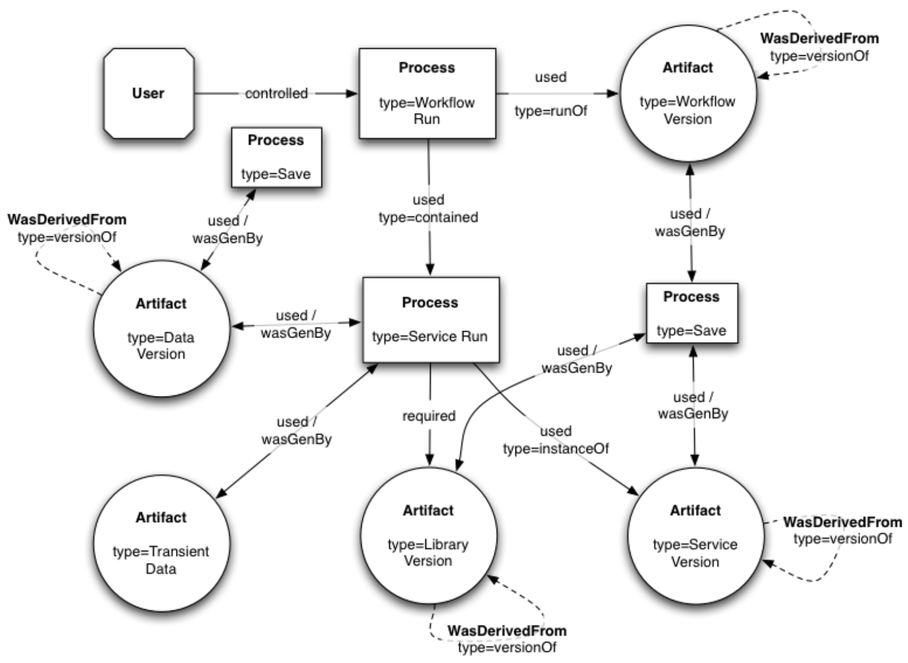
# Prediction of workflow execution time using provenance traces: practical applications in medical data processing

Hugo Hiden, Simon Woodman & Paul Watson
Newcastle University

The use of cloud resources for processing and analysing medical data has the potential to revolutionise the treatment of a number of chronic conditions. For example, it has been shown that it is possible to manage conditions such as diabetes, obesity and cardiovascular disease by increasing the right forms of physical activity for the patient. Typically, movement data is collected for a patient over a period of several weeks using a wrist worn accelerometer. This data, however, is large and its analysis can require significant computational resources. Cloud computing offers a convenient solution as it can be paid for as needed and is capable of scaling to store and process large numbers of data sets simultaneously. However, because the charging model for the cloud represents, to some extent, an unknown cost and therefore risk to project managers, it is important to have an estimate of the likely data processing and storage costs that will be required to analyse a set of data. This could take the form of data collected from a patient in clinic or of entire cohorts of data collected from large studies. If, however, an accurate model was available that could predict the compute and storage requirements associated with a piece of analysis code, decisions could be made as to the scale of resources required in order to obtain results within a known timescale.

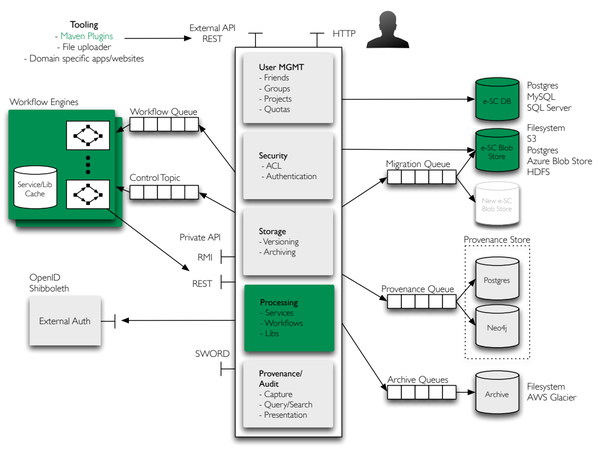
## Using the e-Science Central Provenance Model




The e-Science Central Provenance Model captures details of code versions, data sources and sizes, execution times and transfer data sizes for every invocation of a block within a workflow

