In contrast to at EPCC (Section 4.1), the workload at ARCTUR HPC centre is characterized by a distinct "on-demand" utilisation pattern

Modelling HPC workloads with (some) Machine Learning

Application to HPC Benchmarking

Antonino Bonanni, Simon D. Smart and Tiago Quintino

[1] European Centre for Medium-Range Weather Forecasts, Reading, UK

NEXTGenIO Partners: EPCC, ECMWF, ARCTUR, INTEL, Fujitsu, ARM, BSC, TU-Dresden

Introduction

HPC benchmarking is traditionally performed by testing HPC subsystems in isolation with ad-hoc benchmarking applications and then combining the results to estimate the full-system performance. The main limitation of this approach is the inability to effectively predict and estimate system-wide bottlenecks (e.g. I/O bottleneck) caused by the interaction of several sub-systems. To address this problem, a software named Kronos has been developed at ECMWF. Kronos models and then executes a realistic workload on a target HPC system, benchmarking compute, network and storage sub-systems at the same time. The workload modelling comprises several steps, from jobs profiling data to clustering and finally translation into portable and easily deployable benchmarking applications. Kronos is being developed as part of the NEXTGenIO project which is a 4-year EU-funded Horizon 2020 project started in October 2015. NEXTGenIO is coordinated by the Edinburgh Supercomputing Centre (EPCC) and involves partners from several European countries.

HPC Workload Modelling

Kronos analyses workloads of real HPC systems and generates synthetic workload models through a modelling process (Fig. 1). These synthetic workloads can then be executed as a schedule of simple and highly portable *"synthetic"* applications on a target HPC system.

