MANCHESTER 1824
The University of Manchester

Linking coupled water-energy engineered system simulation models to HPC resources via a generalised web-interface.

Stephen Knox, **James Tomlinson** and Julien Harou

MANCHESTER
1824
The University of Manchester

Contents

- Introduction to large infrastructure decision making.
- Parameterising models with web interfaces and micro-services.
- Multi-objective optimisation.
- Ongoing work with Stefan Kollet

The University of Manchester

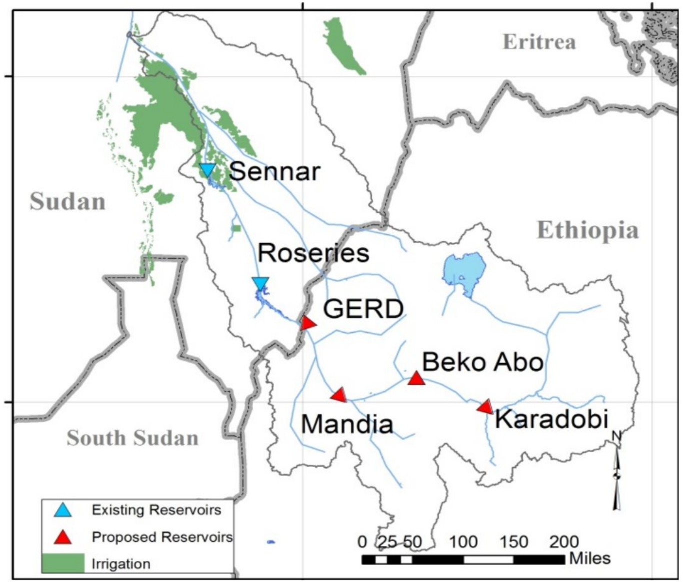
Introduction

Collaborative design and operation of large infrastructure systems.

- No longer recommend the single "optimal" solution.
- Explore the trade-offs.
- Participation of many stakeholders
  - Supported by numerical analysis and system modelling.

Achieving these goals requires,

- The **construction** of complex system models
- The robust **evaluation** of designs
- **Stakeholder interaction** with results.



System simulation based modelling

Traditional approaches are too simplistic

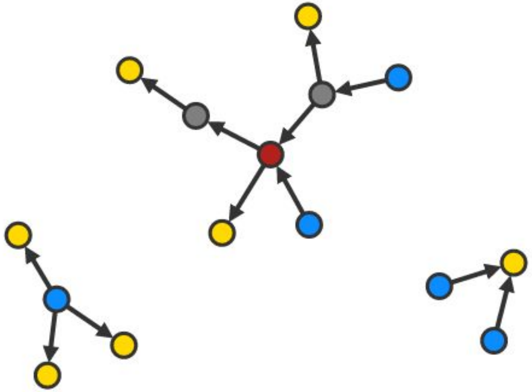
- Single objective, linear
- Minimise cost, supply >= demand.
- Typically solved using a mathematical programming (e.g. MILP).

Decision making under uncertainty (DMU) can help

- Couples with simulation models.
- Multiple metrics of system performance.
- Thousands of scenarios to determine robustness

MANCHESTER 1824
The University of Manchester

Constructing system simulators



Simulation models look like

- A topology of interconnected nodes and links.
- Parameters on components
  - boundary conditions, constraints & other data.

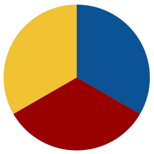
Modellers parameterise the system

- This process should be easy to use and intuitive.
- Expose the details of the underlying model's implementation.

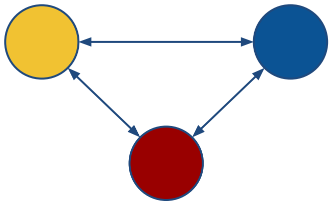
Historically used a tight coupling of model and graphical user interfaces (GUI) in to a single software package.

