DAFNI PILOT 4: SPENSER - Synthetic Population Estimation and Scenario Projection model

Table of Contents

Glossary

Item	Definition
Containers	Containers allow a developer to package up an application with all the parts it needs, such as libraries and other dependencies, and ship it all out as one package.
Docker	Docker is a tool designed to make it easier to create, deploy and run applications by using containers.
Argo	A tool for defining and running multiple container Docker applications. DAFNI makes use of this tool.
Kubernetes (k8s)	Kubernetes is an open-source system for automating deployment, scaling and management of containerized applications.

Key Benefits of this Pilot

Integration and access to data: DAFNI makes access to the underlining data faster and simpler by pre-caching the required values. For example the SIMIM model can be run to perform detailed analysis of how any investment scenario is likely to alter future migration. Population data from the Census and mid-year population updates produced by ONS are stored on DAFNI without the user needing to get access tokens. DAFNI's interface also allows a number of different migration based models to be run with results saved into a database.

Ease of access: DAFNI has provided a new web-based User Interface for SIMIM which has been developed to allow the simple specification of regional investment scenarios in terms of new houses, jobs, etc. This allows remote users to utilise the DAFNI compute and data resources to run SIMIM without any complex installation on their local machines.

Improvements in run time: Through DAFNI it has been possible to apply high throughput computing to allow long running jobs used in other SPENSER packages, such as Microsimulation, to be efficiently processed by the DAFNI Kubernetes cluster. Typical run time has been reduced from 90 minutes on existing systems to 30 minutes using the DAFNI cluster.

Non-technical Users: DAFNI's new user interface allows non-experts to easily load pre-defined scenarios, run the simulation and then view the resulting population changes. Thereafter the user interface allows for a more detailed comparison of results through web-based visualisations of the results. More advanced users can experiment with scenarios to explore more detailed aspects of the simulation.

Shared Knowledge: By using DAFNI to implement scenarios, the model can be used by other researchers and analysts across Academia, Government and Industry, providing insight into a range of infrastructure development scenarios.

Introduction to DAFNI Pilots

DAFNI will provide the National Platform to satisfy the computational needs in support of data analysis, infrastructure research and strategic thinking for the UK's long term planning and investment needs. The platform will support academic research that is aiming to provide the UK with a world-leading infrastructure system that is more: efficient, reliable, resilient and affordable. DAFNI will support big data analytics, simulation, modelling and visualisation.

DAFNI Pilots are a series of projects that run alongside the DAFNI core platform development and seek to take existing established infrastructure codes and implement them in a Cloud based environment that emulates the expected future DAFNI system. DAFNI pilot projects are submitted by the members of the DAFNI community and projects are chosen based on proposers' resource availability, benefits to DAFNI such as validating DAFNI's components, stress testing the DAFNI hardware etc. Each pilot project typically runs for 3-6 months and is supported by the DAFNI pilot team, consisting of 2-3 software developers. This will enable the following benefits to the DAFNI and its community:

- Demonstrate the capabilities of the DAFNI infrastructure.
- Feed the community requirements into improving and maturing the DAFNI infrastructure.
- Provide early access for the modellers to test their models on the DAFNI platform.
- Provide additional access to infrastructure models that may form part of the DAFNI service.
- Allow exploration of visualisation techniques useful to infrastructure modellers.
- Highlight typical data set requirements for infrastructure research.

Overview of SPENSER

The SPENSER (**S**ynthetic **P**opulation **E**stimatio**n** and Sc**e**na**r**io Projection Model) project is a collection of software tools to understand future population growth and migration using dynamic microsimulation. It provides synthetic models of household occupation and types. A synthetic model (SM) is one where attributes are consistent with available data, such as census measures, but not identical with the true state, which is not available. For example the SM might describe the number, size and type of households in a region and while these are not identical with reality, they are consistent with available census data, such as the distribution of occupants in each house.

These tools have been developed by Andrew Smith and Nik Lomax at the University of Leeds to model the future population trends within the UK. The software is freely available via Github and uses various open data sources, such as the population and housing projections provided by the ONS (Office of National Statistics).

