This abstract is presented at the 1st International Workshop on the Swarm at the Edge of the Cloud

Altocumulus: Harvesting Computational Resources from Devices at the Edge of the Cloud

YoungHoon Jung, Marcin Szczodrak, Richard Neill, and Luca P. Carloni
Department of Computer Science, Columbia University, New York, NY, USA
{*jung, msz, rich, luca*}*@cs.columbia.edu*

ABSTRACT

We present our vision for Altocumulus, a new model for heterogeneous computing that establishes an intermediate layer between cloud computing and the devices at the edge of the network by harvesting their resources and coordinating their use. We discuss this idea by illustrating examples of potential applications and summarizing the results of our prior work that provide some base frameworks for Altocumulus.

1. INTRODUCTION

The utilization of smart devices located at the edge of the cloud networks continues to accelerate, leading to new forms of heterogeneous cloud computing models. This trend has been influenced by a few factors. First, the number of intelligent embedded devices in the cloud network, such as smartphones and set-top boxes, has significantly increased. Second, the computational power of these devices continues to grow so that technologies for client-side computation are getting more public interest. More importantly, recent advancement of cloud technologies like Hadoop have facilitated the use of cloud computing and the development of an emerging class of user applications based on the cloud interfaces.

Keeping pace with this trend, more devices on the 'edge' of the cloud start to contribute to the computation and storage aspects. As illustrated in Fig. 1, the edge of the cloud is where the cloud network is connected to end users. Increasingly, computation and storage required for cloud application execution can be partially delegated to the intermediate layer in this network. We argue that the formation of this intermediate cloud layer is emerging across a variety of systems and name it *Altocumulus*[1].

Without physically relocating edge devices, their processing powers and networks can spontaneously participate to form Altocumulus. From the network topology point of view in Fig. 1, Altocumulus is viewed as 'intermediate' by a user device which accesses it from the edge, because it is located on the path from the device to the core of the cloud. Thanks to its structure, Altocumulus features shorter distance to the cloud, easier maintenance of data privacy, and lower network latency. Further, it enables new applications as well as cost savings by recycling existing and owned hardware.

2. ALTOCUMULUS APPLICATIONS

To illustrate more concretely the basic ideas of Altocumulus we discuss two applications that are suitable to this approach thanks to their layered, hierarchical designs. While each of these applications could be implemented also in other ways, to design them so that they can run partially on Altocumulus facilitates the use of resources from the edge devices, thereby alleviating the computation and communication load for the cloud.

[1] An *Altocumulus* is "a globular cloud at an intermediate height of about 2400 to 6000 metres (8000 to 20,000 feet)" [1].

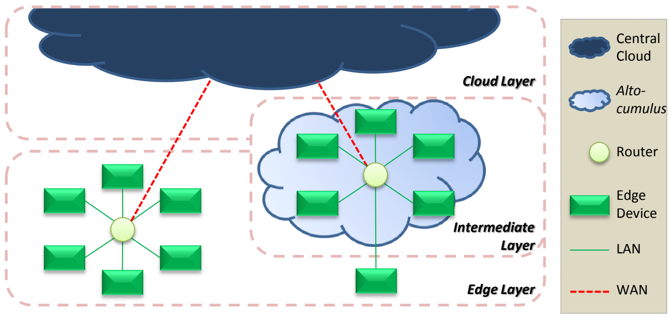
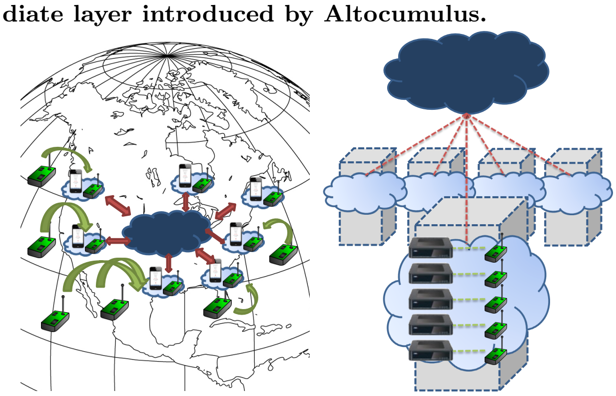


Fig. 1: A cloud network topology with the intermediate layer introduced by Altocumulus.