Separating concerns

Monolithic architecture



Micro-service architecture



Separate the model, data and user interface

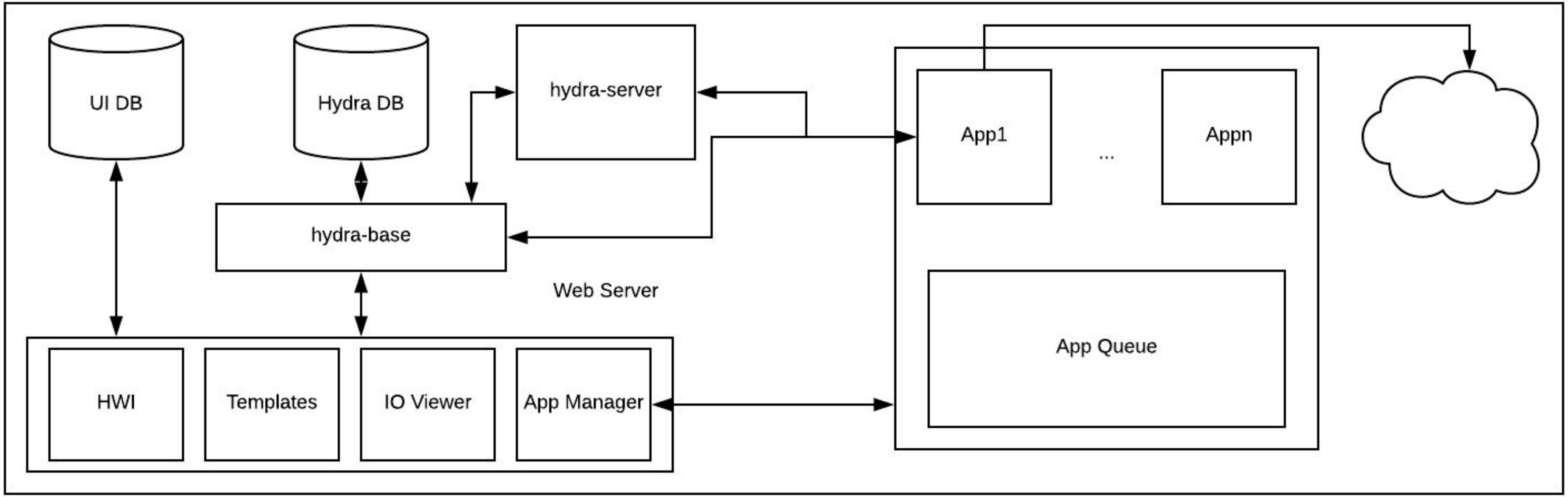
- Approach taken by Hydra Platform and its web interface.
- Generic map-based interface coupled with a database backend.
- Models interact through the “applications” that read and write data.

Can be reused for multiple models

- Model specific customisation through a templating system.
- Templates describe:
  - the types of node and link that are available.
  - the data inputs required for component.

MANCHESTER 1824
The University of Manchester

Hydra system architecture



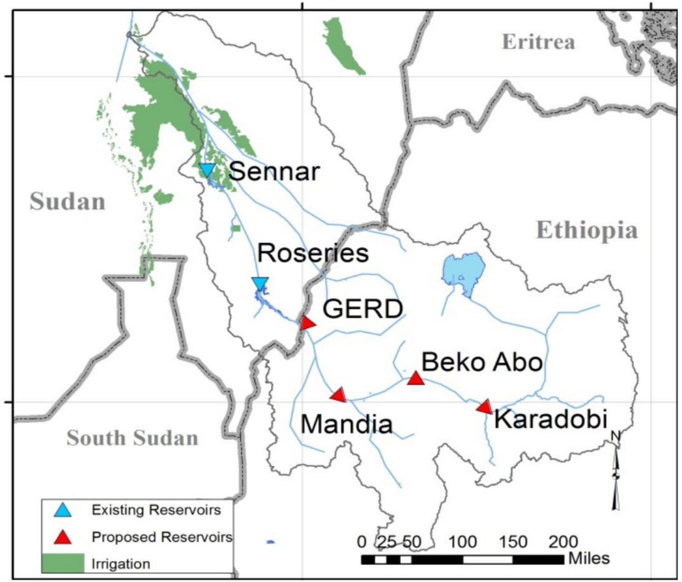
HPC requirement

Discovering trade-offs in complex systems is a general optimisation problem.

- General constrained optimization problem with multiple objectives.
- Objective and constraint values are computed by the system simulator(s).
  - Can take several minutes (or longer) per execution.
- Typically solved using multi-objective evolutionary algorithms (MOEA).

