Scalability of Vehicle Networks through Vehicle Virtualization

Jiangchuan Huang
Systems Engineering Group
Univ. of California, Berkeley
Email: jiangchuan@berkeley.edu

Christoph M. Kirsch
Dept. of Computer Sciences
Univ. of Salzburg, Austria
Email: ck@cs.uni-salzburg.at

Raja Sengupta
Systems Engineering Group
Univ. of California, Berkeley
Email: sengupta@ce.berkeley.edu

We explore extending the paradigm of cloud computing to computing tasks having locations in space. A computing task is a triple $(ArrivalTime, Location, ComputingTime)$. A server is considered to execute such a task by visiting the task $Location$ and staying there for $ComputingTime$, at any time after the task $ArrivalTime$. Sampling applications in time and space, such as those entailed by Google street view [2], mobile sensor networks [8], or real-time traffic reporting by radio stations, are examples of this type of computing task. The servers, henceforth referred to as real vehicles, are networked vehicles each having all or some of the sensing, computation, communication, and locomotion capabilities. Examples include robots like driver-less cars, drones, or manned helicopters, traveling and then pausing to execute computing tasks like taking pictures, measurements or monitoring at specified places. We organize the collection of moving servers as a new type of cloud called the spatial cloud, or the Cyber-Physical Cloud (CPC). CPC may work better than conventional cloud computing because of migration gain, a concept unique to the spatial cloud introduced below.

We extend the idea of the virtual machine [7] used in cloud computing to an idea called the virtual vehicle to create performance isolation [7]. Just as the cloud computing customer has a service-level agreement (SLA) for a virtual machine, our customer has an SLA for a virtual vehicle. To a customer, a virtual vehicle is exactly like a real vehicle that travels at a virtual speed specified in the SLA. For example, suppose a radio station (the customer) uses a helicopter to overfly accident scenes (the task) at 20 mile per hour (mph) for real-time traffic reporting. Such a helicopter would arrive at an accident scene 2 miles away in 6 minutes. We now virtualize this helicopter. Instead of buying or renting, and operating a real helicopter, the radio station would buy virtual helicopter that travels at 20 mph using an SLA. Our cyber-physical cloud would then take responsibility for flying some real helicopter to the accident site in 6 minutes or some near approximation to this time. Our results show that small deviations from the 6 minutes can turn into large numbers of virtual vehicles being realized by many fewer real vehicles. This is the cyber-physical cloud computing gain. The radio station could enjoy large reductions in cost by tolerating small variations in the 6 minutes.

The real vehicles travel to and execute each task such that the real completion time is no later than the expected completion time. The system achieves high performance isolation if a statistically dominant subset [5], e.g., 98%, of the virtual vehicle's tasks are completed no later than their expected completion time such that the customer does not even notice a small overflow over expected completion times. The expected completion time behaves like a "deadline" to the provider. We call it the virtual deadline. The virtual deadlines make the spatial cloud a soft real-time system [3], [1]. We use performance metrics such as tardiness and delivery probability to measure performance isolation. How the provider realizes the virtual deadlines of the tasks is of no concern to the customer. For example, the provider can use a real vehicle to travel to and execute a task as the virtual vehicle does, or migrate the information of the virtual vehicle through a network to another real vehicle closer to the task. In the latter case, the real distance traveled is smaller than the virtual distance resulting in the phenomenon we have called migration gain. The idea works when the communication costs of migrating a virtual machine over the network are small.

The theorems and simulation results show that the provider can support a given number of virtual vehicles with significantly fewer real vehicles while guaranteeing high performance isolation. We quantify the gain by the ratio of the number of virtual vehicles over the number of real vehicles. The gain arises from two phenomena. (i) A customer may not fully utilize her virtual vehicle, enabling the spatial cloud to multiplex several virtual vehicles onto one real vehicle. This type of gain is called multiplexing gain. It is known in communication networks [4] and cloud computing [6]. (ii) The real vehicles save travel distance by migrating the virtual vehicle hosting a task to another real vehicle closer to the task, creating migration gain. Our results focus on migration gain since it is unique to the spatial cloud. Migration gain is the reason spatial cloud computing outperforms conventional cloud computing.

Research supported in part by the National Science Foundation (CNS1136141) and by the National Research Network RiSE on Rigorous Systems Engineering (Austrian Science Fund S11404-N23).

References

[1] L. Abeni and G. Buttazzo. Qos guarantee using probabilistic deadlines. In *Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on*, pages 242–249, 1999.

[2] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43, 2010.

[3] G.C. Buttazzo. *Soft real-time systems: predictability vs. efficiency*. Series in computer science. Springer, 2005.

[4] A. Host-Madsen and A. Nosratinia. The multiplexing gain of wireless networks. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 2065–2069, 2005.

[5] Gerald J. Popek and Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. *Commun. ACM*, 17(7):412–421, July 1974.

[6] Benny Rochwerger, David Breitgand, Amir Epstein, David Hadas, Irit Loy, Kenneth Nagin, Johan Tordsson, Carmelo Ragusa, Massimo Villari, Stuart Clayman, et al. Reservoir-when one cloud is not enough. *Computer*, 44(3):44–51, 2011.

[7] Mendel Rosenblum. The reincarnation of virtual machines. *Queue*, 2(5):34–40, July 2004.

[8] Arun A. Somasundara, Aditya Ramamoorthy, and Mani B. Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Proceedings of the 25th IEEE International Real-Time Systems Symposium*, RTSS '04, pages 296–305, Washington, DC, USA, 2004. IEEE Computer Society.