(a) Weather Data Analysis (b) Smart Buildings

Fig. 2: Two Altocumulus applications.

Weather Data Analysis. This application analyzes the National Climatic Data Center (NCDC) weather dataset [7] through the centralized processing of the weather data, including sky ceiling heights, visibility distances, temperature, and atmospheric pressure, gathered from across the globe. In the traditional approach, the centralized data processing burdens the cloud with a large amount of work that comes from tens of thousands of weather stations. With Altocumulus, this application can be implemented in a hierarchical fashion as illustrated in Fig. 2(a). Each Altocumulus collects data from nearby sensors. Then, only the summarized information necessary to analyze country-wide or worldwide weather statistics is transferred from the Altocumuli to the cloud. For example, the *MaxTemperature* command that finds the highest temperature from a specific region can be processed effectively as follows: each Altocumulus in the region processes the weather data from local sensors and reports summarized information that includes the local maximum temperature to the cloud. Then, the cloud finds the highest temperature among the reports from Altocumuli. This hierarchical approach offers lesser burdens on the cloud and shorter processing time due to its parallelized analysis of the data.

Smart Buildings. The goal of this application is twofold. First, each building manages the use of its resources such as electricity, water, and gas, in a smart way. Second, a cen-

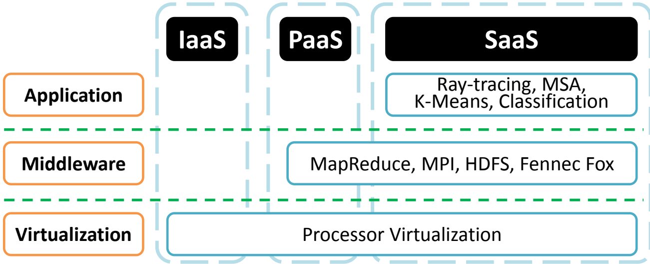


Fig. 3: The Altocumulus system stack.

tral control system monitors the resources in each building for the city-level management, e.g. forecasting power outages, improving resource distribution planning, and saving the energy. Mapping an instance of Altocumulus to a building and the cloud to the city can offer a good solution to this problem. Fig. 2(b) shows Altocumulus configured for each building, where an embedded devices such a set-top box or a base station is placed in every house to interact with a set of sensors. The information gathered from the sensors is stored in Altocumulus and analyzed for understanding the users' behavior patterns by executing machine learning algorithms. These patterns allow Altocumulus to manage the light and temperature control systems to reduce energy consumption of the houses in the building. For energy-consumption tracking and planning at the city-level, the Altocumulus of each building reports to the cloud statistical information about consumed resources and observed patterns. The collected building data is processed to extract the information necessary to the city management.

3. ALTOCUMULUS PROJECTS

To be effective, we argue that Altocumulus must offer the three major cloud computing service models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Fig. 3 describes the Altocumulus system stack for service models. In the System-Level Design (SLD) Group at Columbia University we have been developing various base frameworks and applications across these service models to contribute to the realization of the Altocumulus vision.

PaaS is the area where most of the research has been conducted so far. First, Neill *et al.* [5] demonstrated the idea of harvesting a broadband network of embedded consumer devices, such as set-top boxes (STBs), to provide a heterogeneous parallel computing platform based on the Message Passing Interface (MPI), a widely-used High Performance Computing (HPC) framework. Then, with the realization of a second generation system they showed the feasibility of virtualizing embedded processors to provide a scalable environment for users who want to access a heterogeneous system infrastructure elastically without knowing the details of the system [4]. Combined, these works have proposed Broadband Embedded Computing as an emerging area of research to develop a computing platform that leverages both a collection of embedded devices and a broadband network connecting them.

