Data-aware Declarative Application Management in the Cloud-IoT Continuum

by Jacopo Massa (University of Pisa / CNR-ISTI), Stefano Forti, Patrizio Dazzi and Antonio Brogi (University of Pisa)

Nowadays billions of devices are connected to the Internet of Things and can reach computing facilities along the Cloud-IoT continuum to process the data they produce, leading to a dramatic increase in the number of deployed applications as well as in the amount of data they need to crunch. Following a continuous reasoning approach to speed up the decision-making process, our research proposes a declarative and data-aware solution to determine service-based application placements over the Cloud-IoT continuum while meeting functional and non-functional application requirements.

In recent years, the Internet of Things (IoT) has become increasingly integrated with our everyday lives, driving the definition of new computing models and paradigms. According to the latest reports, there will be more than 30 billion devices connected to the Internet by the end of 2023, leading to a massive amount of data being processed and acted upon, generated by several sources at the edge of the network, majorly IoT devices. There are several reasons why the cloud paradigm alone cannot always meet the constraints among end devices and cloud servers, not to mention data processing and transmission costs along physical links. Secondly, many applications require real-time interactions, i.e. quasi-real-time data exchange, which often cannot be offered due to the significant end-to-end delay (i.e. latency) between nodes. Last but not least, security is becoming more and more important since many applications work with sensitive data that should not traverse the Internet, e.g. for privacy reasons.

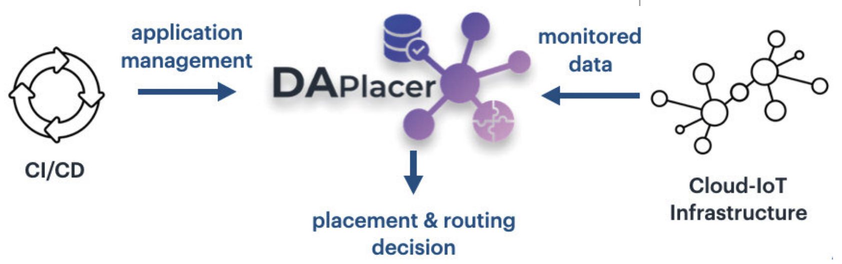

Figure 1: DA-Placer workflow.

To address these challenges, several Cloud-IoT paradigms – e.g. fog, edge, mist, and osmotic computing – have been proposed. They exploit heterogeneous computing capabilities along the Cloud-IoT continuum (e.g. smartphones, access points, gateways, datacentres) to process data close to their IoT sources. In between the cloud and the IoT, computational nodes act both as processing capabilities closer to the devices and as filters over data streams directed towards the cloud.

The previously mentioned paradigms require application services to be adequately placed along the available Cloud-IoT resources to meet all functional and non-functional application requirements. Hence, deciding where to deploy (and possibly later migrate) application services along the Cloud-IoT continuum has been largely studied in the literature. However, despite taming the data deluge and achieving data-awareness being among the main motivations of Cloud-IoT computing, to the best of our knowledge, the characteristics of the data (e.g. security needs, volume, velocity) processed by the application have been used only marginally to drive placement decisions.

To this end, we devised a Prolog open-source tool [L1], named DA-Placer [1], which inputs a declarative description of a multi-service application and its requirements, within a declarative description of a multi-layered infrastructure with dual capabilities. It exploits a declarative strategy that maps each application service to a compatible node, without exceeding nodes and links capabilities, and giving as output the set of placement and routing decisions (see Figure 1). Our declarative strategy takes into account requirements in terms of software, hardware and IoT, but also a taxonomy of security constraints and QoS (quality of service) requirements (latency and bandwidth). Since we are in a data-aware context, we also deal with data characteristics, such as their size, transmission rate and their sources and targets. Prolog has been chosen for the simplicity of its syntax and for managing and updating the code. Above all, the backtracking technique exploited by the Prolog reasoner to find a (possibly existing) solution, ensures an exhaustive search in the solution space.

To tame the EXPTIME complexity of the considered problem for prompter decision-making at runtime, DA-Placer exploits a continuous reasoning approach [2]. Once a deployment has been enacted according to a found placement and routing, continuous reasoning tries to reduce the size of the considered placement problem instances at runtime, by focusing on re-placing those services and data routings that cannot currently meet their requirements. This can mainly happen for two reasons:

- Due to changes in the monitored Cloud-IoT infrastructure (e.g. node crash or overloading, link QoS degradation) that prevent meeting application requirements
- Due to changes in the declared application (e.g. service removal/addition, requirements update, changes in the data types handled) that require (un)deploying services or migrating existing ones.

