LARGE-SCALE COMPUTATIONS WITH QCG-PILOTJOB

Bartosz Bosak[1], Piotr Kopta[1], Michał Kulczewski[1], Tomasz Piontek[1], Paul Karlshoefer[2], Erwan Raffin[2], Jalal Lakhlili[3], Hamid Arabnejad[4], Derek Groen[4]

[1]Poznań Supercomputing and Networking Center
{bbosak, pkopta, piontek, michal.kulczewski}@man.poznan.pl

[2]Center for Excellence in Performance Programming ATOS
{paul.karlshoefer, erwan.raffin}@atos.net

[3]Max-Planck Institute for Plasma Physics - Garching, Germany
jalal.lakhlili@ipp.mpg.de

[4]Brunel Univesity London
{Hamid.Arabnejad, Derek.Groen}@brunel.ac.uk

CHALLENGES

The access to HPC systems is regulated by the policies of resource providers and restricted by local resource management system configurations and their implementations. These regulations give preference to massively parallel simulations, e.g. based on MPI, and are targetted at users who submit only a few large jobs at a time. For users who want to run concurrently a huge number of conceptually different tasks, the regulations are problematic.

The simplistic approach to mitigate the limitations present on HPC machines is to define a customised processing scheme for multi-task execution in a scripting language like bash or python and run it inside a single allocation of queing system. It can resolve a problem for short-term, but since such a script is typically a *proof-of-concept* and therefore prone to many bugs and inefficiency, it is not sufficient for sustainable usage.

SOLUTION: QCG-PILOTJOB

QCG-PilotJob is a handy, user-space tool that brings an additional tasks management level within the already created allocation and without any external services

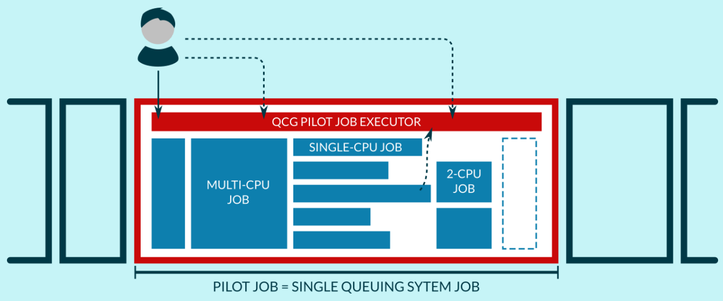


Figure 1: The basic idea of QCG-PilotJob. The Executor service supervises tasks run within an allocation. It can be dynamically accessed using Python API or remote interface

QCG-PilotJob is developed by PSNC

From the job scheduler's perspective, QCG-PilotJob is only a single regular task, but for a user it is a second-level, fully-managable lightweight queuing system he can launch on its own

Users can interact with QCG-PilotJob in a similar way as with any queuing system. They can submit new tasks, list tasks or cancel them. QCG-PilotJob ensures that tasks are executed efficiently

QCG-PilotJob allows to define dependencies between tasks and create iterative tasks, as well as it provides a built-in resume mechanism to support fault tolerance at a workflow level

TOWARDS EXASCALE

The ultimate aim behind QCG-PilotJob is to propose scallable and efficient tool, keeping its usage easy. In order to achieve it, the following target architecture has been defined:

- a single instance of top-level tasks Scheduling Queue, being a main entry-point to the pilot job,
- one or many Executor services responsible for management of tasks inside particular allocations,
- Node Launchers associated with elementary nodes and dedicated for efficient starting small tasks.

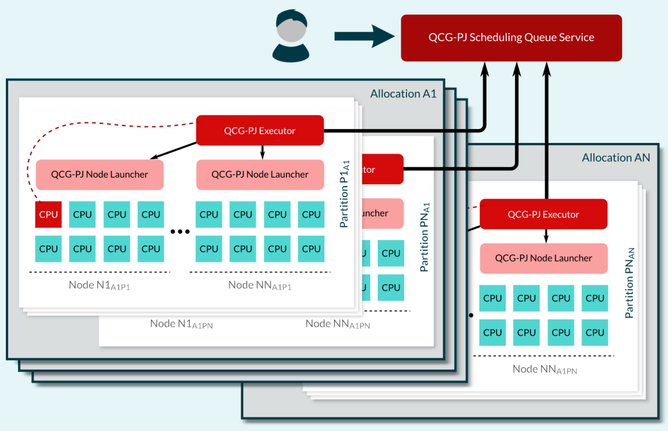


Figure 2: Hierarchical architecture of QCG-PilotJob aimed to provide good scallability, performance and flexibility of the system.

PERFORMANCE EVALUATION

To ensure high performance of QCG-PilotJob, we are in the process of its validation on large-scale supercomputing machines such as SuperMUC-NG at LRZ and Eagle/Altair at PSNC where tests with 48 000 cores at Altair@PSNC are already being set-up.

So far we run up to 6000 pilot job's tasks on a single SLURM allocation containing 1920 CPU cores. For tasks which take in average 300 seconds to complete, merely about 0.8% of the execution time is spent in the QCG-PilotJob Executor to govern the execution.

