Cognitive packet network for self-aware adaptive clouds

Lan Wang
Intelligent Systems and Networks Group
Department of Electrical and Electronic Engineering
Imperial College London
Email: lan.wang12imperial.ac.uk

***Abstract*—This paper summarizes the RNN-based approach we have proposed to design an algorithm and develop a Task allocation Platform (TAP) for real-time task allocation in Cloud servers in response to user required QoS when there is a great diversity in the types of jobs, the class of QoS goals and the resources which are required by workloads and which are possessed by servers. This paper also present our ongoing work, inspired by the idea of Cognitive Packet Network , on an extension of TAP which distributes the intelligence at each server so that on arrival of a job, the server is able to decide whether to execute the job locally or dispatch it to another server to which it is connected in order to receive better performance.**

***Index Terms*—Random Neural Network, Reinforcement Learning, Sensible Algorithm, Task allocation, Cloud Computing, task Scheduling**

I. Introduction

Cloud computing is now widely integrated into the Internet. It enables the consolidation of an increasing number of applications from the general public or enterprise users which generate diverse sets of workloads in terms of resource demands and performance requirements [1]. Moreover, the heterogeneity in the hardware configuration of physical servers or virtual machines in terms of the specific speeds and capacities of the processor, memory, storage, and networking subsystems further complicates the distributing of applications to available machines. In order to address the challenge for Cloud service providers to dispatch incoming tasks to servers with the assurance of the quality and reliability of the job execution required by end users while also improving efficiency in the usage of resources, we have proposed an approach which uses Reinforcement Learning with the Random Neural Network to make fast, judicious and efficient decisions on the best path based on the knowledge learned from the past observations, while adapting to changes in workload and on-going performance of the network and Cloud environment. The approach benefits from limited measurement overhead and can be easily deployed over a large population of machines.

We have exploited the approach to design an RNN with Reinforcement Learning algorithm for real-time task-to-resource allocation in cloud servers. Inspired by the idea of Cognitive Packet Network [2], [3], [4], we have also designed and implemented the Task Allocation Platform (TAP) which carries out constant monitoring and measurement in order to keep awareness of the stat of cloud environment and service performance. With knowledge learned from these observations, the system can employ the task allocation algorithms that we have designed, to make online decisions to achieve the best possible QoS as specified by the owers of the tasks, while adapting to conditions that vary over time. TAP benefits from limited measurement overhead as it collect online measurements in an efficient manner and pay more attention to the part of the Cloud where better QoS can be offered, while still exploring less frequently a wider range of alternative systems that can in the future prove to provide improved QoS if the current set of frequently used subsystems result in poor QoS.

Since such schemes can be used more easily at the overlay level, we have also modified CPN so as to use it as an overlay network routing capability. We have shown that if the number of overlay hops is limited, a big data approach to CPN with data collected over long periods of time can result in significant improvements in end-to-end delay [5].

Cloud systems include both locally based servers at user premises and remote servers and multiple Clouds that can be reached over the Internet. We have designed a smart distributed system that combines local and remote Cloud facilities. It operates with a decision system that allocates tasks dynamically to the service that offers the best overall Quality of Service, and includes the effect of a routing overlay which minimizes network delay for data transfer between clouds.[6].

In this paper, we summarize the work we have done on real-time adaptive task allocation in Cloud servers. We also present our ongoing work on an extension of TAP which distributes the intelligence at each server so that on arrival of a job, the server is able to decide whether to run the job locally or dispatch it to another server to which it is connected.

II. Adaptive task allocation with a centralized cotroller

TAP, we designed in [7], composes of a centralized controller for accommodating task allocation algorithms and measurement agents on each host in a cloud for observing the system state related to the QoS requirements of end users. In TAP, end users are allowed to declare the QoS requirements related to the tasks they submit. TAP accepts these directions by translating the QoS requirements into one or more QoS metrics which constitute a function called the “goal function”. In this way, the QoS requirements are transformed into a goal function to be minimised, e.g. the minimisation of the task response time. The goal function determines which system parameters need to be measured and how task allocation will be optimised. RNN-based task allocation algorithm uses a distinct RNN to cover each distinct goal function G. Each neuron is identified with a particular host which can accommodate jobs. The algorithm effectively learns from the online measurements related to the given QoS goal(s) of tasks that have been allocated to hosts in the cloud and tends to increase the probability q_i of those neurons which correspond to hosts that yield a smaller value of G_i. In this way, each time TAP assigns a task to a host, it selects the host i that corresponds to the largest q_i and provides best possible QoS.

We have evaluated the proposed algorithms on a laboratory test-bed where TAP is implemented as a Linux kernel module which can be easily installed and loaded on any PC with the Linux OS. We have validated the RNN-based algorithm outperforms the Round Robin and sensible algorithm [8] in adaptively distributing workloads across available servers within a cloud in response to user required QoS when there is a great diversity in the types of jobs, the class of QoS goals and the resources which are required by workloads and which are possessed by server.

In the scenario where there are two types of jobs (CPU intensive jobs and I/O bound jobs) which are dipatched across heterogenous hosts having different processing capacities of the processor and I/O devices, the experimental results have shown that the RNN based algorithm outperforms the Round Robin and sensible algorithm and is able to dispatch I/O bound jobs to the hosts where I/O is less stressed and dispatch CPU intensive jobs to the hosts which provide better CPU capacity.

