RADICAL - OSI project

IoT environments require immediate processing of intermediate and transient data, often under resource constrained environments. Such applications consist of distributed cyberinfrastructure with heterogeneous resources such as “edge devices”, “fog” or “cloudlets”, with connections to clouds and high-performance computing systems. In particular, many sensors act as “edge devices” which increasingly can support non-trivial computing, and thus in principle time-sensitive computation can be offloaded onto these edge devices.

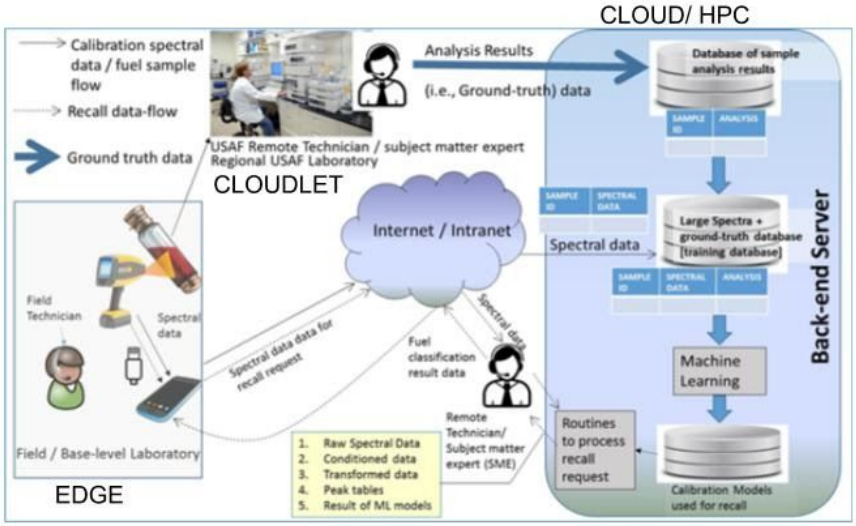

Fig 1. Use case for OSI

Although more responsive, computation and data storage on edge devices is challenging as they are more performance constrained than cloud or HPC resources, and typically do not scale dynamically. Alternatively, though cloud resources provide an increase in computational power and data storage, challenges, such as security, reliability, and network bandwidth can hinder application production. Thus there is a tension between low-latency immediate response but limited capability on edge devices, with remote but significant resources in the cloud. Thus, in order to exploit and obtain optimal performance, there is a growing need to understand and manage the distribution of workloads on heterogeneous and distributed resources.

Our work attempts to identify the fundamental components as well as the interactions involved in modeling and emulating such a distributed cyberinfrastructure to provide optimum QoS and performance via distributed workload placement. We defined a coarse-grained conceptual model followed by a finer-grained reference architecture, adding details to each fundamental component. The Conceptual Model describes the correspondence between two fundamental elements, Data Acquisition (DA), and Resources. It is developed to increase the understanding of the system to be modeled. We extend our discussion by developing a Reference Architecture, which provides a comprehensive study of the elements, their sub-elements and the directed interactions amongst them.

Together based on the conceptual model and reference architecture, we implemented a prototype of the emulator to advance our understanding of optimal load balancing in a heterogeneous and distributed environment. Currently, the emulator enables (1) creation of workloads with heterogeneous or homogeneous tasks; (2) creation of resources with homogeneous and dynamic cores; (3) creation of heterogeneous resources with varying core performance such as cloud and edge devices; (4) introduction of network latency considerations and (5) specifying criteria for scheduling decisions.

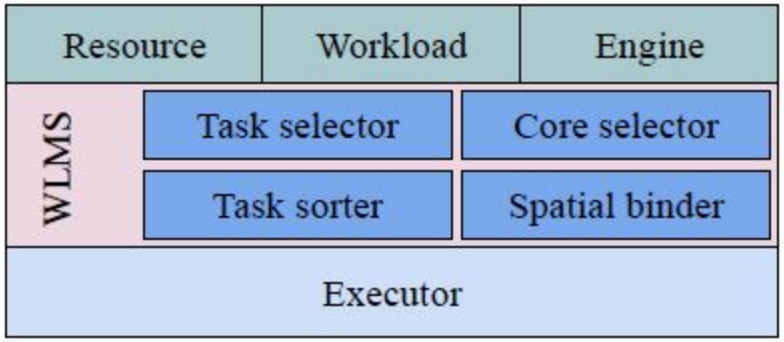

Fig 2. Architecture for the Emulator (Vivekanandan Balasubramanian. 2019 RADICAL Lab)

The emulator helps us understand suitable configurations required to compute a workload in a distributed cyberinfrastructure of edge devices, clouds and high-performance computing systems. This leads to determining the optimal distribution of training and inference over HPC/Clouds versus edge devices as well as determining efficiency and time-to-solution of ML algorithms. Figure 1 shows the placement of N heterogeneous tasks with sleep operations on heterogeneous resources with varying performance in terms of operations per second. The resources emulate three configurations: (1) N tasks are completely executed on the N Cloud devices; (2) N tasks are equally divided Edge and Cloud resources; (3) N tasks are completely executed on N Edge devices. The emulator provides key analysis on the configuration that leads to a response in well-defined bounded time.