The current components of SPENSER are illustrated in Figure 1. The components deal with many different aspects of population modelling including:

- UK Census Data API – An interface to the UK Census data with automatic caching of the results [6]. For England and Wales the main source of data is the Nomisweb site, for Scotland and Northern Ireland their own government websites are searched. This interface is used by other packages such as SIMIM.
- UK Population – This interface is to more detailed population data from the various UK government statistics agencies, providing a single unified interface to the various different sources for England, Wales, Scotland and Northern Ireland [7]. This is also used by SIMIM and other tools to efficiently access the underlining data.

- NewOrder – A dynamic microsimulation package for detailed prediction of the structure of a population and its evolution through time [5]. For simulation of the whole UK population this task can be run on HPC hardware as the complex calculation can make effective use of parallel resources.
- Microsimulation – This package is used to estimate population details down to the household level based on Census data and ONS predictions. It currently only does static microsimulation. The country wide assignment step, where the occupancy of individual households is predicted, is a computationally expensive step. This is best processed on a HPC system.
- SIMIM - Spatial Interaction Models of Internal Migration [4]. This package aims to build a flexible custom population projection generation tool based on spatial interaction models of internal migration within the UK. The tool models the impact of large and long-term infrastructure changes on population distribution and growth. It does this by taking base projections from the ONS and then trying to model the effects of new housing and work developments will have on these predictions.

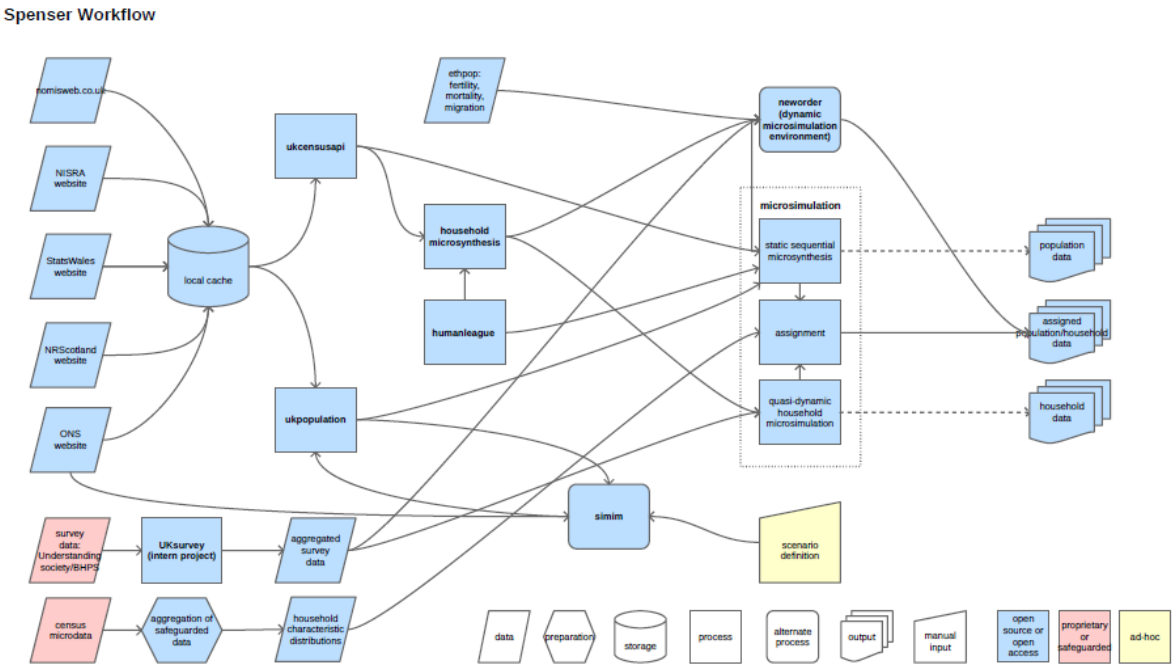

Figure 1: SPENSER workflow.

Pilot Objectives

Providing access and visualisation for the SIMIM model

Since SPENSER has many components it was not practical to include all of them in the pilot project. It was decided that the main focus would be on the SIMIM model which runs relatively quickly but requires a range of input parameters and gives several outputs which can usefully be visualised.

