Demonstration of Dynamic Distributed Orchestration of Node-RED IoT Workflows Using a Vector Symbolic Architecture

Richard Tomsett†,Graham Bent†,
Christopher Simpkin*, Ian Taylor*, Daniel Harborne*,Alun Preece*,
Raghu K. Ganti‡
†IBM Research UK
{rtomsett, gbent}@uk.ibm.com
*School of Computer Science and Informatics, Cardiff University
{simpkinc, taylorij1, HarborneD, PreeceAD }@cardiff.ac.uk
‡IBM Research USA
{rganti}@us.ibm.com

***Abstract*—Traditional service-based applications, in fixed networks, are typically constructed and managed centrally and assume stable service endpoints and adequate network connectivity. Constructing and maintaining such applications in dynamic heterogeneous wireless networked environments, where limited bandwidth and transient connectivity are commonplace, presents significant challenges and makes centralized application construction and management impossible. In this demonstration we present an architecture which is capable of providing an adaptable and resilient method for on-demand decentralized construction and management of complex time-critical applications in such environments. The approach uses a Vector Symbolic Architecture (VSA) to compactly represent an application as a single semantic vector that encodes the service interfaces, workflow, and the time-critical constraints required. By extending existing services interfaces, with a simple cognitive layer that can interpret and exchange the vectors, we show how the required services can be dynamically discovered and interconnected in a completely decentralized manner. There are a large number of workflow systems designed to work in various scientific domains, including support for the Internet of Things (IoT). One such workflow system is Node-RED, which is designed to bring workflow-based programming to IoT. The main focus of this demonstration is to show how we can migrate Node-RED workflows into a decentralized execution environment, so that such workflows can run on Edge networks.**

***Index Terms*—Decentralized Workflows, Vector Symbolic Architecture, Machine Learning, Dynamic Wireless Networks**

I. Introduction

During the last decade, there has been an explosion in the quantity, variety and complexity of data generated routinely by research and industry. This has been driven by the development of new virtualization technologies (e.g. containers, virtual machines, and so forth) which allow applications to adapt elastically on-demand, as well as the adoption of service interfaces (e.g. micro-services) which break down complex problems into smaller and repeatable tasks. In addition the emergence of smart devices and sensors, many of which are located at the edge of wireless networks, collectively known as The Internet of Things (IoT) represents a rapidly burgeoning requirement for distributed communications and data analytics in a distributed environment. Whilst IoT devices are widely distributed geographically, the current approach for management of such devices is cloud based and therefore similarly centralised. However, considering the huge and growing volume represented by the IoT it will become more and more expensive and impractical to manage and coordinate billions of devices in centralised server farms.

Vector Symbolic Architectures (VSAs) [1, 2, 3, 4] are a set of lossy dimensionality reduction methodologies that enable large volumes of data to be compressed into a fixed size vector in a way that captures associations and similarities as well as enabling categorizations between data to be built up. Such vector representations are recursive as originally proposed by Hinton [5] in that they allow for higher level abstractions to be formulated in the same format as their lower level components. VSAs are capable of supporting a large range of cognitive tasks such as; *(a)* Semantic composition and matching; *(b)* Representing meaning and order; *(c)* Analogical mapping; *(d)* Logical reasoning; *(e)* They are highly resilient to noise; *(f)* They have neurologically plausible analogues which may be exploited in future distributed cognitive architectures. Consequentially they have been used in natural language processing [6, 7, 8] and cognitive modeling [9, 10].

In this demonstration we show how a VSA can be used for distributed semantic discovery and orchestration of remote devices into different workflow configurations without any knowledge of the 'IP' location of such devices and without the need for a central point of control. To do this we created a new operational mode for Node-RED [11], a well known cloud based graphical workflow management system for IoT devices. We show how this new mode can be used to constructs the same workflow in a mobile ad-hoc network (MANET) environment modelled in a CORE/EMANE [12], real-time network emulator. Further, we describe how the VSA can be used to orchestrate more complex data analytics tasks and, as a test case, use the new Node-Red mode to run a distributed simulation of the Montage Pegasus[13] workflow, in the same emulated MANET environment.

II. Architecture of the Demonstration

Figure 1 presents a high level overview of the demonstration architecture. The required workflows are composed using the Node-RED graphical interface. The interface allows a user to specify the the required services and the connectivity of each workflow step. Our scheme converts the Node-RED micro-service workflow into a symbolic vector representation. Each vector can be transmitted in a single wireless packet(10 kbit), which can be multi-cast from any wireless node in the emulated MANET environment. The micro-services are represented as a set of decentralized proxies, which call the corresponding web service from within the emulated MANET environment. Each proxy service is equipped with a cognitively-aware wrapper which facilitates the decentralized discovery, construction and execution of the workflow by exchanging symbolic vectors over the MANET network. Services are discovered and linked together using only their semantic vector description and therefore IP addresses are not required, as per the normal Node-RED scheme.

