Physics-Inspired Placement of Analytics Services on Heterogeneous Resources for Multisensor Fusion

Brent Kraczek
US Army Research Laboratory
ATTN RDRL CIH-C
brent.e.kraczek.civ@mail.mil

Prithwish Basu
Raytheon BBN Technologies
prithwish.basu@raytheon.com

Theodoros Salonidis
IBM T. J. Watson Research Center
tsaloni@us.ibm.com

Ali Sydney
Raytheon BBN Technologies
asydney@raytheon.com

Tom La Porta
Penn State University
tlp@cse.psu.edu

Bongjun Ko
IBM T. J. Watson Research Center
Bongjun_ko@us.ibm.com

Kevin Chan
US Army Research Lab
kevin.s.chan.civ@mail.mil

ABSTRACT

In future tactical environments, situational awareness will require fusion of data from sensors distributed across the battle domain, while computing resource access and network bandwidth are likely to be limited. We consider this problem as the ultimate limit of edge computing for sensor fusion analytics, in which AI calculations used for data fusion will also need to be distributed, employing every usable device on the network. In abstract, this maps to an optimal graph embedding problem (OGEP), with a logical graph embedded within a physical graph. The nodes and edges of the logical graph consist, respectively, of the compute components of distributed AI calculations and the associated communications needed for data collection and transfer. The nodes of the physical graph consist of sensors, available computing resources, and other data source, while communications links act as the physical graph edges. In this scenario, the distribution of tasks performed within a sensor fusion engine could span regions of the physical graph, with (possibly momentarily) fixed sensor and data resources and "agile" compute tasks and communications. In our model, we treat the interactions between nodes in each graph as particles, interacting like atoms in a molecular simulation. Because the available devices will likely *have heterogeneous capabilities, we balance logical node placement through use of Coulomb-like forces, with charges determined by resource needs on the logical graph and available resources on the physical graph. We introduce the balanced utilization index (BUI), an adaptation of Jains Fairness Index, to measure this balance. Spring-like forces, based on the physical properties of wireless networking protocols, capture the energy cost of communications. Based on these properties we define an objective function that, when minimized, simultaneously minimizes the communications cost while maximizing the BUI. Although the OGEP is generally intractable, with a highly non-convex solution space, we have developed a simulated annealing method that rapidly and reliably obtains good solutions from the set of local optima of this objective function. We demonstrate this through simulation of an abstract model of a sensor fusion engine, combining data from collocated and distributed sensors, incorporating other analysis data.*

1.0 INTRODUCTION

The battlefield of the future will be composed of millions of networked devices,[1] ranging from fixed, independent sensors to soldier-carried radios to components of large vehicles to high-performance computers. Fusion and analysis of both small-sample and large-volume data will be essential to understanding the flow of the battle from local situational awareness to global strategy. In the commercial setting, such analysis is typically performed via workstation- or data-centre-class computing hardware, either on-platform, such as in self-driving cars, or in large data centres. An alternative model is edge (or "fog") computing, which proposes to take advantage of the general growth of available computing power on devices and/or on the network.[2] While there are a variety of models for edge computing, we consider a model in which all processing capacity on a local network can be tapped to create a distributed large-scale computer. In the tactical environment, this model of edge computing would facilitate flexibility and robustness, which may be necessary, given the uncertainty of device operability and communications reliability, and general latency of communications of data over distances.

As simple example, we consider an array of cameras, microphones and similar sensors used in a security or surveillance application. The devices would built on a cheap, cell-phone-like platform, including a radio (e.g. Wi-Fi), a processor, small amounts of RAM and persistent storage, a battery and a solar cell. In data fusion applications the devices would include different sensor types and support hardware, making the computing system heterogeneous. In addition, for any given event of interest, it is likely that the system load will be non-uniform, with one to a few sensors being primarily engaged. Varying sunlight exposure, battery capacity and system power requirements would cause the available energy for processing and communications to be non-uniform as well. The robustness of such a system will depend on the distributed algorithms that coordinate the analytics: any given node of the system might fail, run out of energy, or be subject to a physical or malicious attackers, but the system should still be able to perform the needed analytics. Furthermore, it would be helpful to balance usage so that each device can remain available for sensing, reducing congestion of network and/or compute resources, etc., for as long as possible.

