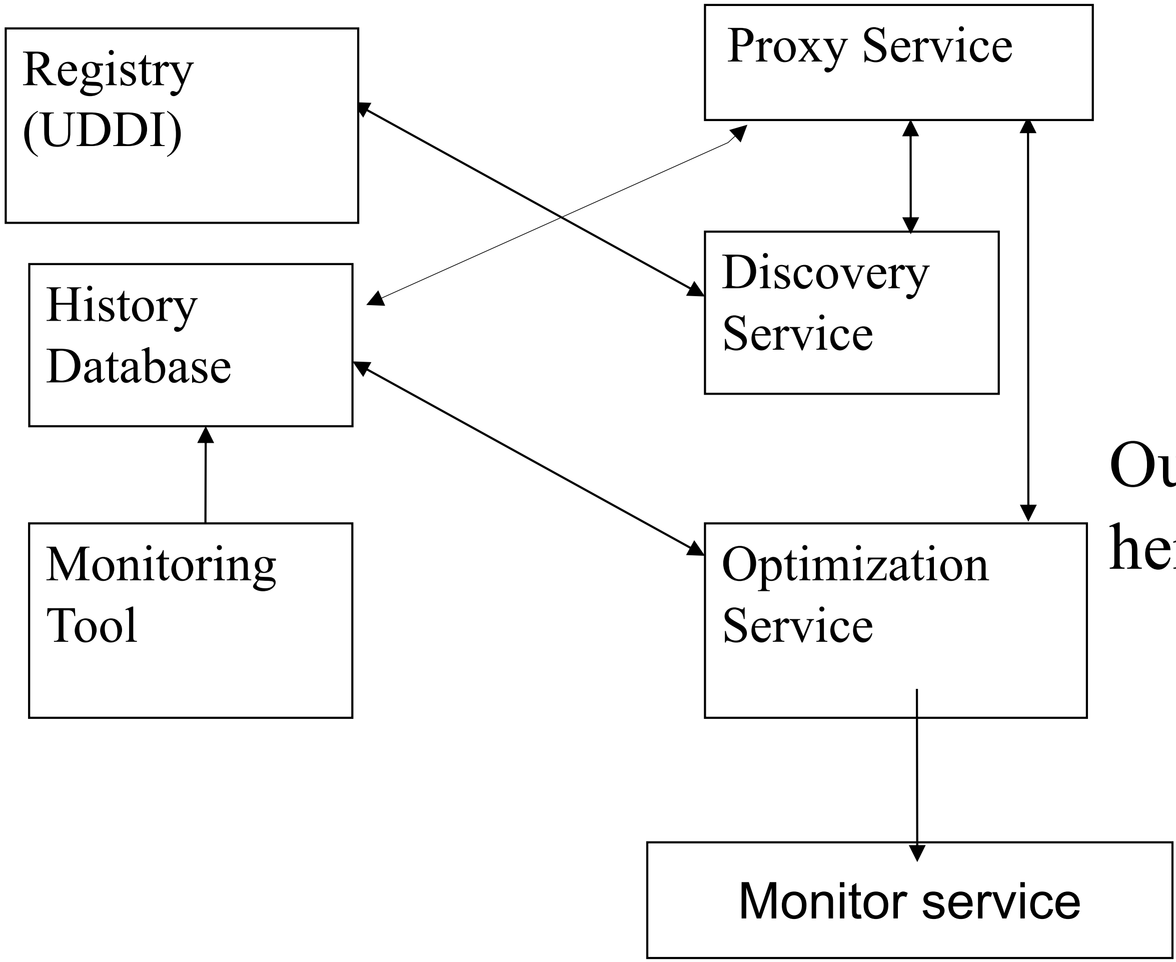
WOSE Architecture



Our work fits here

Developed by Cardiff University

WOSE Architecture



Our work fits here

Developed by Cardiff University

Our Approach

- Previous Optimisation Framework: Service-by-service basis approach of scheduling services and relies on real-time load information for making scheduling decisions. No QoS support

- Our Approach: Provides sufficient QoS guarantee whilst respecting QoS requirements of workflows for entire lifetime of workflows and uses Queuing Theory + Stochastic Programming approaches (doesn't rely on real time information)

Our Approach

- Stochastic Programming : It is a technique to solve optimisation problems involving uncertainty
- Stochastic Programming = Deterministic Mathematical Programming + Uncertainty
- Stochastic Programming : coefficient of variables having probability distributions
- Deterministic Mathematical Programming : coefficient of variables are known numbers

Our Approach

- Formulate workflow scheduling problem as a 2-stage stochastic program

- Scheduling program: Workflow structure + States of services (mean, variance of waiting times) + Performance models of workflow tasks + QoS requirements of workflow and its tasks

Our Approach

- Why is it stochastic?
- workflow tasks need to be scheduled now [Stage-1], whilst providing guarantee that future workflow tasks will still meet QoS requirements of workflow (uncertain) [Stage-2]
- [Stage-2]: Uncertain as demands for Grid services are random, service times are not deterministic, workflows are dynamic, services themselves may disappear

Our Approach

- Formulate workflow scheduling problem as 2-stage stochastic program

- Stage-1 is fairly straight-forward: select services which satisfy QoS requirements of workflow tasks that need to be scheduled immediately (now)
- Stage-2: Since coefficients of variables have probability distributions, we compute their expectations by SAA (sample average approximation) [Shapiro et al.]