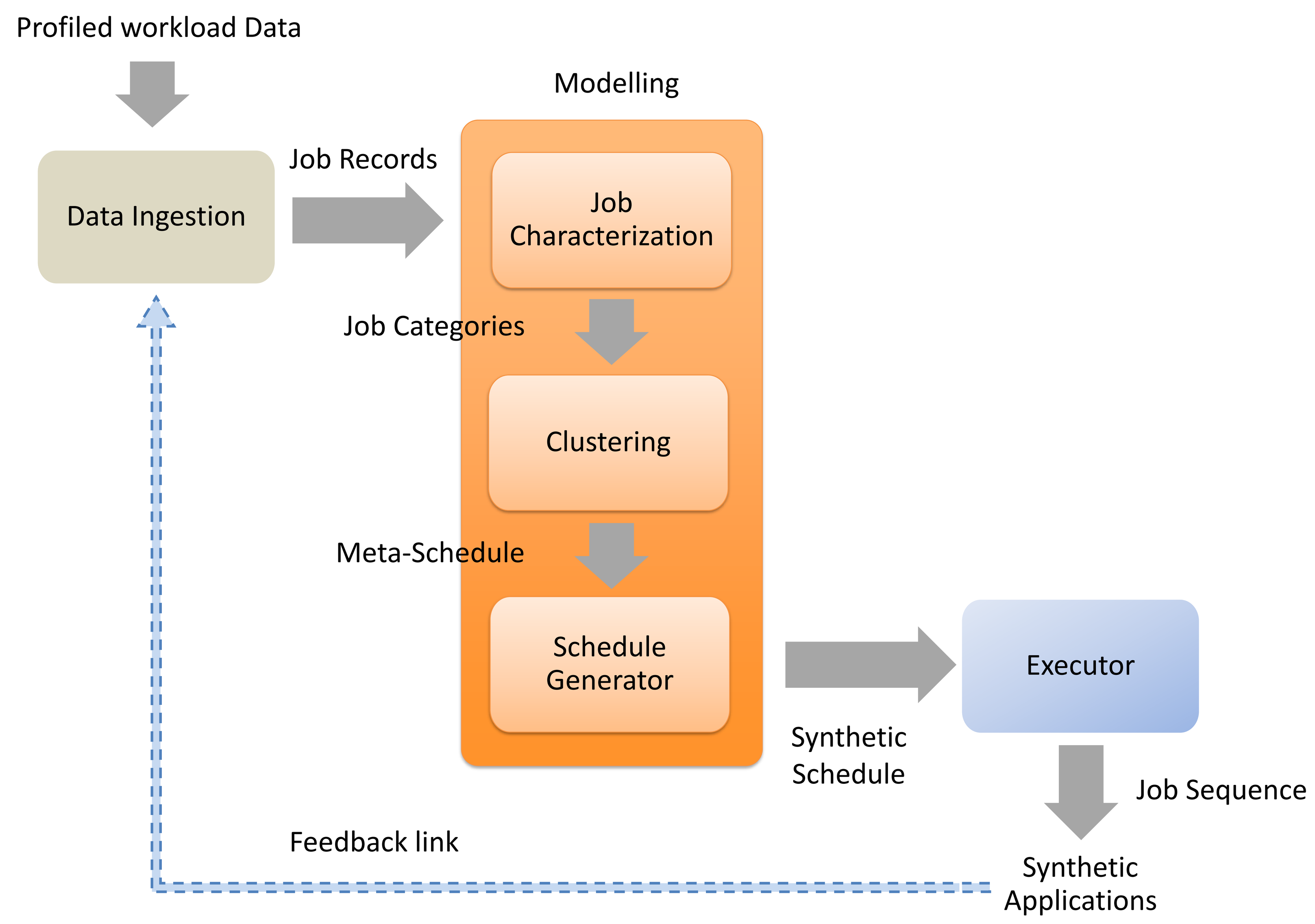


Figure 1: Kronos modelling process.

1. ***Data ingestion***. Workload profiling data is collected by profiling tools. This data contains job records with metadata and time-dependent profiled values corresponding to a series of defined HPC metrics (Figure 2).
2. ***Job characterization***. Workload data is generally (very) sparse; only a subset of metrics are available for each of the jobs and Machine Learning techniques are explored to fill in the incomplete information.
3. ***Clustering***. Jobs are clustered according to their associated data such that jobs with similar characteristics form clusters from which a reduced set of prototypical jobs can be formed. These are then used to spawn the synthetic applications.
4. ***Schedule generation***. A schedule is built for abstract, idealised synthetic applications to reflect the profiled workload. This takes into account any modification or scaling factors passed in by the user.
5. ***Execution***. The abstract schedule is passed to the Kronos Executor. This submits concrete synthetic applications to the scheduling queue of a target HPC system.
6. **Feedback.** Profiling data collected when running the synthetic applications on the target HPC system can optionally be fed back into the modeller to fine-tune the target parameters.

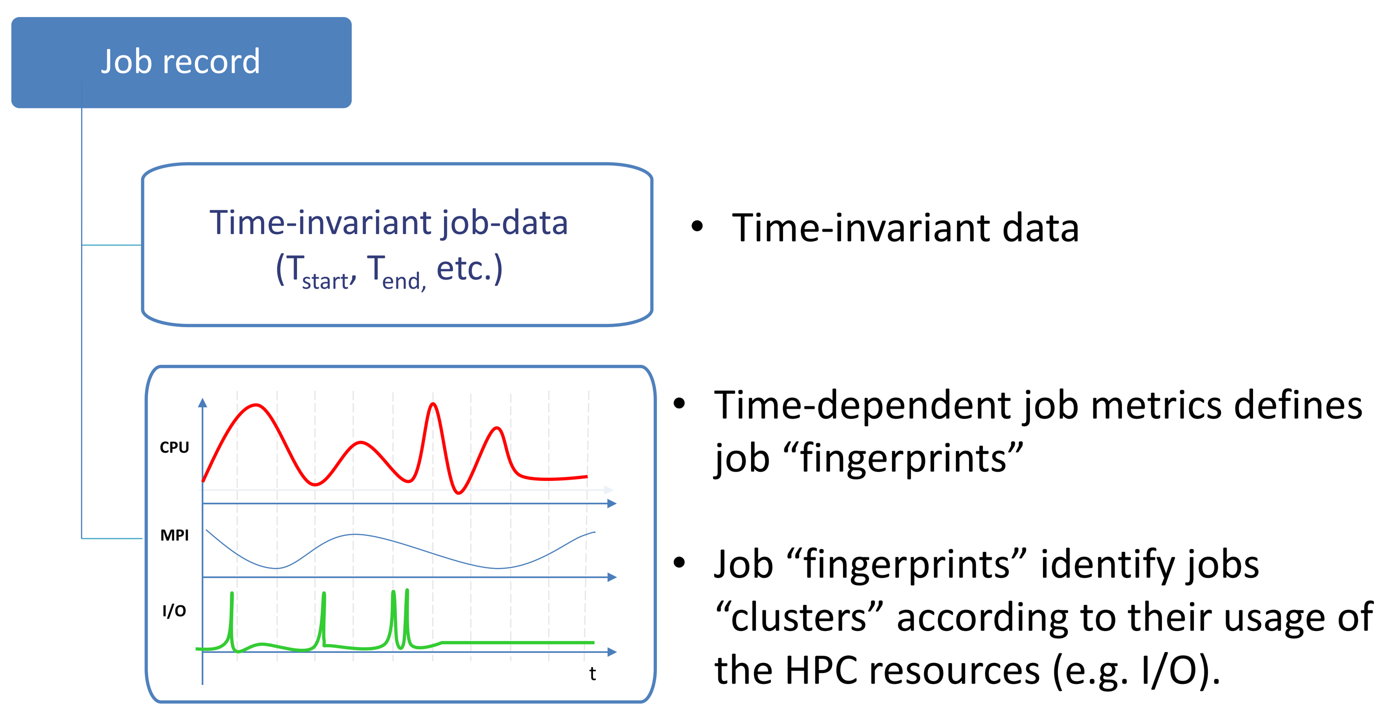


Figure 2: Time-invariant and time-dependent job data.