In this paper we address the problem of the load-balancing of multi-sensor fusion analytics as a placement problem of the component compute tasks on such a distributed wireless network. The hardware networking and compute nodes can be treated abstractly as a *physical graph*, with hardware represented by graph nodes and communications connections by graph edges. Similarly, we represent the analytics calculation by a directed, acyclic *logical graph*, in which separate data input and analysis stages are represented by nodes, while data transfers are represented by edges. Thus, the problem of distributing the data fusion calculation on the network of devices is rendered as an optimal graph embedding problem (OGEP), which is typically intractable.[3,4] Our approach uses a particle-based formulation, inspired by molecular modelling, which has been well-studied and is amenable to a distributed solution. This problem solution is based using two key components. 1. Interactions between resource types, physical and logical, are inspired by physics-like interactions, such as spring forces for communications and Coulomb-like forces to balance the distribution of code and data "particles". 2. An objective function is defined and optimized using simulated annealing (SA),[5] based on the Metropolis Monte Carlo (MMC)

method commonly used for atomic-scale simulation,[6,7] denoted below as MMCSA.

The use of physics models and physics-inspired models for network optimization has been addressed by a number of groups.[8] For example, Adamic *et al.*[9] have demonstrated that power-law graphs can enhance search algorithms for networks that include a few nodes with a high degree of connectivity. More recently, Yeung *et al.* have solved the network mapping problem for sparse graphs using statistical physics approaches.[10] However, none of these approaches use such physics-inspired formulations for distributed analytics placement. In recent work,[11] we considered a simplified version of the present problem and evaluated the efficacy of three types of physical forces to model the movement of agile code and data objects, namely gravity, elastic and a Coulomb-like interaction. While each has its pros and cons in modelling our problem, in our previous work we only applied them individually and not together. In a follow-up paper,[12] we expanded this formulation, applying both the elastic and the Coulomb-like force models simultaneously, as done in molecular modelling, and demonstrated its use in a simple sensing problem. In the present paper, we apply this to the multisensor fusion problem, with the MMCSA method used to find optimal embeddings of the logical graph.

The balance of the paper is structured as follows. In section 2 we briefly discuss how the MMCSA method given here balances resource utilization and how we determine trade-offs between minimization of energy usage and load balancing of the devices. In the third section we give show and application of this method to a data fusion problem. We conclude with a discussion of future directions for this research.

2.0 METHODOLOGY

Here we give a brief overview of the method employed. The notation is similar to that found in ref. [12], to which we refer the reader for further details. We define the logical (analytics) graph G = (VA,EA,q,r,ωAE), composed of nodes/vertices VA and edges EA, with weights q : VA →R+, r : VA →R+, ωAE : VA×VA →R≥0, set respectively as the constant processing energy cost of each analytics stage, RAM requirement of each analytics stage and the directed data transmission (in megabytes) between each pair analytics stages (the majority of which will be zero). Similarly, we define the physical (resource) graph H = (VP,EP,Q,R,ωPE), composed of composed of nodes/vertices VP and edges EP, with weights R : VP →R−, Q : VP →R−, ωPE : VP ×VP →R+, set respectively as the processing energy capacities and RAM of each physical node and the data transmission cost (in Joules per megabyte) between physical nodes. The placements of individual logical nodes on the physical graph are given by π : VA → VP, and are updated throughout the calculation.