SIMIM can be used to predict changes in internal migration that likely to arise from major infrastructure projects. For example if it is known a large investment in new jobs and houses will occur in a specific region then it is likely that additional people will move to that location. The initial distribution of population is taken from the 2011 census data along with more recent updates from the ONS. In addition to this baseline data the model uses the default population predictions made by

the ONS of regional population growth over the coming decades. Growth is based on three key factors: fertility, life expectancy and migration. The ONS principle estimates of each of these is labelled the *ppp* assumption and used as the default in SIMIM. Other inputs to SIMIM model are:

- Coverage: the model can be run on the whole of Great Britain just England, Wales or Scotland
- Model type: the model of migration can be either gravity based (migration related to size of source and destination, inversely to distance) or production.
- Model sub-type: either a power law or exponential sub-model can be selected.
- Emitters: The source of migrations can be one or more of people, jobs or geographical areas.
- Attractors: The destination of migrations can be set as one or more of households, jobs, GVA or geographical area.

These configuration parameters define how the SIMIM migration model will run and are given as input in a JSON configuration file. In addition to these it is necessary to define the particular scenario that the user wishes to model. This is defined in terms of changes expected in areas to jobs, houses and GVA. The model resolution is to the LAD (Local Authority District) level. Typically a major investment to a certain region will be expected to create new housing and jobs in that region which will cause enhanced migration to the area. One such example is the proposed investment in the Cambridge-Milton Keynes-Oxford corridor (CaMKOx). Investment in new housing and infrastructure in this area would be expected to lead to greater migration and SIMIM can be used to estimate the scale of this. The input data to the model is the estimated additional growth in jobs, households and GVA in each LAD over the time period of the intervention. In the existing model this data is given as a CSV file of yearly changes to the values in the selected LADs.

Figure 2 illustrates the output of a SIMIM simulation of the CaMKOx scenario using the gravity migration model over the period 2015-2040. These were produced with the existing software in Python.

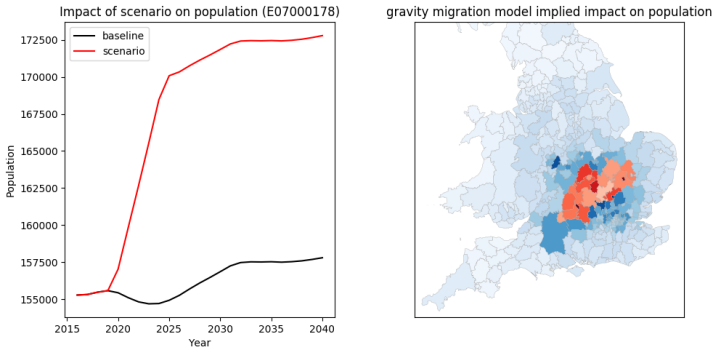

Figure 2: SIMIM Change in Oxford population due to enhanced migration (left) and overall regional changes (right).

The run time of the SIMIM model is modest, once the necessary data from remote sources has been loaded in the cache. However installation can be a time-consuming process requiring installation of Python and many supporting packages. Also, the definition of the configuration and the scenario files can be awkward for new users having to edit the JSON configuration file and set up a CSV file with LAD area codes. Visualisation of the results is also limited by the Python interface which currently just plots some predefined views of the data.

The main aim of the Pilot project in this case was to take the SIMIM model and implement it on DAFNI to allow easy remote access without the need to install any software on the client side. Instead a secure web interface has been developed which allows the user to run existing scenarios

or define new ones using an intuitive interactive map. A simple way to select the configuration parameters will also be provided. Runs can then be submitted to the DAFNI job processing queue with the results of each run saved into a database.

For visualisation the web interface will be extended to allow interactive plotting of LAD results from any model run along with new visualisation types.

Parallel execution of Microsimulation jobs

A second part of the Pilot will address more computationally intensive aspects population modelling. Both Microsimulation and NewOrder involve analysis that can take several hours to run, even when run in parallel on a HPC system. While DAFNI is not primarily designed to be a HPC system, it is a powerful computational resource offering hundreds of processing cores with a large amount of memory. The SPENSER microsimulation workloads are high throughput tasks since there is no communication requirement between jobs and hence it is well suited to the DAFNI platform.