When possible, after identifying the deployment portion affected by the changes above, continuous reasoning attempts to determine a new placement and data routing only for such a portion.

The research that led to the prototyping of DA-Placer is only in its infancy and will continue to expand the knowledge barrier in several directions, including further management decisions (e.g. backup/replicas), multi-objective optimisation (e.g.

energy consumption, operational costs), or exploitation of local capabilities to guarantee resilience to churns in a distributed/decentralised context, as well as theoretical results to prove the correctness and completeness of the devised declarative approaches.

This work has been partly supported by the EU H2020 ACCORDION project, and by the project "Energy-aware management of software applications in Cloud-IoT ecosystems", funded with ESF REACT-EU resources by the "Italian Ministry of University and Research" through the "PON Ricerca e Innovazione 2014—20".

Links:
[L1] https://github.com/di-unipi-socc/daplacer

References:
[1] J.Massa et al., "Data-Aware Service Placement in The Cloud-IoT Continuum", in Service-Oriented Computing in CCIS,vol. 1603, p.139-158, 2022.
[2] Forti et al., "Declarative Continuous Reasoning in the Cloud-IoT Continuum" in J. of Logic and Computation, 2022.

Please contact:
Jacopo Massa, University of Pisa / CNR-ISTI, Italy
jacopo.massa@phd.unipi.it

IndeGx: An Index of Linked Open Datasets on the Web

by Pierre Maillot, Catherine Faron, Fabien Gandon and Franck Michel (Inria)

Where can we find the data we need for a given task? IndeGx aims to create an index of knowledge graphs published on the web, in the form of linked open datasets, for humans and machines alike. This framework provides descriptions of the knowledge graphs to draw a picture of their content, quality and compliance with the standards. A companion webpage is also provided to support data providers in the description of their datasets in compliance with the latest best practices.

In recent years, a large number of knowledge graphs have been built and published on the web in the form of RDF datasets, in fields as diverse as linguistics or life sciences, along with other general datasets such as DBpedia or Wikidata. The reliable exploitation of these datasets requires specific knowledge about their content, access points and commonalities. Yet, it is usually difficult to have such a clear and up-to-date description as most datasets have neither a machine-readable nor a human-readable description, and not all access points can handle the complex queries required to automatically generate such descriptions. The data providers are commonly regarded as responsible for describing the datasets they publish. However, this requires specific efforts, costs, and skills that some providers, who are not necessarily experts in semantic web technologies, do not have or cannot afford. In particular, these descriptions rely on a deep understanding and joint use of specialised vocabularies and there is no standard model or tool for generating and updating these descriptions.

IndeGx is a transparent, declarative, collaborative and extensible framework designed to generate the description of a knowledge graph solely based on information that can be extracted from a SPARQL endpoint that serves that knowledge graph. It is part of the ongoing effort to help humans and machines in the use of knowledge graphs on the web. It creates an open repository of descriptions to guide agents in selecting knowledge graphs, and supports a variety of use cases. For instance, they could be used to fuel the faceted search of a dataset catalogue meant for human agents, or a query federation engine could leverage the statistics on the usage of the classes and properties in a dataset to efficiently rewrite a query over multiple graphs.

IndeGx relies on generation rules expressed in standard RDF vocabularies and SPARQL to yield common metadata features such as provenance information and lists of classes and properties, as well as new kinds of metadata features not covered by previous approaches, such as quality indicators, lists of vocabularies and new statistics. Furthermore, the processing of a dataset at different points in time makes it possible to track its evolution. The descriptions generated by IndeGx are represented in RDF and published as a regular, public knowledge graph.

The rules that generate the descriptions are written with the same vocabulary as the one used by the W3C to describe test suites in RDF. When joined with the execution traces that are also kept and represented in RDF, the process of generating a dataset description is fully transparent and traceable. Moreover, the rules and the IndeGx application are available under open licences [L1], and anyone can replicate them, extend them or create their own set of rules, as long as they use the same vocabulary.

The KartoGraphI website [L2] relies on IndeGx to draw a picture of the open sources of the semantic web. At the time of writing, it provides different visualisations computed to monitor, over an 8-month period, 339 datasets with SPARQL endpoints retrieved from well-established dataset catalogues such as the Linked Data Cloud website and Wikidata. The details of

Figure 1: Geolocalisation of the endpoints described in IndeGx as shown on the KartoGraphI website.