We employ an objective function consisting of three primary terms,

Φ = Φcomms + cproc Φproc + cRAM ΦRAM , (1)

where Φcomms is the energy associated with internodal communications, Φproc is a term that both constrains and balances the processing energy cost for each stage of the analytics, and ΦRAM is a penalty term associated with ensuring that the available memory on a given node is not exceeded by the assigned stages of the analysis. The constants cproc and cRAM are tuning parameters used to alter the relative importance of the three primary terms, combining them into a numerical objective for optimization. In our previous work,12 we found that there is a limited range over which this tuning is necessary, especially for cRAM, which effectively acts as a binary check to ensure that the memory constraint is satisfied.

We employ MMCSA to obtain the set of placements on the nodes of the physical graph that minimize the objective function Φ. Following standard MMCSA procedure (algorithm 1), we employ a temperature τ, which follows a cooling schedule. Because the system is not physical motivated, τ and the cooling schedule are selected based on the scenario under consideration. In the current implementation we employ an exponential function for τ that decays from τinit at s = 0 (initialization) to τmin at the MC step s = smax. At each MC step s, a logical node is chosen at random and subjected to a trial move on the physical graph. With each trial move, if the change ΔΦ is negative, the trial is accepted automatically. If ΔΦ is positive it is accepted if $\exp(-\Delta\Phi/\tau) >$ rand(0,1).

The evolution of the system during the MMCSA method is the search for the optimal solution and not the actual evolution of the system, so it is only important that the soft constraints, handled by the penalty terms Φproc and ΦRAM, are not violated in the final solution. It is often beneficial during the MMCSA calculation that both the RAM and processing energy capacities are exceeded early in the calculation, when values of τ are high, to discourage trapping of the solution in a less-optimal, local minimum. Naturally, the choice of cooling schedule τ(s) determines the quality of the final configurations obtained by the MMCSA calculations, while sensible initial placements of the logical graph nodes on the physical graph can decrease the number of iterations required.

The present implementation optimizes processing load through the use of Coulomb-like interactions in Φproc that simultaneously act as penalty functions to ensure the analytics placement does not exceed (violate) the processing energy capacity of any given physical node. While the standard fairness measure, Jain's Fairness Index (JFI),13 is an appropriate measure for homogeneous resources, it does not address the case of heterogeneous resources. To measure this, we employ an adaptation of JFI, which we call the balanced utilization index (BUI), introduced in.12 In short, like JFI, the minimum and maximum values of the BUI lie between 0 and 1, with the specific, possible minimum and maximum values depending on the specific problem. A higher value denotes a more favourable balance.

Algorim 1: MMCSA
MMC method for optimizing "logical" graph placement on a "physical" graph. The algorithm is MMC for a fixed τ and MMCSA for τ that varies with s.

Set physical graph within a 2D Euclidean space.
Embed logical graph(s) in physical graph, with intermediate nodes distributed evenly between endpoints.
Calculate Φ for initial configuration.
for $s = 1 : s_{max}$ **do**
    Randomly choose loical graph node u.
    Displace node u from physical graph node z to trial node z^0.
    Compute $\Delta\Phi \leftarrow \Phi(\pi(u) = z^0) - \Phi(\pi(u) = z)$.
    if $\Delta\Phi < 0$ **then**
    Set $a \leftarrow$ True.
      else if $e^{-\Delta\Phi/\tau(s)} >$ rand(0,1) **then**
        Set $a \leftarrow$ True
    else
        Set $a \leftarrow$ False
    end if

if a **then**
$\Phi \leftarrow \Phi + \Delta\Phi$
$\pi(u) \leftarrow z^0$
end if
Record data
end for
return Placement of logical graph nodes $\{\pi(u)\}$.