The batch analysis of data for these tools tends to be a “one time” process that is only repeated occasionally, for example when updated results are available from the ONS. Hence a simple command line interface has been developed to run these parallel jobs.

Analysis of Software

SIMIM

The SIMIM package is written in Python and utilises a number of libraries for data manipulation and analysis, along with the SPENSER packages UkCensusAPI and UK Population. A typical run time for an analysis step is about 30 seconds on a high end CPU with the data cache pre-populated. The software makes some use of parallel processing in its analysis, but this is limited and may be in external libraries.

SIMIM uses data from the ONS and other sources to provide a projection of UK population and households at the LAD level out to 2040. These are accessed via the UKCensusAPI which caches the data fetched from remote sites to improve performance. The data from these sources is used to give a base line for internal migration based on past data and the projections used by the ONS, assuming a standard model of growth. The current choice is to use the principle population projection from the ONS data for the default LAD population estimates. Other projections may be made available in future releases.

SIMIM allows the user to choose the factors that will be used in the migration model, such as population, jobs, etc. and the driving forces which are dealt with using the Spatial Interaction modelling (SpInt) module in the python spatial analysis library (PySAL). A paper describing the model is available [1]. There are a number of commonly used models types which can be selected, including:

- Gravity – an unconstrained version of the model
- Production – where migration is limited by the migration origin flows
- Attraction – where migration is limited by the migration at destinations
- Doubly – where migration is limited by both origin and destination flows.

In addition there is the choice of exponential or power-based variation of the migration with the distance measure, referred to as the sub-model type.

The idea is to fit the selected model to the available migration data from the ONS using appropriate statistics. Having fitted the model to the existing data it can then be used to predicate how the future migration values will be changed by the expected changes to the drivers, such as jobs and housing in each LAD.

The output from the model includes an origin-destination (OD) matrix giving the migration flows between each LAD for every year of the scenario. The current software provides graphical display of the OD matrix, though this is not very easy to interpret due to the large number of LADs involved. More useful visualisation is the type shown in Figure 2 of the population variation in a selected LAD with and without the scenario, and the heat map of total population change at the end of the scenario period. It was suggested that CirCos plots [2] could be used as a better visualisation of the actual migration flows predicted for the new model.

Implementation of SIMIM within DAFNI

As with previous pilots, a Docker build was made of the SIMIM software directly from the github repository. This ensures the software runs in a well-defined Python environment with all the required libraries. A specific tagged release of the software is chosen to avoid unexpected updates which might break the inputs and outputs of the model.

For each run of the model two sets of inputs are required:

1. The definition of the scenario that is to be run in terms of the changes to the households, jobs, etc. in selected LADs over time and options to the model.
2. The set of ONS and related population data that must be fetched for the base projections of migration. This data only changes occasionally with mid-year updates.

The scenario definition data will be stored in a database for each job that is submitted to the system along with the job name. To save time the second set of data will be stored in the base Docker image using the SPENSER libraries that can cache the required data when they are first run.

SIMIM simulations will run as DAFNI jobs that are processed by the Kubernetes system through a queue that reads the scenario information from the database and writes the results back into the same database. This is the same structure that has been successfully used in other pilots.

The user is provided with a web-based user interface that allows specification of the job to be run in terms of the expected scenario and the model options. When these are fixed the job is submitted to the queue and the interface monitors the job until completion. Multiple jobs can be submitted at the same time to investigate different scenarios.

Results Visualisation

When jobs have been completed the results can be inspected through the web-based interface. Three types of visualisation are provided:

1. A heat map of the change in population at the LAD level that is predicted from the scenario. This data will be a function of time through the scenario so a slider is provided to select the year that the results should be displayed for.
2. The change in population of a selected LAD due to the scenario can be shown as a plot against time. The LAD to be shown can be interactively selected from the heat map display.
3. A “chord” visualisation of the change in migration between selected LADs. This plot is based on the Circos methods [2] and allows the user to select a set of regions to be displayed, since trying to display all LADs simultaneously would not be practical or informative.