From HPC workload profiles to benchmarking models

To provide Kronos with real-life input data, a vast workload profiling exercise was carried out in 2016 as part of the NEXTGenIO project at **ECMWF**, **EPCC** and **ARCTUR** in collaboration with **ARM** and **TU-Dresden** (examples in Fig.3 and Fig.4).

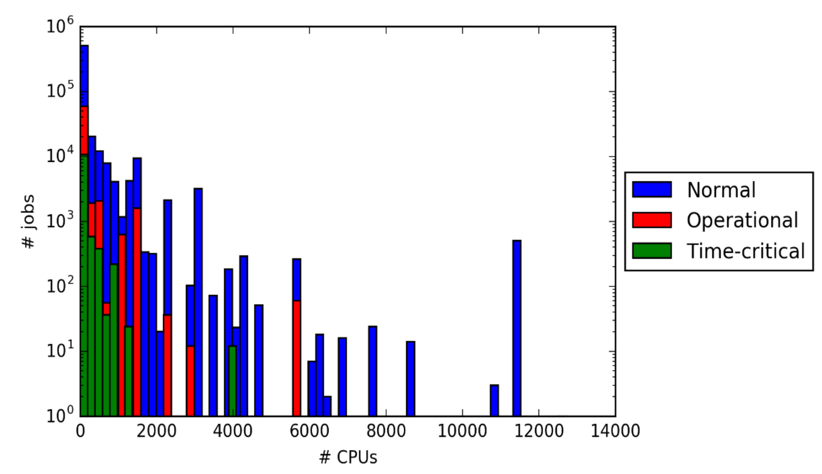


Figure 3: Histograms of jobs run at ECMWF centre categorized according to the type of workload.