3.0 APPLICATION TO THE MULTISENSOR FUSION PROBLEM

Here we demonstrate the operation of the method on an information fusion problem employing a combination of interpretations data fusion approach (i.e. the individual inputs are classified first, followed by analysis of information). For the logical graph, we simulate the distributed use of five instances of the AlexNet image processing algorithm,14 with the 10 layers of the algorithm distributed on separate nodes of the logical graph. The results of these analyses are then fused via an abstract, 4-stage algorithm. The initial and final points of the logical graph are fixed on the physical graph, representing a observing sensors and a designated node for transmission from the network to an outside observer; the intermediate nodes of the logical graph are moved by the MMCSA method. The energy costs are based on 100 evaluations of AlexNet on a Samsung Galaxy S5 smartphone, though with the RAM limited to 256MB per node. For communications, we define the Euclidean positions of the physical nodes and determine $\omega_P^E : V_P$ based on the Euclidean distances between the nodes with the energy costs derived from the minimum multi-hop energy costs under the 802.11ac 20MHz DS 2x2 MIMO wireless spec, as measured by Saha et al.15 Numerical experiments demonstrate that with this configuration, the parameters employed within the MMCSA method can be readily determined and are robust across different physical configurations. In order to demonstrate the efficacy of the MMCSA method in balancing resource utilization, the processing energy capacities of physical nodes were chosen to be non-uniform and represent a system with sufficient resources to process on the order of seven sets of images, representing a system that is near its loading capacity. To define the physical graph, we use a geometric random graph with 80 nodes, distributed in four square kilometres.

In figure 1 we show schematics of the processing energy and RAM for the initial (s = 0) and final (s = 20000) configurations, as well as one intermediate configuration (s = 8000). The initial condition was chosen to lie on a short path between the initial and final positions, with the analytics positioned along this path. The areas of blue left-semicircles are proportional to the available capacity for RAM or processing energy for the physical node located at the semicircle's origin. The areas of the opposing, right-semicircles represent the size-proportional utilization. These are coloured red at nodes that violate the capacities and green at those that are within capacity. The grey lines connecting the physical nodes show the available Wi-Fi connections, based on the distances employed in ω_P^E. The heavy coloured lines show which connections are used for communications by the distributed analytics calculation, with different colors for distinct analytics calculations. A circle represents the point where the sensor makes an observation, and thus starts a calculation, while process communications are represented by the heavy lines of the same color. The black triangle is the point where the individual images are completed and passed to the fusion engine, while the black square is the point where the final analysis is delivered. The initial placement of logical nodes distributes the nodes as uniformly as possible along the lowest-cost communications pathway between the observation point and the final point. As can be seen in the schematics in figure 1, the initial placements contain violations of both the processing energy and RAM capacities. After running the MMCSA algorithm the placements satisfy the capacities on the system. In figure 2 we show the evolution of Ecomms and the BUI of the processing energy as a function of the MMCSA iteration number. Thus both the communications energy and the BUI have relatively small initial values. The cooling schedule is a simple, decaying exponential. Thus, the both quantities quickly increase and fluctuate wildly in the initial "high-temperature" portion of the annealing. As the temperature decreases, the primary contributions to the objective function become fixed, followed by the lesser contributions.