Examples of the UI are shown in the following figures. Figure 3 shows the welcome screen. Data for new scenarios can be defined on the page shown in Figure 4. The visualisation of results is illustrated in Figures 5 and 6.

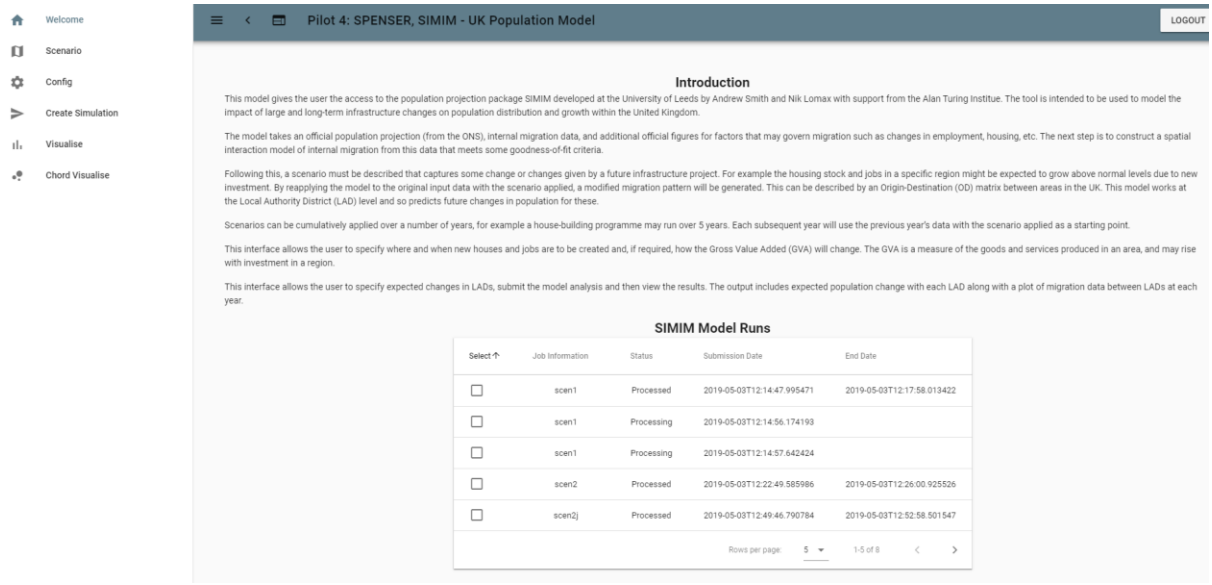

Figure 3: The welcome screen shown after login. This also has the table of jobs that have been run. Completed jobs can be selected for visualisation.