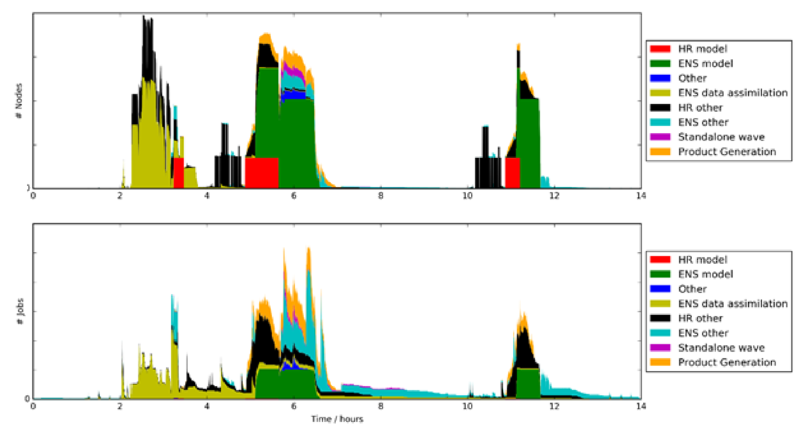
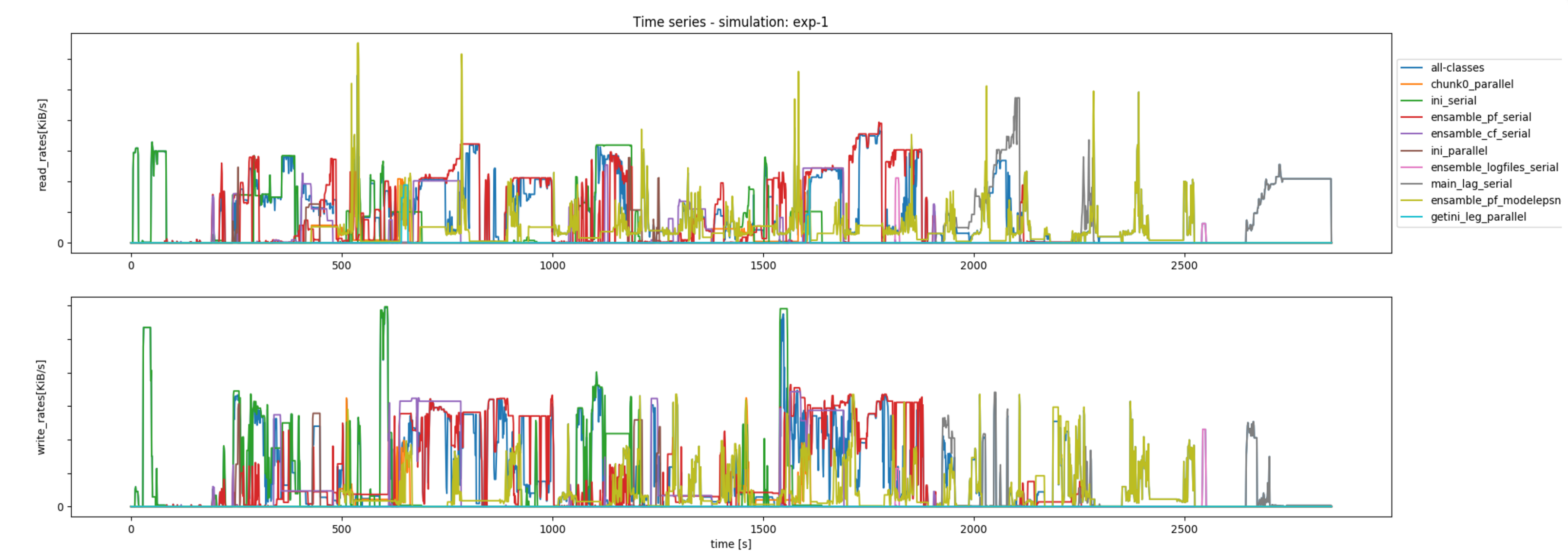
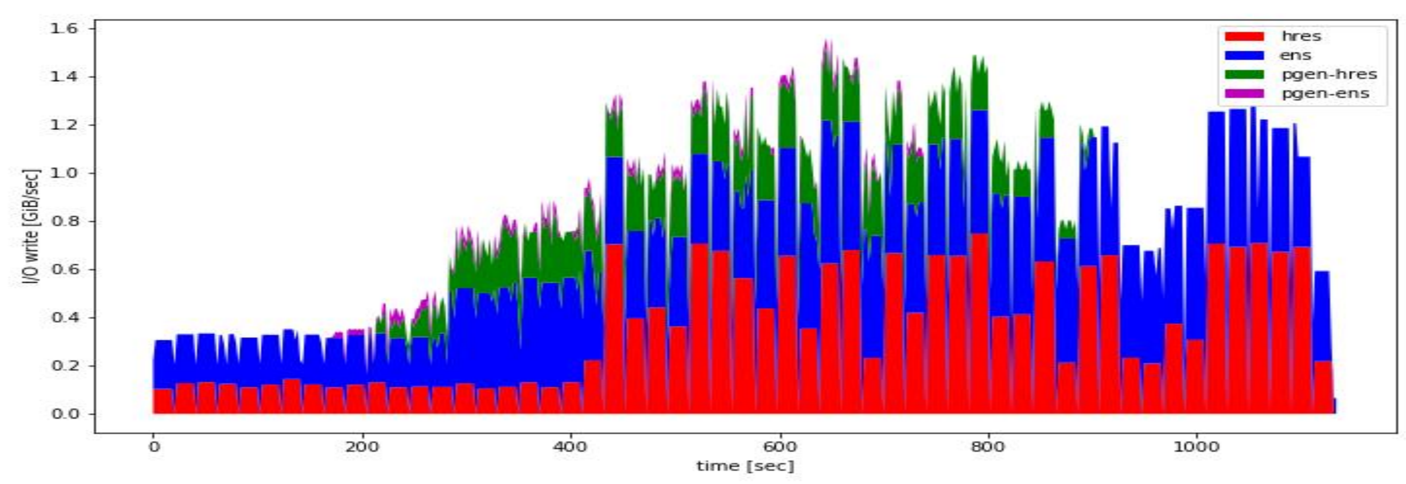


Figure 4: Example of timeline workload profiled collected at ECMWF (#jobs and #processors – sanitized values)

Kronos process: From workload data collection to modelling and simulation



Example 1: Full-scale workload generated from modelling a representative ECMWF weather forecast research experiment (executed with 341 synthetic applications of various types).



Example 2: Example of I/O footprint of a small-scale model of the ECMWF forecast operational workload. Kronos is here used to drive a workflow of real applications: Main IFS models (High-resolution and Ensembles) and Product Generation jobs.

Summary and Conclusions

Kronos software has been developed within the NEXTGenIO project with the intent of exploring HPC benchmarking by taking into account not only the HPC system subcomponents in isolation but also their interactions. This approach allows a more comprehensive benchmarking and possibly highlights system bottlenecks (e.g. I/O bottleneck). The development comprised collecting workload data from three HPC centres and then exploring various modelling strategies including machine learning algorithms for job classification and missing data estimation (through recommendation system algorithms). Kronos is currently being used by ECMWF for the procurement of their new HPC system that is due to become operational by 2020.

Further Reading

1. NEXTGenIO project (GA: 671951): *www.nextgenio.eu*
2. *ECMWF: https://www.ecmwf.int*
3. ARM tools: https://www.arm.com/products/development-tools
4. Score-P. from *http://www.vi-hps.org/projects/score-p*