This provenance model was extended with a number of additional parameters (CPU usage, concurrent workflow count, machine architecture etc) and the data used to build predictive models for workflow block executions

## Data capture architecture

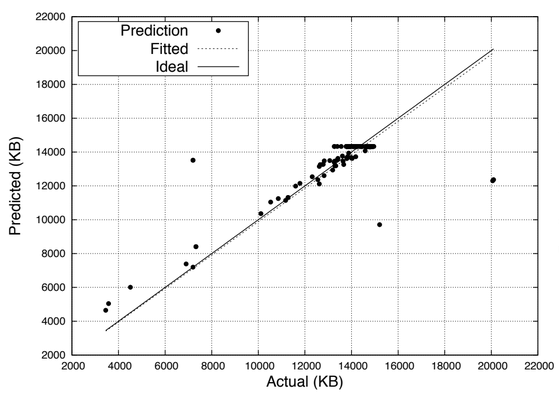



Data was captured via JMS queues to minimise the impact on the running system.

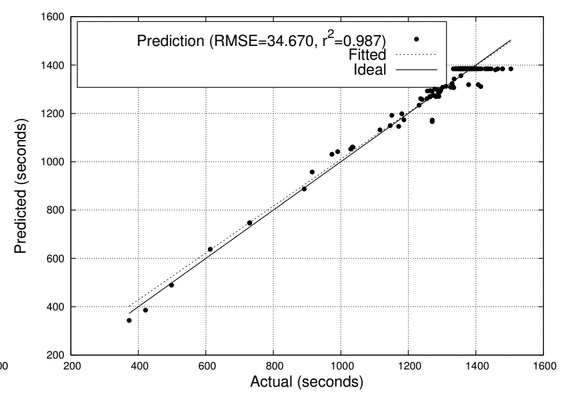
Models were build periodically on separate cloud resources and used to generate predictions

## Movement monitoring results

Good models were obtained for IO intensive code such as GGIR processing of GENEActiv data.



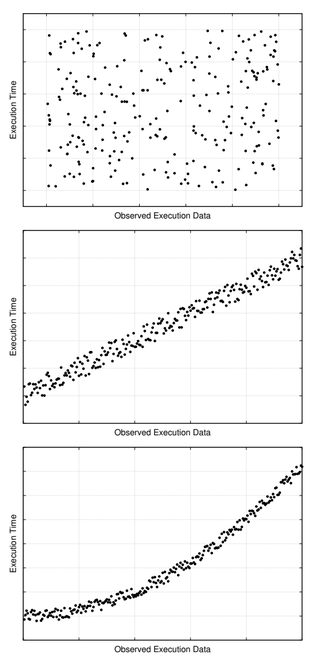

Output size model




Duration model

## Model types

The system has the capability to construct multiple models for each block and use the most appropriate. This can account for the following types of scenario:



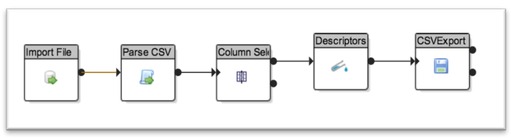

**No relationship**: There is no discernable relationship between block execution time and any collected data.

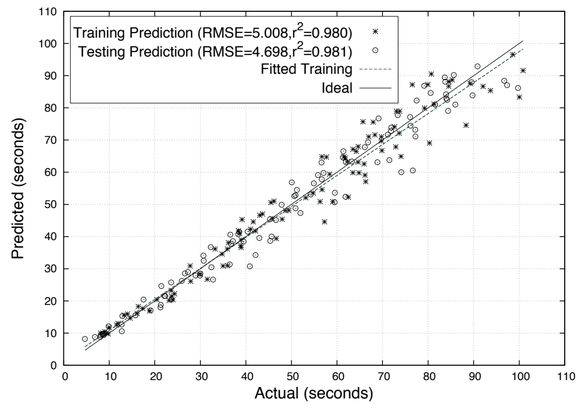
**Linear relationship**: The block displays a linear relationship between execution time and the collected data

**Non-linear relationship**: The block displays a more complex non-linear relationship between execution time and the collected data

## Modelling whole workflow execution time

Models for entire workflow runs were generated by linking together predicitons for individual blocks. This worked well for simple workflows




e-science central

Newcastle University