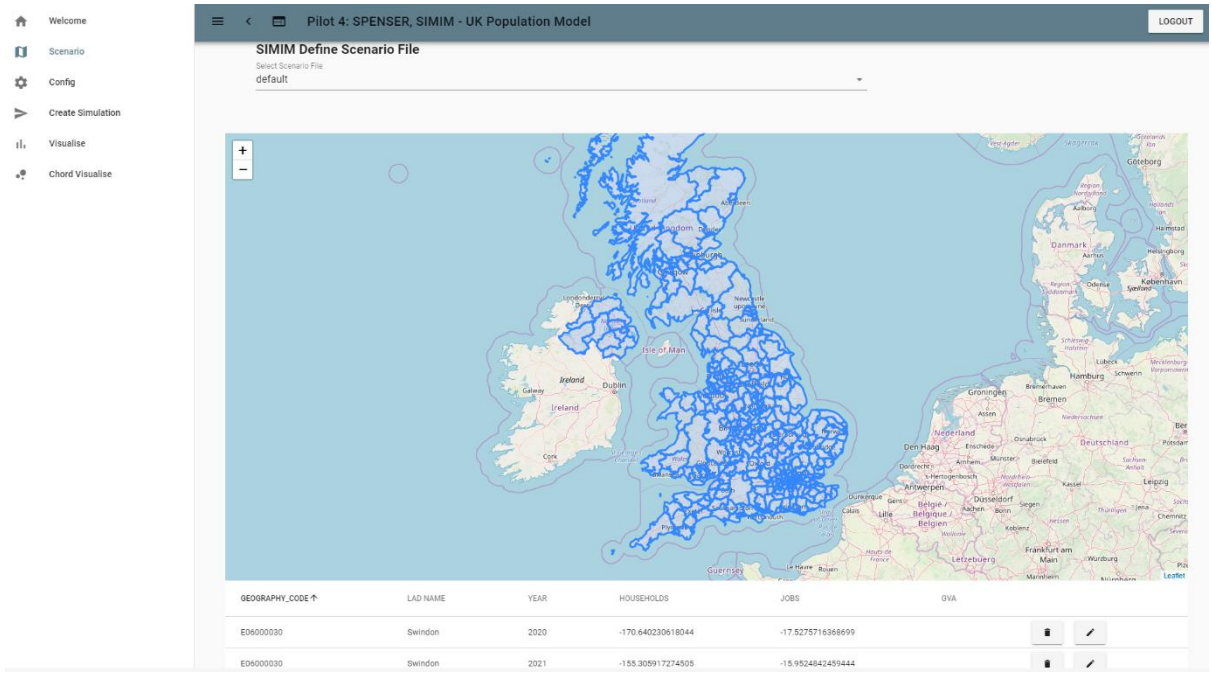

Figure 4: The scenario definition page. A new scenario can be defined on this page by selecting the LAD then giving the number of additional households, jobs, etc. that are expected, plus the years this will occur over. It is also possible to read existing scenario files.

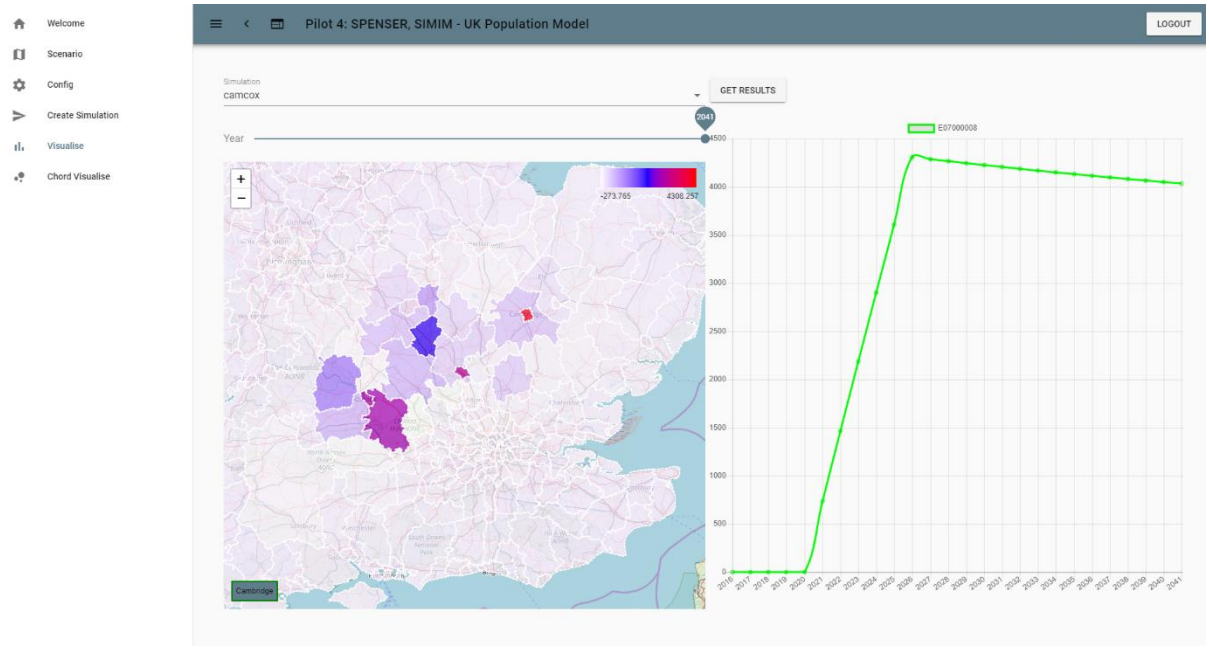

Figure 5: Visualisation of the output of a simim scenario for development of the Oxford-Milton-Keynes-Cambridge corridor. The heat map (left) gives the change in population at the selected date by LAD. The graph on the right is the population for the selected LAD through the scenario period.

