Scheduling of Distributed Information Processing

Alex Bird, Supervised by Prudence Wong, Jelena Milisavljevic Syed, Alasdair Hunter

EPSRC Centre for Doctoral Training in Distributed Algorithms, University of Liverpool, Liverpool, UK

Abstract

Problem: We need to be able to reason about what we might calculate elsewhere within a network both in present and in the future, without actually performing the calculation itself.

We consider a network of sensors connected to a central server. A scheduler needs to decide how to distribute the processing of sensed data to maintain various performance metric.

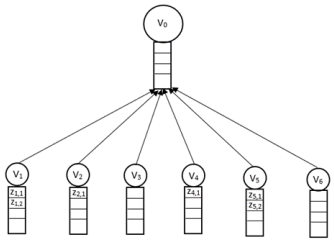


Figure 1, Network Model

Model

In order to investigate my research problem I am constructing a model and corresponding simulation to test different optimisation strategies within different network settings.

The current model consists of a set of sensor nodes and a central server. The sensor nodes randomly receive input which they must then choose to either compute locally or send to the central server for processing. The central server can compute the tasks with a higher degree of accuracy however this must be balanced with the speed and efficiency at which the tasks are completed. The overall goal of the model is to maximise the accuracy of the computed information whilst minimising the time taken to compute tasks.

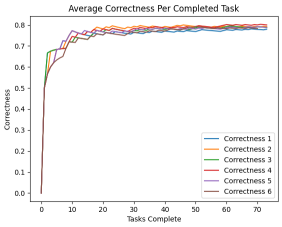
Figure one demonstrates the layout of the simulated network. Here we can see each individual node (V_1...V_6) with the incoming tasks and their connection to the central node (V_0)

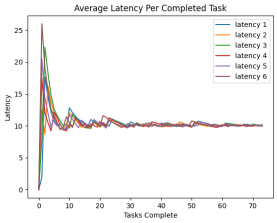
Simulation

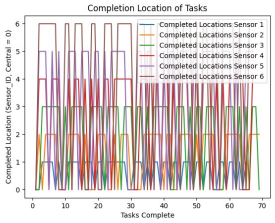
In conjunction with this model we have created a Python based simulation to test out different optimisation techniques / parameters.
The simulation utilises four different decision functions, based on two optimisation variables, to decided whether a task should be computed locally at a sensor or at the central server.
We chose the Python programming language to create the simulation as it offers comprehensive libraries for a simple simulation implementation as well as extensive libraries for the future extensions such as statistical emulation and reinforcement learning

Preliminary Results

The current simulation outputs data based upon the current performance metrics – Latency, Correctness and over-all objective function that's a ratio between Latency and Correctness. The output and performance differs depending on which decision algorithm we implement. So far the network performs best when we consider the Latency in the decision function as this results in the most tasks being complete in a reasonable amount of time.







Future Work

Expanding the Model
There are multiple different ways in which we plan to extend the functionality of the current model. These include but are not limited to: Priority tasks, bandwidth and energy consumption for transmission and a more sophisticated implementation for energy consumption.

Statistical Emulation
The main solution I have proposed for my given research problem is statistical emulation. Statistical emulators are used to approximate the output of complex physical models (3). In scheduling, statistical emulation can be used to quantify uncertainty, calibrate model parameters and improve computational efficiency (1).

This improvement in efficiency makes statistical emulation very effective in time sensitive environments where the time taken to completely process all input data may result in to slow of a response by the system (2).

Conclusion

Based on my current research, I believe statistical emulation will offer an interesting and efficient solution to my given research problem. The simulation demonstrates how a network can be optimised for different factors by making intelligent decisions on where to compute incoming tasks. In the future I aim to extend the capabilities of the network to include heterogenous sensors and a wider array of optimisation factors.

References

1. P. Tagade and H. Choi, "An emulator-based rapid source localization approach in informative sensor planning," *2012 American Control Conference (ACC)*, 2012, pp. 691-696
2. Pfeiffer, András & Kádár, Botond & Monostori, Laszlo. (2003). Evaluating and improving production control systems by using emulation.
3. M F Kasim *et al* "Building high accuracy emulators for scientific simulations with deep neural architecture search" 2022 *Mach. Learn. Sci. Technol.* 3 015013

Engineering and Physical Sciences Research Council

UNIVERSITY OF LIVERPOOL

dstl The Science Inside