Uncovering the Perfect Place: Optimising Workflow Engine Deployment in the Cloud

Michael Luckeneder and Adam Barker
School of Computer Science
University of St Andrews, UK
adam.barker@st-andrews.ac.uk

ABSTRACT

When orchestrating highly distributed and data-intensive Web service workflows the geographical placement of the orchestration engine can greatly affect the overall performance of a workflow. We present CloudForecast: a Web service framework and analysis tool which, given a workflow specification, computes the optimal Amazon EC2 Cloud region to automatically deploy the orchestration engine and execute the workflow. We use geographical distance of the workflow, network latency and HTTP round-trip time between Amazon Cloud regions and the workflow nodes to find a ranking of Cloud regions. This overall ranking predicts where the workflow orchestration engine should be deployed in order to reduce overall execution time. Our experimental results show that our proposed optimisation strategy, depending on the particular workflow, can speed up execution time on average by 82.25% compared to local execution.

Keywords

Workflow optimisation, Scientific Workflows, IaaS Clouds

1. INTRODUCTION

Scientific workflows are typically orchestrated using a workflow engine running locally within an organisation's network. However, if the Web services in the workflow are data-intensive and spread across many geographical regions, the data might have to move long distances in order to flow from the data sources to the Web services via the orchestrator. This in turn most likely slows down the execution of the workflow and degrades the overall performance.

A possible solution could be to use an Infrastructure as a Service (IaaS) Cloud, such as an Amazon EC2 instance, it is possible to automatically deploy the orchestrator into a suitable EC2 region that is "closer" to the data source and Web service nodes; in turn data would not have to travel as far and therefore execution times could be reduced.

We design, implement and evaluate CloudForecast, a pre-deployment analysis tool[1] which can dynamically compute the "optimal" Cloud region to deploy the orchestrator, given a specific workflow consisting of multiple distributed services. We focus on Directed Acyclic Graph (DAG) based workflows since these are heavily used in the scientific community [1, 2]. In order to characterise the "optimal" Cloud region, we use total geographical distance, network latency and HTTP round-trip time of the workflow.

2. ARCHITECTURE

Figures 1 show a simple, sequential workflow with a data source (wikimedia.org) and two workflow nodes hosted on PlanetLab (`princeton.edu`, `surrey.sfu.ca`). It takes an image from `wikimedia` and then sends it to the `princeton.edu` node for processing. The result from this step is then sent to the `sfu.ca` node.

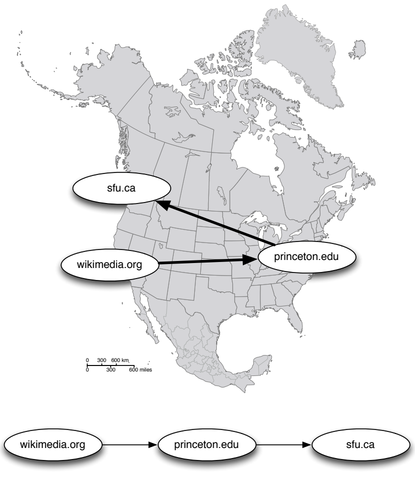

Figure 1: Example workflow

Figure 2 shows the workflow illustrated in Figure 1 in relation to the closest Amazon EC2 Cloud regions (us-east-1, us-west-1, us-west-2). This illustrates the dilemma faced when deciding on the correct Cloud region to deploy the workflow orchestrator to: which Cloud region will result in the lowest execution time?

CloudForecast attempts to solve this problem by evaluating the workflow based on: total geographical distance, network latency (as measured by the UNIX "ping" command) and HTTP round-trip time (as measured by the UNIX "curl" command) between every pair of adjacent nodes. It takes a workflow specification and builds several candidate workflow graphs, which represent the data flow. An example of a candidate graph for the workflow in Figure 2 can be seen in Figure 3. The tool further requires a list of Cloud regions (e.g. Amazon EC2 regions) that should be considered for workflow execution. Every candidate graph will be based on a different one of these Cloud regions and represent a possible optimised workflow using the Cloud instance as the workflow orchestrator. Every edge represents the data flow-