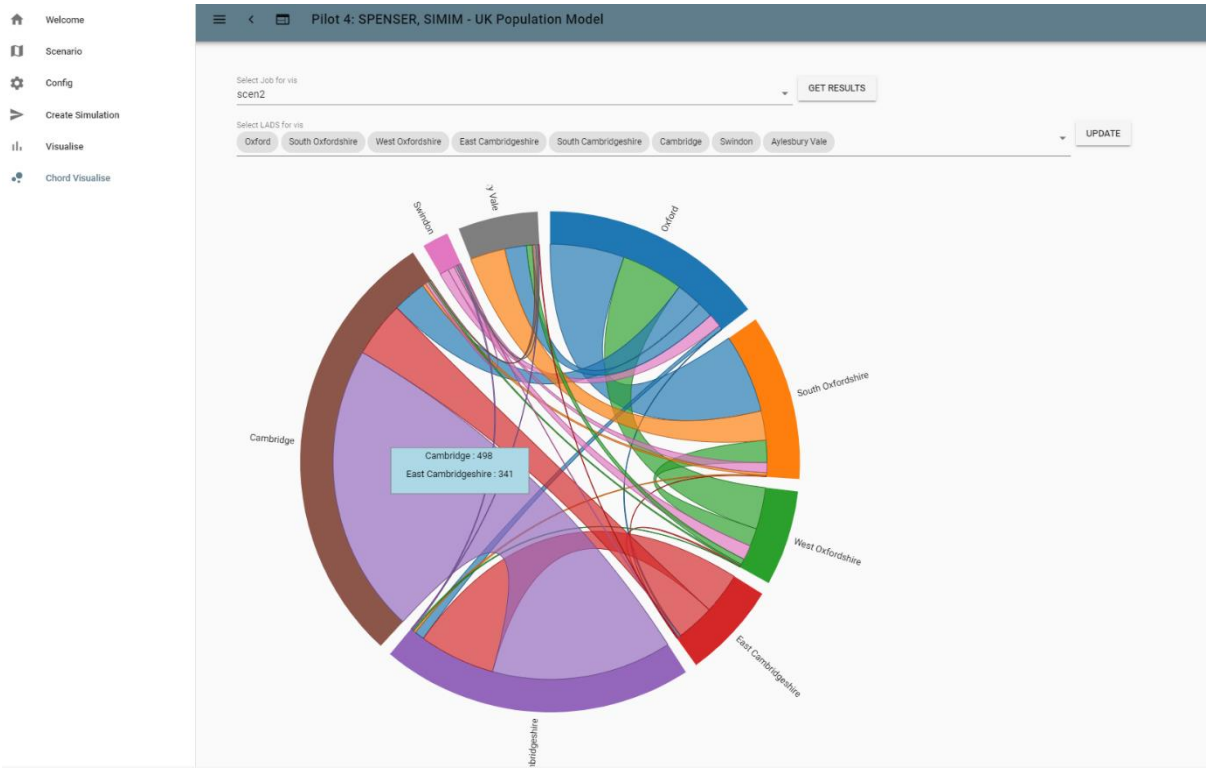

Figure 6: A plot showing the migration between selected LADs in the new scenario. Each chord is coloured by the major flow direction. The mouse can be used to show the actual numbers for each chord. LADs of interest can be selected in the UI.

Batch Processing of Microsimulation

The microsimulation component of SPENSER, shown in Figure 1, is used to create synthetic data for the households, etc. within each area of the UK. This calculation is usually run on census data, and so

is only updated occasionally. However the computation is quite significant and is run in parallel on a HPC cluster at the University of Leeds. The calculation is broken down by LADs and is in fact "embarrassingly parallel", i.e. no communication of information is required between separate LADs, so the jobs can run independently of each other.