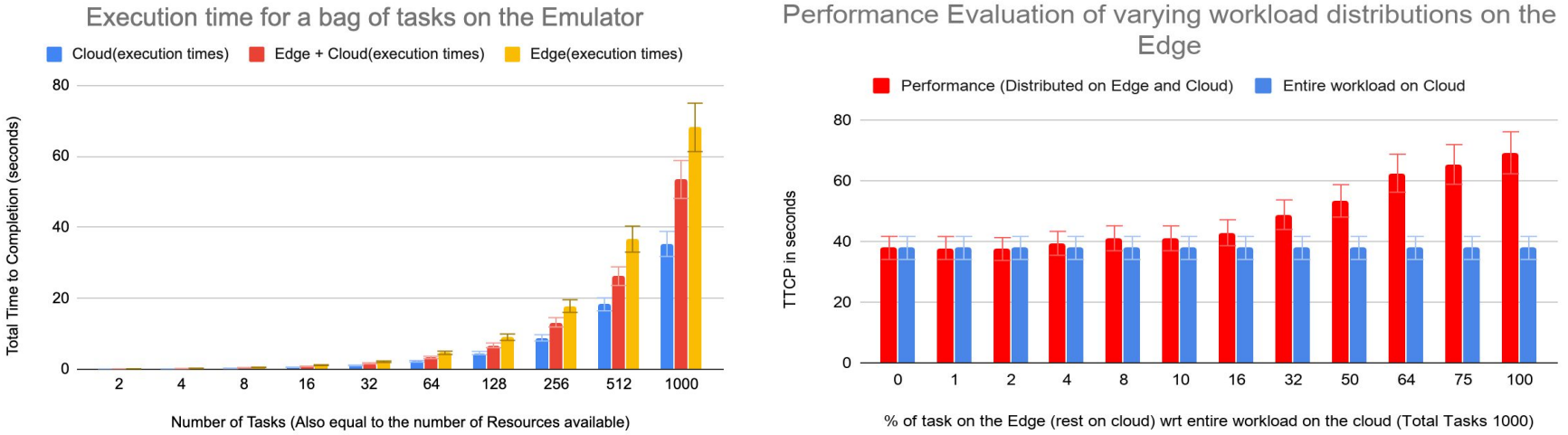

Figure 3. Performance Evaluation of distributing workload in a heterogeneous environment

Performance Vs Economic Model

The performance model helps us understand the distribution of workloads in an heterogeneous environment consisting of edge, cloutlet and cloud & HPC resources. It helps us understand suitable configurations required to compute a workload in a distributed cyberinfrastructure of edge devices, clouds and high-performance computing systems. This leads to determining the optimal distribution of training and inference over HPC/Clouds versus edge devices as well as determining efficiency and time-to-solution of ML (dependent on the problem, frameworks and platforms employed).

An economic/cost model can also provide potential users of the system with a meaningful understanding, for instance, it can help estimate the cost price involved in different configurations in an heterogeneous environment. Cloud and HPC services offer on-demand, pay-as-you-go, and reservation-based payment models. These cost models are variable depending upon the service, where users pay only for the individual services they need, for as long as they use them.

For instance, figure 3 provides the following trade-offs between a cost and a performance model for 1000 tasks (1) if TTC is a consideration (for instance, less than a minute) but the price to acquire resources is not, the user would choose blue and red configurations (2) If TTC is not a consideration but price to acquire the resources is, then use yellow configurations (3) If TTC is a consideration (less than a minute) and the price is also a consideration, use red configuration.

Future Work / Next Steps:

Currently, the emulator prototype is capable of executing heterogeneous workloads (tasks with sleep operations) on heterogeneous resources. In future work, we will extend the workloads and resources to approximate real manufacturing scenarios and input/output operations. We also propose to connect the results from the emulator to an economic model, how the metrics the emulator is reporting, e.g., total time to completion, data rates, can be used to get a monetary evaluation.

The Pilot-Abstraction (developed at RADICAL Lab) is a unified abstraction for resource management on heterogeneous infrastructure for high-performance & high-throughput computing, big data, and cloud for distributed applications. We are working on extending Pilot Abstraction to Pilot-Edge, the abstraction for resource management on Edge Computing devices. For this, we aim to develop an application simulator for edge environments, a Pilot-Agent for edge devices and integration with the Pilot-Streaming framework. Further, we will develop an application that processes data on both Edge Devices and cloud resources to evaluate the performance of the Pilot-Edge.