[1] `https://github.com/bigdatalab/movingdata`

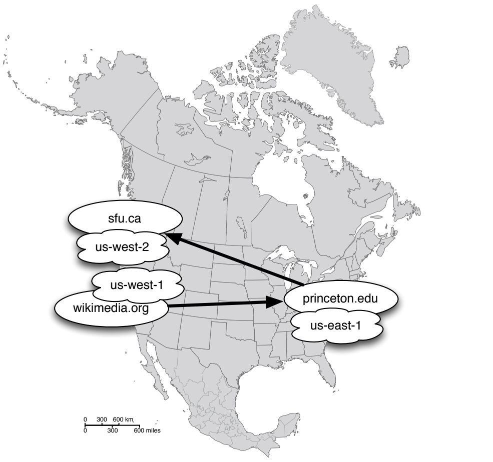

Figure 2: Example workflow with closest Amazon EC2 regions

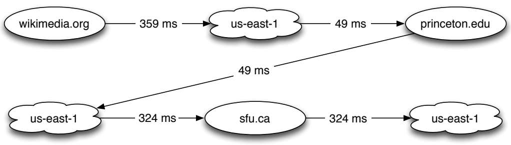

Figure 3: Candidate workflow graph using EC2 region us-east-1 with network latency metric

ing from a Web service node to the Cloud region or vice versa. Since there are 8 EC2 Cloud regions and 3 metrics, the analysis tool will generate a total of 24 candidate graphs.

Based on every candidate graph and metric used, CloudForecast computes an overall score. Using the geographical distance, the n shortest Cloud regions are selected. Then, these are evaluated using the "ping" and HTTP RTT metrics and a final score is calculated. The final output is a table which ranks the Cloud regions by their predicted execution times. Table 1 is an example of the output table for the workflow displayed in Figure 2.

Table 1: Analysis tool result output

EC2 endpoint	final score
us-east-1	92530.42
us-west-2	186251.487
us-west-1	186374.351
sa-east-1	366450.152
ap-northeast-1	421102.237
ap-northeast-2	510982.726
ap-southeast-1	532180.129
eu-west-1	500178094.532

3. EXPERIMENTS

To verify the functionality of CloudForecast, we randomly generated workflows, analysed them and then executed and timed them several times. We chose to use randomly generated workflows to simulate real-life applications of workflows. Both simple sequential (A – G) as well as more complex workflows (H – I) involving sequential, fan-in and fan-out patterns with 2, 3, 4, 5, 7, 10, 12 and 13 nodes were generated.

In order to verify the optimisations proposed by CloudForecast, we implemented a very simplified, bare-bones workflow orchestrator which uses a set of custom workflow nodes implemented on top of PlanetLab.

Figure 4 illustrates the speedup in mean execution time for each sample workflow due to being run in the first-ranked Cloud region compared to local execution. The speedups range from 3% to 188% with a mean of 82.25%. We can conclude that the analysis correctly ranks the Cloud regions to reduce execution time. Please refer to [3] for a more thorough presentation of the framework and evaluation.

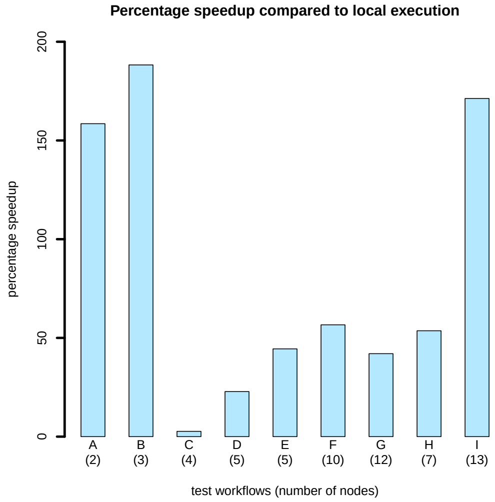

Figure 4: Overall performance gain due to predeployment analysis

4. CONCLUSIONS

We discussed how to increase the performance of highly distributed Web service workflows by dynamically deploying the workflow orchestrator on an IaaS Cloud rather than orchestrating remote services locally. We conclude that orchestrating workflows in the Cloud significantly reduced overall execution time.

5. REFERENCES

[1] A. Barker, J. B. Weissman, and J. I. van Hemert. Reducing data transfer in service-oriented architectures: The circulate approach. *IEEE T. Services Computing*, 5(3):437–449, 2012.

[2] P. Besana, V. Patkar, A. Barker, D. Robertson, and D. Glasspool. Sharing choreographies in openknowledge: A novel approach to interoperability. *JSW*, 4(8):833–842, 2009.

[3] M. Luckeneder and A. Barker. Location, location, location: Data-intensive distributed computing in the cloud. In *IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom 2013*, pages 647–654. IEEE, 2013.