Bringing Long Running Geostatistical Calculations to Public Health Professionals in the Developing World

Tom Nicholls
Research Software Engineer

Centre for Health Informatics, Computing and Statistics
Lancaster University
Coming up...

- The Science
- The problem
- Proposed Architecture
- Current Prototype
The Science – NOT MY WORK!

• “Neglected Tropical Diseases” are a set of 7 serious diseases affecting developing countries. The WHO has a target to eliminate 7 of these diseases by 2030.

• Statisticians in our team have developed geostatistical methods* of analysing prevalence data of cases of each of these NTDs.

• In short the methods are key for designing efficient surveys

• We need to make these calculations available as a web application to health workers in the developing world.

*e.g. Claudio Fronterre and others, Design and Analysis of Elimination Surveys for Neglected Tropical Diseases, *The Journal of Infectious Diseases*, Volume 221, Issue Supplement_5, 15 June 2020, Pages S554–S560, [https://doi.org/10.1093/infdis/jiz554](https://doi.org/10.1093/infdis/jiz554)
Survey Design 😊

### Baseline

- Any STH prevalence (%):
  - 0 - 2
  - 2 - 3
  - 3 - 7
  - 7 - 10
  - 10 - 14
  - 14 - 17
  - 17 - 21
  - 21 - 25
  - 25 - 31
  - ≥ 31

### Predictions at Baseline

- Predicted mean prevalence (%):
  - 0 - 2
  - 2 - 3
  - 3 - 7
  - 7 - 10
  - 10 - 14
  - 14 - 17
  - 17 - 21
  - 21 - 25
  - 25 - 31
  - 31 - 59

---

Lancaster University

---

Coordinates (km):
- 0
- 50
- 100
- 150
- 200
- 250
- 300
Existing “R Shiny” Application

Browser

REQUEST

output.html

RESPONSE

baseline.csv

schools.csv

mda.csv

counties.gpkg

UI

R Script

Geospatial
The Technical Problem

- Geospatial Calculations are very long-running: these can run for hours.
- Difficult to scale satisfactorily with a Shiny App
- Need to be able to run a range of different analysis calculations
- Need a bespoke web application and a way to send notifications to the user
Using a Message Broker as a Task Queue
Message Broker

According to IBM:

“A message broker is software that enables applications, systems, and services to communicate with each other and exchange information.”

According to VMWare:

“Asynchronous messaging allows producers and consumers to send and receive messages independently and at different times without blocking.”
Scaling Up Using a Message Broker
Asynchronous Processing

A Message Broker:
• Typically provides a Message Queue which can act as a task queue in our scenario
• Can allow our processes to communicate even when one process temporarily goes down
• We can scale up the number of “workers” running our calculations without affecting the web application
• We can have as many message queues as is required; one per different analysis code (R Script) that needs to be run
Proposed Architecture

- BROWSER
  - Request
- EMAIL
- App Database
- “View” (Django)
  - Upload Files
  - Javascript code or components
- Task Producer
- Email Process
- MESSAGE BROKER
  - Calls R Script
  - Task Worker
  - R Script: Geospatial
  - R Script: Graph
  - R Script: CSV
Main Components

- Django web application for user interaction
- Database (for user details, sessions, historical results etc)
- Message broker (REDIS or similar) with message queues configured for each different analysis script
- The Web Application interfaces with the message broker via the Python Celery library
- Cloud storage for files that are inputs/outputs of the analysis
Current Prototype

[Diagram showing the workflow of a current prototype with a browser initiating a request to a Django/Bootstrap Web App, uploading files to an app database, task producer, emailer (Django), and task worker. Files include baseline.csv, schools.csv, mda.csv, counties.gpkg, and output.html.]
Where We Are...

- All processes running within a single virtual machine (and a single Docker instance)
- Different containers for different components
- As a “cheat” we currently store data files inside a shared Docker volume (that all processes can access via a mounted file system)
Open Source?

• To open source the application we would need a truly flexible interface between web application and analysis script
• Specification of the list and types of parameters plus any web-tier validation would be in configuration not code
• By exploring the universe of likely processes we can get a good sense of how to specify this configuration