Our Approach

- Scheduling Problem:

  minimise[stage-1 error + E(stage-2 error)]

  subject to: various execution, deadline, cost, reliability etc constraints
- E(stage-2 error) is computed using SAA problem
- Error is the penalty of failing to meet the QoS requirements

Our Approach

- The variables associated with penalty (one per constraint) are also present in the constraints such as execution, cost constraints etc
- If the constraints are infeasible, it forces the penalty variables to bind with some value
- Hence the objective reflects a value
- The coefficients of these variables in the objective are the inverse of the maximum coefficient in the relevant constraint.

Our Approach

- SAA Problem: Solve stage-1, use its result in N stage-2 programs. These N programs are generated by sampling (Monte-Carlo or Latin Hypercube)
- Take an average value of minimised objective values of these N programs and the stage-1 error. That is SAA problem
- Stage-2 programs are similar to stage-1 programs
- Stage-1 program: obtains scheduling solutions for workflow tasks that need to be immediately scheduled
- Stage-2 programs: obtain for future workflow tasks (of course respecting constraints)

Our Approach

- Probability distributions of variable co-efficients: many such as waiting time for web services, service time for web services
- 1 stage-2 program is a joint realisation of their values (1 sample)
- N stage-2 programs means N samples

Algorithm for stochastic scheduling of workflows

- **Step 1**: Choose sample sizes N and N' $\geq$ N, iteration
- count M, tolerance ε and rule to terminate iterations
- **Step 2**: Check if termination is required
- **for** m = 1, . . .,M **do**
- **Step 3.1**: Generate a sample of size N and solve the SAA problem. Let the optimal objective be O^m for corresponding iteration
- **end for**
- **Step 3.2**: Compute the average and variance as L and Var^L (M values)
- **Step 3.3**: Generate a sample of size N', use one of the feasible stage-1 solution and solve the SAA problem and compute average and variance as U and Var^U (N' values)
- **Step 3.4**: Estimate the optimality gap (Gap = |L - U|) and the variance of the gap estimator ($Var^{Gap} = Var^L + Var^U$)
- **Step 3.5**: If Gap and/or Var^{Gap} are large, tighten stage-1 QoS bounds, increase the sample sizes N and/or N', and return to **step 2**
- **Step 3.6**: If Gap and/or Var^{Gap} and stage-1 objective value are small, choose stage-1 solution and stop
- **end for**

Algorithm in a nutshell

- The algorithm obtains epsilon-optimal solutions and sample size N guarantees that
- The algorithm ensures that QoS requirements can be satisfied with sufficient guarantee and variability of penalty is minimum
- If it is not then cost and time allocations to stage-1 workflow tasks are reduced so that in the next iteration probability of satisfying QoS requirements of stage-2 tasks increases

Scheduling Strategies

- The SP (stochastic programming) scheme (similar to 2nd scheme) is compared with 2 traditional schemes

- 1st scheme: Obtains scheduling solutions for all workflow tasks at the same time. Hence is static
- 2nd scheme: Obtains scheduling solutions for workflow tasks dynamically, meaning as and when required

Experimental Results

- 1st scheme just solves 1 ILP which obtains solutions respecting the QoS requirements and keeping the penalty to a minimum
- In the other two schemes, cost and time allocations to stage-1 workflow tasks initially is done using upper bound of the 95th confidence interval of execution distribution of workflow tasks
- In all the 3 schemes, the expected execution time for stage-1 workflow tasks is calculated as the upper bound of the 95th confidence interval of execution distribution of workflow task and waiting time distribution of services

Experimental Results

- The SP scheme is different to 2nd scheme in the way the scheduling solutions are obtained
- 2nd scheme just solves 1 ILP based on the cost and time allocations of workflow tasks
- SP scheme obtains solutions iteratively through the algorithm and in the process solves numerous ILPs. Cost and time allocations of workflow tasks thus get changed, which don't in the 2nd scheme.

Experimental Setup

- Simulation developed in SimJava
- Experimented with simple, complex and heterogenous workflows
- Results collected for low and high arrival rates, low and high CV of execution distributions of workflow tasks
- Different QoS requirements of workflows
- Statistics (mean response time, cost, failures, utilisation etc) collected for 1000 jobs following 500 jobs that require system initiation

Results

- SP approach performs considerably better over other traditional schemes
- The SP scheme provides sufficient QoS guarantee over the entire life-cycle of workflows
- The scheme performs better particularly when workflow complexity and heterogeneity are high
- At both low and high arrival rates of workflows the SP scheme is a winner
- Average utilisation of services increase in the SP scheme

Future Work

- Experiment with workflows having slack periods
- Enhance the scheduling model (more constraints and more realistic model of web services)

- Thank You