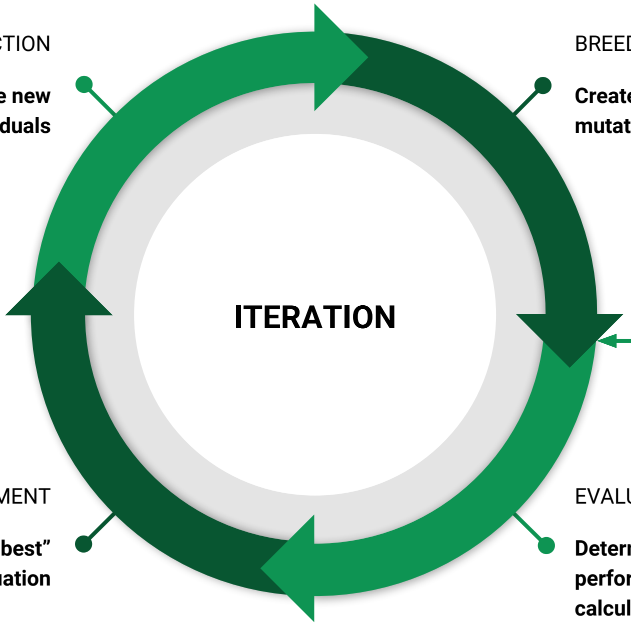
Computational intensive to rerun a simulator over multiple scenarios.

- Require HPC to perform system (population) evaluation in parallel.



MANCHESTER 1824
The University of Manchester

Multi-objective evolutionary algorithm (MOEA)



SELECTION

Selection the parents from the new population to breed new individuals

BREEDING

Create new individuals by applying mutation and crossover to parents.

INITIAL POPULATION

Generate an initial population of individuals to seed the search.

EVALUATION

Determine how well new individuals perform. I.e. run the simulation and calculate objectives (fitness)

REPLACEMENT

Update the population with the “best” performing individuals post evaluation

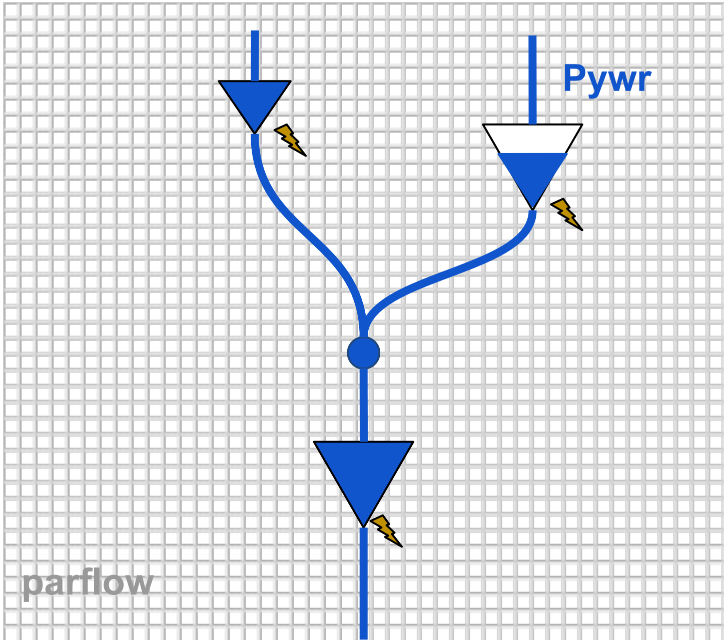
MANCHESTER 1824
The University of Manchester

Microservice framework for distributed parallel MOEA evaluation

Current research linking Hydra to distributed MOEA micro-services.

- Typical use case has many independent long (minutes) evaluation times.
- Problem looks more like a distributed task queue
  - Not a traditional parallel MPI job.
- MOEA simulation managed by the Hydra framework through an “app”.
  - Simplifies user experience with a graphical interface
  - Fault tolerant and dynamic job queues handled by the app

Integrated simulation with parflow, Pywr and Hydra



Work in progress collaboration with Stefan Kollet

- Detailed catchment flow simulation with parflow
- Linked to water system simulator
  - Written in “Pywr”
  - Parameterised with Hydra
- Both coupled to MOEA; optimise:
  - Reservoir release rules (Pywr)
  - Land-use variables (parflow)
  - Maximise hydro-power output
  - Maximise reservoir resilience
- Computationally expensive problem!

MANCHESTER
1824
The University of Manchester

Summary & vision

Utilise “Web 2.0” technologies to create a platform that can,

- Create, parameterise and deploy models without looking at code.
  - Including HPC infrastructure.
- Facilitate advanced decision making methods.
  - Stakeholder led results exploration.
- Integrate with complex hydrological system (and other) models.
  - Explore land-use change and other variables.
  - Impacts on other sectors, e.g. energy, food.

MANCHESTER
1824
The University of Manchester

Visit us on GitHub

All our code is open source and we welcome collaboration.

Hydra:

https://github.com/hydraplatform

Water simulation library Pywr:

https://github.com/pywr/pywr/

Email me:

james.tomlinson@manchester.ac.uk

Thank you for listening.

Any questions?