A more recent work brought this idea close to the cloud world: Jung *et al.* have ported Hadoop, one of the top cloud computing core computing technologies, to a heterogeneous cluster of blade servers and embedded devices [3]. This study established a redundant, reliable distributed file system, Hadoop Distributed File System (HDFS) as well as the very popular data processing computational framework, MapReduce, on an Altocumulus architecture.

Fennec Fox, introduced by Szczodrak *et al.* [6], shows the feasibility of deploying multiple heterogeneous applications on top of a single low-power wireless network. Heterogeneous applications execute without loss of performance because during context-switch, the network adapts the communication protocols to the ones that are required by the application currently executing. Also, Fennec Fox enables the scheduling of execution of the cloud tasks across the low-power wireless networks, which find applications in cyber-physical systems to ensure data privacy.

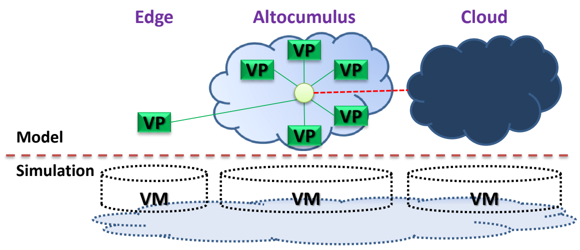


Fig. 4: The VP-on-VM model for Altocumulus.

A number of applications have been ported to the PaaS over Altocumulus, thus providing SaaS examples: Raytracing and Multiple Sequence Alignment (MSA) run on Altocumulus using MPI, while many MapReduce applications including data mining programs such as K-Means and Classification execute on Altocumulus using Hadoop's MapReduce framework.

Finally, Jung *et al.* have developed NETSHIP, a novel CAD tool for the analysis, design, and simulation of applications that run from the edge of the cloud, across the Altocumulus intermediate layer, and on the data servers at the core of the cloud [2]. NETSHIP uses the novel VP-on-VM model, which executes multiple Virtual Platforms (VP) on a Virtual Machine (VM) instance provided by cloud computing, as shown in Fig. 4. The VP-on-VM model allows users to easily build, manage, and scale up the target system.

4. CONCLUDING REMARKS

One of the critical considerations for Altocumulus design is scalability. Any distributed system can certainly pull some computational power out of the devices in the network. However, in a system with a poor scalability, gathering enough computational power for a large problem might be inefficient. Each of our prototype systems has shown through their own experiments that they are scalable.

Also, to find the optimal proportion for computational and storage delegation between the cloud and Altocumulus is one of the pivotal areas of future research.

By forming Altocumulus the end devices enable new user-oriented applications and provide the necessary resources to low-power wireless embedded devices, thus establishing an important foundation for the Internet-of-Things and Cyber-Physical Systems applications.

5. REFERENCES

[1] HarperCollins Publishers Ltd. *The Collins English Dictionary*.

[2] Y. Jung et al. netship: A networked virtual platform for large-scale heterogeneous distributed embedded systems. In *Proc. of DAC*, pages 1–10, June 2013.

[3] Y. Jung, R. Neill, and L. Carloni. A broadband embedded computing system for MapReduce utilizing Hadoop. In *Proc. of CloudCom*, pages 1–9, Dec. 2012.

[4] R. Neill et al. Embedded processor virtualization for broadband grid computing. In *Proc. of Grid*, pages 145–156, Sept. 2011.

[5] R. Neill, A. Shabarshin, and L. P. Carloni. A heterogeneous parallel system running Open MPI on a broadband network of embedded set-top devices. In *Proc. of CF*, pages 187–196, May 2010.

[6] M. Szczodrak, O. Gnawali, and L. P. Carloni. Dynamic reconfiguration of wireless sensor networks to support heterogeneous applications. In *Proc. of DCOSS*, pages 52–61, May 2013.

[7] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2009.