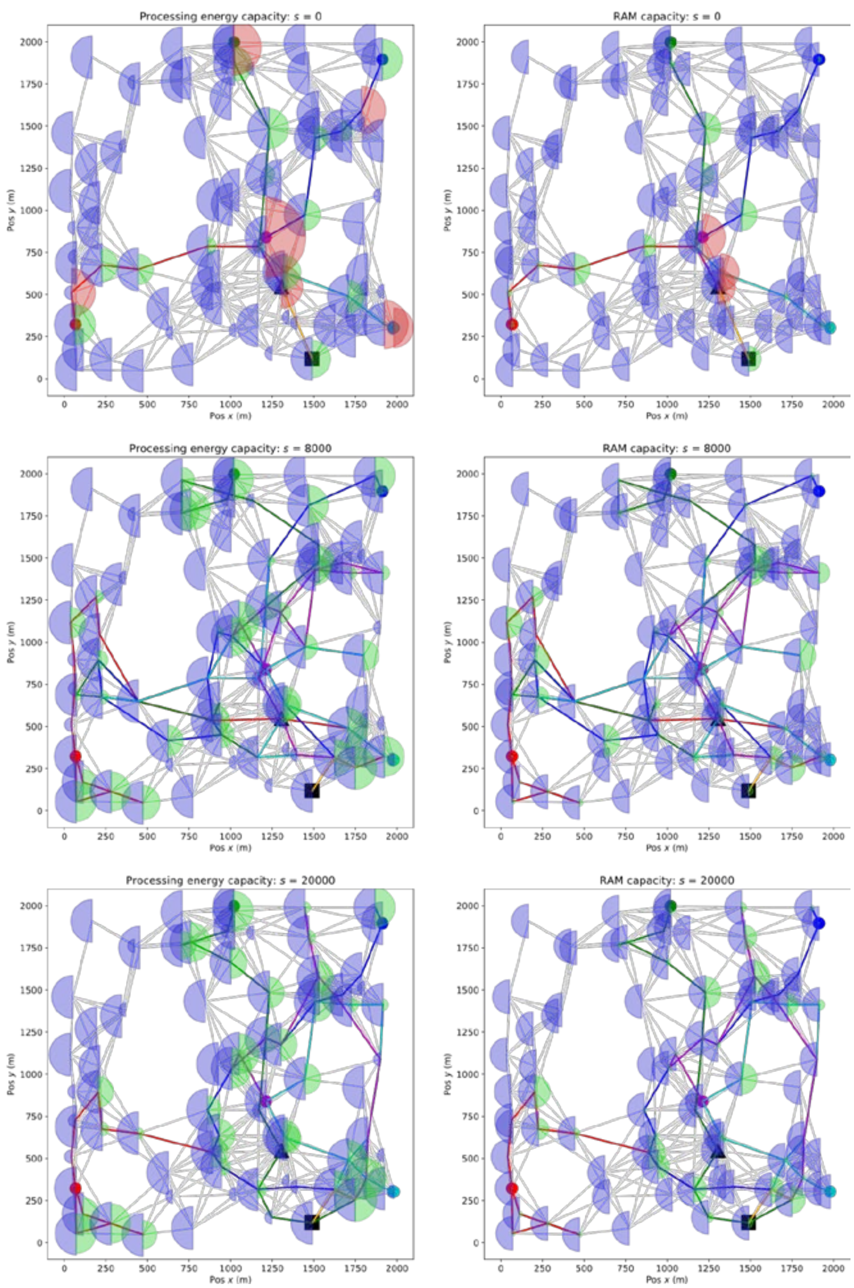

Figure 1: Evolution of placement solution for fusion of data from 5 sensors Schematics showing the available resources on the physical graph and utilization by analytics on the logical graph for experiment 1. Blue left-semicircles represent the capacity at each physical node. Red right-semicircles represent a utilization that is over capacity, while green right-semicircles represent a utilization that is within capacity. The areas of the semicircles are proportional to the

capacity/utilization, illustrating the actual utilization. The value of *s* denotes the iteration number.

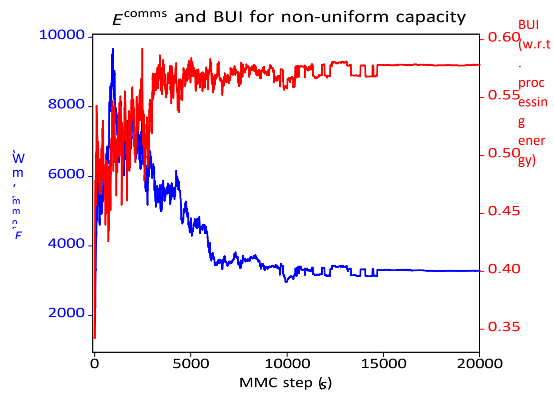

Figure 2: Evolution of metrics with MMCSA step.
Plot of communications energy (blue) and BUI of the processing energy (red), as a function of iteration number.