Workloads are often characterised differently in terms of their resource and performance requirements. Distinct QoS classes may also generate different levels of income for a cloud service, and will also have different running costs on different hosts. In addition, they will have distinct service level agreements (SLAs), and the violation of the SLAs will often have different financial consequences. Therefore, cloud service providers aims to maximize their profit by minizing the cost of the hosts for running jobs while maintaining service level agreements (SLAs). Based on this requirement, TAP defines a corresponding goal function. We setup an experimental scenario where there are three hosts which have disctinct processing speeds resulting in different running costs and two types of jobs which require distinct SLA levels. Experiments have shown that the RNN algorithm does better in reducing the overall cost while maitaining the SLA level specified by each type of job which results in the minimal SLA violation penalty.

In [9], we present a further study on the potential of the RNN-based algorithm in improving the energy usage efficiency of each available machine while maintaining the best possible job response time under varied load conditions overtime. optimizing the two contradictory criteria: reducing the energy consumption per job while maintaining the best possible job response time, which is of crucial interest for cloud providers. The resulting new approach improves the energy efficiency of each available machine with a given voltage supply by adaptively consolidating a certain amount of workload on the machine while optimising the specified response time.

III. Adaptive task allocation with distributed cotrollers

We extend TAP by injecting the intelligence into each host, which provides each host the ability to decide whether to execute the received job locally or to dispatch the request to the adjacent host which is able to share the workload (as shown in 1). That is to say, on arrival a job request at a host, the host will execute the requested job locally if it has enough capacity. If the workload in the host increase, the host could decide whether to forward the request to one of the adjacent hosts in order to balance the load and offer better QoS to all the requests it receives. The decision would rely on a variety of considerations, such as QoS, security, cost, and energy consumption. We currently only concern ourselves with the QoS that tasks receive, in particular the response time observed by tasks.

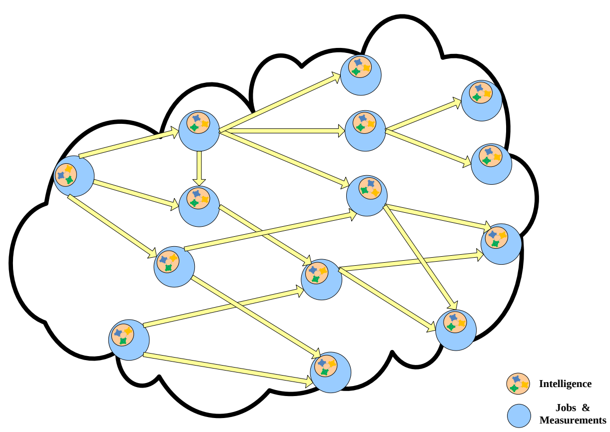

Fig. 1. Task allocation system with distributed controllers

Each host monitors the performance of itself and its adjacent hosts by collecting performance measurements at both the adjacent hosts and itself. Based on these on-line measurement related to the user required QoS, we design the Reinforcement Learning with the Random Neural Networt algorithm to make judicious task allocation decisions. Each RNN running on a host has as many neurons as the number of potential hosts which include the host itself and the hosts to which it is connected.

IV. Future work

A problem of RNN-based Reinforcement Learning algorithm is if the online measurements are not updated on time, the dispatching decisions might be made with errors. In the future, we will introduce data analysis techniques and machine learning approaches to modelling the system performance and workload patterns based on the historical measurement data collected at each host in order to correct the dispatching decisions.

References

[1] C. Delimitrou and C. Kozyrakis, "Qos-aware scheduling in heterogeneous datacenters with paragon," *ACM Trans. Comput. Syst.*, vol. 31, no. 4, pp. 12:1–12:34, Dec. 2013. [Online]. Available: http://doi.acm.org/10.1145/2556583

[2] E. Gelenbe, Z. Xu, and E. Seref, "Cognitive packet networks," in *Proceedings of the 11th International Conference on Tools with Artificial Intelligence*, Nov. 1999, pp. 47–54.

[3] E. Gelenbe, "Steps toward self-aware networks," *Commun. ACM*, vol. 52, no. 7, pp. 66–75, 2009. [Online]. Available: http://doi.acm.org/10.1145/1538788.1538809

[4] E. Gelenbe, R. Lent, and A. Nunez, "Self-aware networks and QoS," *Proceedings of the IEEE*, vol. 92, no. 9, pp. 1478–1489, Sep. 2004.

[5] O. Brun, L. Wang, and E. Gelenbe, "Big data for autonomic intercontinental overlays," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 575–583, March 2016.

[6] L. Wang, O. Brun, and E. Gelenbe, "Adaptive workload distribution for local and remote clouds," in *Proceedings of IEEE International Conference on SYSTEMS, MAN, AND CYBERNETICS (SMC2016)*, Budapest, Hungary, Oct. 2016.

[7] L. Wang and E. Gelenbe, "Adaptive dispatching of tasks in the cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 33–45, Jan 2018.

[8] E. Gelenbe, "Sensible decisions based on qos," *Computational management science*, vol. 1, no. 1, pp. 1–14, 2003.

[9] L. Wang, "Online work distribution to clouds," in *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Sep. 2016, pp. 295–300.