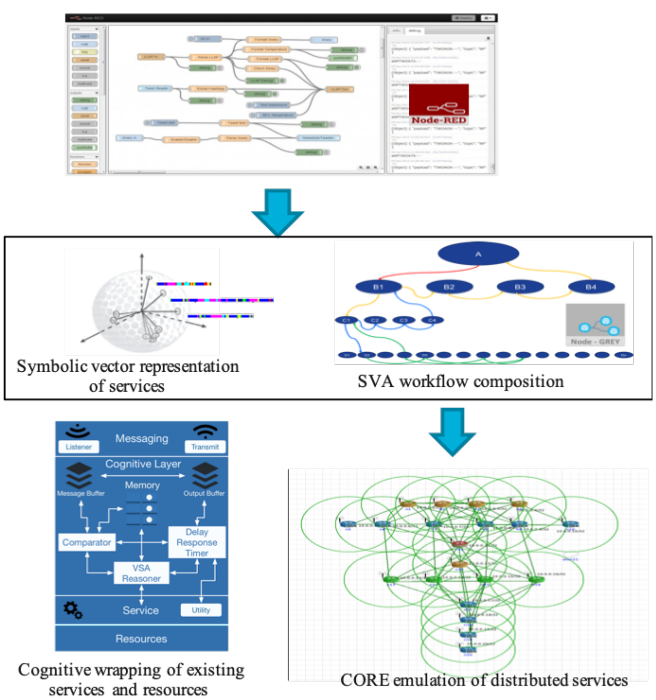

Fig. 1: Architecture of the Demonstration

The VSA scheme scheme is used to represent workflows by combining vectors that represent services, edges, sub-workflows, branches, etc. In the case of smaller workflows (10-20 services) a single vector is sufficient. In the case of larger workflows the vectors are further combined into a hierarchical set of vectors that represent the workflow. These extremely compact representations are transmitted across the MANET (e.g., using multicast) and only services that semantically match a particular vector will proceed with the work. The vectors have the property that they can then be unbound to unravel the workflow and therefore, the decentralized operation is simplified to a transmit-unbind and re-transmit procedure, since each unbind unravels the next step of the workflow [14].

This simple, and yet powerful approach can provide semantically rich representations of workflows that have several interesting by-products, including: *(a)* the ability to make semantic comparisons at each level of the architecture (e.g., semantic searches are scoped within a sub-group of services in a workflow); and *(b)* and the ability to bind extra metadata along with the workflow structure, which is only accessible to the services that can consume it (e.g. it enables policy enforcement).

III. Traffic Congestion Use Case

The demonstration is based on a traffic monitoring scenario involving sensing (e.g., via a network of traffic cameras) and decision making (e.g., routing traffic to avoid congested areas) supported by an interactive question-answering interface ([15], [16]). The concept for this interface is to provide decision support for a user tasked with managing the state of city or region-wide traffic. In [15], we explored detecting traffic congestion using a number of services which could be both distributed and owned by multiple agencies (i.e., operating as a coalition). In [16], we explored how natural language queries relating to traffic could be answered by taking advantage of the output of distributed data sources and processing services. In both these pieces of work, we did not outline the co-ordination of these distributed resources, merely providing specific architectures and the Node-RED workflows that could provide the required answers.

In the demonstration we show how our VSA enabled Node-RED can be used to semantically describe and cognitively wrap the existing services and how we construct the workflow vector that is used to orchestrate the discovery and execution of the workflow across the distributed resources.

A. Data Sources & Processing Resources

The main data source we have taken advantage of is the Transport for London (TFL) traffic camera API[1]. This allows access to imagery and video from around one-thousand traffic cameras situated around London. The imagery and video is updated every five minutes and the video provided is a ten second clip recorded at the beginning of the five minute interval.

[1] http://www.trafficdelays.co.uk/london-traffic-cameras/

To detect cars, we process the imagery from the traffic camera feeds using an object detector (MobileNetSSD[2]) supplied within the OpenCV (Open Source Computer Vision) library [3]. Finally, to convert the list of detected cars to a count we use a simple service that is designed to count the items in a list it receives. Having this as a service (and not hard coded in to the interfaces processing of the result for example) allows for this list to count function to be used within workflow construction.

B. Moving to a Dynamic and Decentralized Environment

Within a decentralized environment, as illustrated in Figure 2, these resources need to be discovered dynamically amongst a distributed array of services. Once the nature of the query is established, the correct services must be identified and chained together in order to answer the query. During the discovery process there are two key considerations, services may be replicated identically providing redundancy and thus there may be multiple services that provide a perfect fit or the required functionality. These must be discovered and selected appropriately. Secondly there may be services available which, although do not meet the functionality exactly, still provide the functionality required. For example, when counting the number of vehicles on a road, a vehicle detector (a detector that identifies cars, bikes, vans etc) is a perfect fit but if detectors for these individual concepts exist (individual car detector, van detector etc), their output could be aggregated and provide an output that may still be appropriate if no vehicle detector is available.