4.0 CONCLUSION AND FUTURE WORK

In this paper we have demonstrated a physics-inspired MMCSA algorithm to optimize on a random network of compute nodes the distribution of analysis components of a data fusion engine. This is done by formulating the problem as an OGEP problem and designing the objective function to employ penalty functions to enforce system capacities and balance system utilization. We employed the balanced utilization index (BUI), an extension of Jain's Fairness Index, as a metric of how well different placements balance the remaining energy for processing, after the analytics calculations are completed.

In the future, we anticipate that the performance of the method, both in terms of decreasing the number of iterations required and the optimality of solutions, could be improved by choosing more complicated cooling schedules and/or by using an energy window for MMC acceptance steps, rather than a threshold. For example, at large τ, many of the early placement fluctuations are of logical stages that would still move at lower values of τ, potentially leading to more-expensive communications, while not significantly improving the final BUI solution. Finally, we anticipate that with a modified objective function, this method could be applied to the optimization other network topologies, including those of heterogeneous computing platforms employed for largescale machine-learning calculations, such as those composed of many CPUs and GPUs.

ACKNOWLEDGMENTS

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

[1] Kott, A., Swami, A., and West, B. J., "The internet of battle things," *Computer* **49**, 70–75 (2016).

[2] "Fog computing and the internet of things: Extend the cloud to where the things are," White Paper, Cisco Systems, Inc., San Jose, CA (2015).

[3] Bokhari, S. H., "On the mapping problem," *IEEE Trans. Comput.* **30**, 207–214 (Mar. 1981).

[4] Monien, B. and Sudborough, H., *Embedding one Interconnection Network in Another*, 257–282, Springer Vienna, Vienna (1990).

[5] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by simulated annealing," *Science* **220**(4598), 671–680 (1983).

[6] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equations of state calculations by fast computing machine," *J. Chem. Phys.* **21**, 1087–1091 (1953).

[7] Frenkel, D. and Smit, B., *Understanding Molecular Simulation*, Academic Press, Inc., Orlando, FL, 2nd ed. (2002).

[8] Saad, D., Yeung, C. H., Rodolakis, C. H., Syrivelis, G., Koutsopoulos, I., Tassiulas, L., Urbanke, R., Giaccone, P., and Leonardi, E., "Search in power-law networks," *IEEE Communications Magazine* , 144– 151 (November 2014).

[9] Adamic, L. A., Lukose, R. M., Puniyani, A. R., and Huberman, B. A., "Search in power-law networks," *Phys. Rev. E.* **64**, 046135 (2001).

[10] Yeung, C. H. and Saad, D., "Competition for shortest paths on sparse graphs," *Phys. Rev. Lett.* **108**, 208701 (2012).

[11] Ko, B., Kraczek, B., Salonidis, T., Basu, P., Chan, K., Porta, T. L., Martens, A., and Lambert, J., "Proc. 1st workshop on distributed analytics infrastructure and algorithms for multi-organization federations (DAIS)," in *1985 24th IEEE Conference on Decision and Control*, (2017).

[12] Kraczek, B., Salonidis, T., Basu, P., Saghaian, S., Sydney, A., Ko, B., Porta, T. L., Chan, K., and Lambert, J., "Balancing distributed analytics' energy consumption using physics-inspired models," in *SPIE Proceedings: Disruptive Technologies in Information Sciences*, **10652** (2018).

[13] Jain, R., Chiu, D. M., and Hawe, W., "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Tech. Rep. DEC-TR-301, Digital Equipment Corporation (1984).

[14] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., eds., 1097–1105, Curran Associates, Inc. (2012).

[15] Saha, S. K., Deshpande, P., Inamdar, P. P., Sheshadri, R. K., and Koustonikolas, D., "Power-throughput tradeoffs of 802.11n/ac in smartphones," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 100–108 (2015).