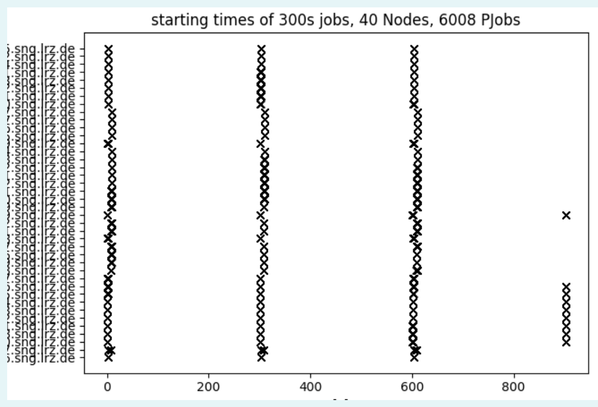


Figure 3: Performance evaulation results from SuperMUC-NG. For the investigated scenario there are practically no delays caused by QCG-PilotJob

QCG-PILOTJOB AS PART OF VECMA TOOLKIT

The goal of the VECMA project is to enable Verification, Validation and Uncerainty Quantification (VVUQ) for a diverse set of large-scale applications, multiscale and multiphysics in particular. The project delivers an open source toolkit called **VECMAtk**, which includes components for automatisation of VVUQ procedures and efficient execution of required computing tasks on peta- and emerging exa-scale machines. **QCG** tools and services are an integral part of the toolkit, while the role of **QCG-PilotJob** is essential for the flexible parallelisation of computations.

EasyVVUQ
offers reusable VVUQ techniques for different kinds of applications

EasySurrogate
facilitates creation of surrogates for multiscale simulations.

FabSim3
is an automation toolkit for complex simulation tasks

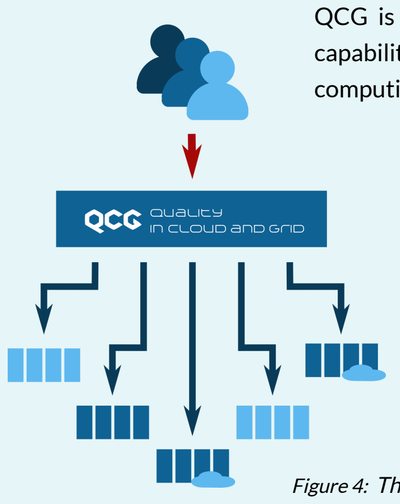
MUSCLE3
is a coupling library for multiscale applications

QCG
enables simple access to HPC systems and makes computing efficient

Integration with QCG-PilotJob via **EQI** library

Integration with QCG-PilotJob via **QCG-Client**

QCG: QUALITY IN CLOUD AND GRID



QCG is a fully-featured solution developed by PSNC that provides advanced capabilities for the unified execution of complex jobs across single or multiple computing resources

QCG offers well-defined remote interface enabling access to local resource management systems, SLURM in particular

In order to satisfy different needs of users QCG provides diversified set of client tools: graphical (QCG-Portal, QCG-Now) and commandline (QCG-Client)

This ecosystem is complemented by **QCG-PilotJob**, which effectively circumvent natural limitations of batch systems

Figure 4: The basic idea of QCG as a system integrating many computing resources

USE CASES

UrbanAir

PSNC develops the UrbanAir application with the aim of instant prediction of pollution in urban areas. This is a challenging field not only because of algorithmic complexity, but also because of the uncertainty of many inputs. In order to perform Sensitivity Analisis for the application to figure out optimisation points, the authors has applied VECMA toolkit. With help of EasyVVUQ and QCG-PilotJob, it was possible to execute sufficient number of evaluations and find the most sensitive variables.

Fusion

Researchers from Max-Planck Institute for Plasma Physics need to quantify uncertainties in turbulence code being a part of fusion application. This is computationally expensive 3D parallel code and needs 512 to 16384 MPI processes. Running UQ with such a model requires a very large number of computing jobs. In a result of works performed within the VECMA project, it was possible to execute required simulations with QCG-PilotJob in a single batch allocation.

CovidSim

QCG-PilotJob has been integrated into FabSim3 automation toolkit in order to support FabSim's application campaigns that require pilot job functionality. One of already implemented scenarios was Uncertainty Quantification performed with EasyVVUQ on CovidSim epidemiological code aimed to evaluate the quality of the results produced by CovidSim and their usefullnes for epidemiological forecasting in UK. The results of this work have been published in Nature Computational Science.

REFERENCES

- D. Groen et al. “VECMAtk: A Scalable Verification, Validation and Uncertainty Quantification toolkit for Scientific Simulations”, *Philosophical Transactions A*
- D. W. Wright et al. “Building confidence in simulation: Applications of EasyVVUQ”, *Advanced Theory and Simulations*, 2020
- W. Edeling et al. “Model uncertainty and decision making: Predicting the Impact of COVID-19 Using the CovidSim Epidemiological Code”, *Nat Comput Sci*, 2021
- T. Piontek et al. "Development of Science Gateways Using QCG — Lessons Learned from the Deployment on Large Scale Distributed and HPC Infrastructures", *JGC* (2016)
- QCG Tools and Services for VECMA (movie) - https://www.youtube.com/watch?v=sxuaZD6mrrw
- QCG-PilotJob: https://github.com/vecma-project/QCG-PilotJob

FUTURE WORK

- Finishing the implementation of Scheduling Queue Service
- Development of the monitoring service for tracking progress of executions and presenting various statistics
- Closer integration of QCG-PilotJob with EasyVVUQ
- Portability tests on several high-end European clusters and further scalability tests

The work was funded as part of the European Union Horizon 2020 research and innovation programme under grant agreement nos. 800925 (VECMA project; www.vecma.eu)

Runtime of the project:
15/06/2018 to 14/12/2021

The performance tests were performed on Eagle cluster at the Poznan Supercomputing and Networking Center (PSNC), SuperMUC-NG cluster at Leibniz Supercomputing Centre (LRZ) and MARCONI supercomputer at CINECA.