This processing step does not require a UI since there are no user options to vary. A simple command line interface to run the job has been developed. The workload consists of one job for each of the UK LADs, approximately 400 in total. However the computational work for each LAD depends on the population that is contained within it. As the range of population in LADs is from a few thousand to over one million (Birmingham) there is a need to load balance the computation so as to efficiently process this work load. Measurements show that the computational cost is not a simple linear function of the population size, but includes a quadratic term. Hence a partitioning algorithm has been developed that more accurately takes into account the run time of each job.

In the current DAFNI system it is simple to allocate 32 cores to a task using an Argo workflow [3]. The separate LADs can be split into 32 groups with load balancing, and processed to produce the synthetic distributions for each LAD.

The run time for processing a microsimulation in this way is found to be 31 minutes on 32 cores. This compares with 91 minutes that was measured previously on the Leeds HPC system. The improvement in run time may be partly due to using a load-balancing algorithm; though the exact number and type of CPUs employed on the Leeds HPC system was not given. Nevertheless this illustrates that large high-throughput compute jobs can be efficiently processed on the current hardware.

Batch Processing for NewOrder

The dynamic microsimulation of NewOrder allows the user to write new models in Python, while implementing the computationally intensive parts of the computation using C++ libraries. For time-consuming calculations, such as the microsimulation of people in the whole of the UK, the code makes use of MPI for parallel execution. MPI is a standard parallel environment that is implemented on virtually all HPC systems to enable jobs to execute in parallel with efficient communication between separate instances. For NewOrder the computation can be partitioned by regions, as for the previous microsimulation case, which means that there is virtually no communication between tasks. Hence these computations are again "embarrassingly parallel" and do not require fast communication between separate jobs.

To run NewOrder in parallel on the DAFNI Kubernetes based cluster, it is necessary to load an MPI library that will manage the parallel processes and then allocate them to a node with a sufficient number of cores. On the current system this allows easy scaling to 32 processes, the maximum number of cores on a cluster node. To scale beyond this would require an MPI implementation that runs across multiple nodes. Some projects are starting to address this issue, for example Kubeflow and Kube-OpenMPI, but these are not yet supported on the DAFNI cluster. Hence we present results using NewOrder on up to 32 cores.

Number of cores	Elapsed time(secs)	Speed up (wrt 2 cores)	Efficiency
2	891.4	1.00	1.00
4	289.7	3.08	1.03
8	98.0	9.10	1.30
12	59.3	15.0	1.36
24	26.4	33.8	1.47

32	21.7	41.1	1.36

Table 1: Performance of NewOrder with varying number of cores on the DAFNI cluster. The efficiency has been calculated assuming that one processor is dedicated to management tasks.

The parallel efficiency is significantly greater than unity, which is surprising. So called super-linear speed up can occur when adding additional cores also adds more cache memory which the partitioned algorithm can exploit. Another possible cause of the super-linear speed up could be the load balancing of the tasks. Investigation of which effect is most important would require adding detailed monitoring into the code.

Conclusions

Models of population at both national and local levels are vital to many aspect of future infrastructure planning. SPENSER offers a wide range to population models for the UK which take existing data, such as ONS census data and projections, and builds detailed models future outcomes. In this Pilot we have demonstrated how DAFNI can provide a simple user interface to exploit SIMIM scenarios and visualise the predictions using a web-based interface.

In addition this pilot has shown that DAFNI compute infrastructure can be used for long-running high-throughput computations. Using an updated partitioning method, 32 cores on DAFNI has shown a significantly improved run time over the HPC processing that was available to the developer on a local system.

References

1. Taylor Oshan (2016). *A primer for working with the Spatial Interaction modeling (SpInt) module in the python spatial analysis library (PySAL).* Retrieved from http://openjournals.wu.ac.at/region/paper_175/175.html

2. What is Circos?: http:// http://circos.ca/
3. Argo: https://argoproj.github.io/
4. SIMIM: https://github.com/nismod/simim
5. NewOrder: https://github.com/virgesmith/neworder
6. UKCencusAPI: https://github.com/virgesmith/UKCensusAPI
7. UKPopulation: https://github.com/nismod/ukpopulation