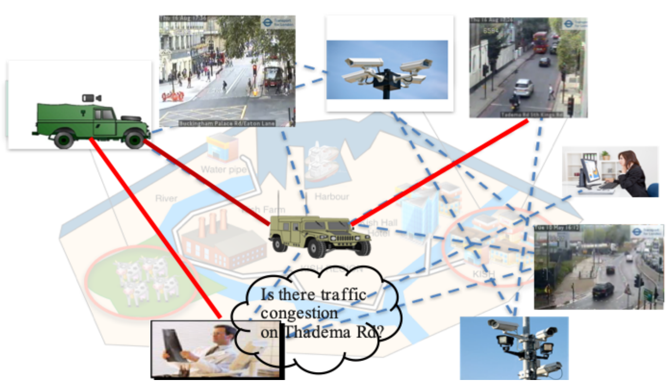

Fig. 2: Distributed architecture for answering the question ”How many cars on Thadema Road?”

A method of discovery and execution within a distributed setting must factor these two properties in order to best take advantage of the resources made available and to maximize the queries that are answerable.

IV. Conclusions And Future Work

In this demonstration we show how we can migrate Node-RED workflows into a decentralized execution environment, so that such workflows can run on Edge networks, where nodes are extremely transient in nature. We show how using vector symbolic architectures (VSA), in which symbolic vectors can be used to encode workflows containing multiple coordinated sub-workflows, that applications containing multiple distributed services can be constructed on-the-fly and executed in a completely decentralized way.

We believe that in future the VSA approach offers the potential to combine the workflow, self-describing services and data into vector representations that will enable alternative service compositions to be automatically constructed and orchestrated to perform tasks specified at higher levels of semantic description. Our future work will therefore focus on such self-describing service compositions in order to realize the vision set out in [17].

[2]MobileNet-SSD: https://github.com/chuanqi305/MobileNet-SSD
[3]https://github.com/opencv/opencv

V. Acknowledgements

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] T. A. Plate, *Distributed representations and nested compositional structure*. University of Toronto, Department of Computer Science, 1994.

[2] R. W. Gayler, “Vector symbolic architectures answer jackendoff’s challenges for cognitive neuroscience,” *arXiv preprint cs/0412059*, 2004.

[3] P. Kanerva, “Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors.” *Cognitive Computation*, vol. 1, no. 2, pp. 139–159, 2009. [Online]. Available: http://dblp.uni-trier.de/db/journals/cogcom/cogcom1.html#Kanerva09

[4] D. Kleyko, “Pattern recognition with vector symbolic architectures,” Ph.D. dissertation, Luleå tekniska universitet, 2016.

[5] G. E. Hinton, “Mapping part-whole hierarchies into connectionist networks,” *Artificial Intelligence*, vol. 46, no. 1-2, pp. 47–75, 1990.

[6] M. N. Jones and D. J. K. Mewhort, “Representing word meaning and order information in a composite holographic lexicon,” *psychological Review*, vol. 114, no. 1, pp. 1–37, 2007.

[7] G. E. Cox, G. Kachergis, G. Recchia, and M. N. Jones, “Toward a scalable holographic word-form representation,” *Behavior research methods*, vol. 43, no. 3, pp. 602–615, 2011.

[8] G. Recchia, M. Sahlgren, P. Kanerva, and M. N. Jones, “Encoding sequential information in semantic space

models: comparing holographic reduced representation and random permutation," *Computational intelligence and neuroscience*, vol. 2015, p. 58, 2015.

[9] T. A. Plate, *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. Stanford, CA, USA: CSLI Publications, 2003.

[10] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338, no. 6111, pp. 1202–1205, Nov. 2012. [Online]. Available: http://www.sciencemag.org/content/338/6111/1202

[11] "Node-RED: Flow-based programming for the Internet of Things," https://nodered.org/.

[12] J. Ahrenholz, C. Danilov, T. Henderson, and J. Kim, "Core: A real-time network emulator," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, Nov 2008, pp. 1–7.

[13] "Workflow Generator Pegasus," https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator.

[14] C. Simpkin, I. Taylor, G. A. Bent, G. de Mel, and R. K. Ganti, "A scalable vector symbolic architecture approach for decentralized workflows."

[15] D. Harborne, C. Willis, R. Tomsett, and A. Preece, "Integrating learning and reasoning services for explainable information fusion," *International Conference on Pattern Recognition and Artificial Intelligence*, 2018.

[16] D. Harborne, D. Braines, A. Preece, and R. Rzepka, "Conversational control interface to facilitate situational understanding in a city surveillance setting," *The fourth Linguistic and Cognitive Approaches to Dialog Agents Workshop (LACATODA 2018)*, 2018.

[17] D. Verma, G. Bent, and I. Taylor, "Towards a distributed federated brain architecture using cognitive iot devices," in *9th International Conference on Advanced Cognitive Technologies and Applications (COGNITIVE 17)*, 2017.