The Cloud and the Economics of the User and Customer Experience

THE EXPERIENCE THAT USERS SUCH AS CUSTOMERS AND EMPLOYEES HAVE OF PRODUCTS, SERVICES, AND PROCESSES IS INCREASINGLY THE KEY BATTLEGROUND FOR GLOBAL COMPETITION. It encompasses numerous physical, tangible dimensions, including design, aesthetics, and materials. Consider products such as a cool smartphone, an elegant gold watch, or the physical characteristics of healthcare services, such as the color of hospital walls or the appearance of a medical diagnostic machine, or the ambience of dining services, including the food presentation. Today, of course, a substantial portion of the user experience is based on virtual components helped or hindered by application characteristics such as response time, performance, availability, and interface design. Therefore, the total user experience can be helped—or hindered—by the cloud. In food service, the dining experience is a combination of the presentation and service experience: customer-facing processes, such as how the food is transported from the kitchen to the table, and back-end processes, such as cooking and plating. Quality back-end processes can aid in the presentation or vice versa. In virtual services—say, e-commerce or online entertainment—the complete experience is also a function of the presentation (layer) and service experience, and related processes such as (network) transport and back-end processing.

Time is a key dimension of the user experience. For example, a website—or app—that takes too long to load negatively affects the user's perception of the service quality, or worse, means that a user will switch to a competitor's offering. Slow internal applications will hinder productivity. Broadly speaking, as shown in Figure 1, the total response time for a cloud-enabled application depends on processing within the end device, such as a tablet or sensor, including activities such as context switching, interrupt processing, and network stack traversal; end-to-end network latency, based on physical propagation delays, router hops, and retransmission due to packet loss; and remote cloud processing time, which can be impacted by architecture, noisy neighbors, load balancing approaches, virtual machine (VM) or container provisioning intervals, and the degree of parallelization.

This month's column will explore some of the economics of time and the user experience.

The Business Value of Time

Cloud computing and complementary approaches such as DevOps can increase the ability of businesses to respond proactively or reactively to threats or opportunities. For example, infrastructure as a service can reduce provisioning intervals dramatically, enabling businesses to scale up to meet market demand. Platform as a service and microservices can enable firms to accelerate their development efforts.

JOE WEINMAN

joeweinman@gmail.com

2325-6095/15/$31.00 © 2015 IEEE

Time also plays a key role not only during development and scale up, but in enhancing the user experience during runtime operations.

For rote knowledge work using internal applications, such as processing insurance claims or invoices or staffing a contact center, the first-order effects of the loss are easy to determine, as is the gain from better performance. It's based on the percentage of time that a user uses the application, and the application's speedup or slowdown. For example, let's assume that a warehouse worker needs to identify the next order item to be picked, and then goes to the specific bin to pick it. If the information-based step takes 10 seconds, and the picking operation takes 90 seconds on average, then speeding up the first step by 50 percent (to 5 seconds) will reduce cycle time by 5 percent and thereby improve productivity by just over 5 percent (100/95 = 1.0526). These effects can be substantial: it's not uncommon to hear a customer service representative say "our computers are slow today," and in a firm with 1,000 such agents, this is the equivalent of 50 such agents. Of course, there are additional complexities, such as whether tasks can be overlapped or pipelined, the accuracy of inventory or customer data, and so on.

For more complex tasks, improved application performance can potentially have outsized effects. For knowledge workers engaged in collaboration for innovation, reduced time might mean getting a better product to market more quickly, leading to first-mover advantage and dramatically higher revenue and profit.

Perhaps the ultimate example of the benefits of time compression lies in financial markets, where a few microseconds might mean the difference between successfully executing a trade that exploits a momentary arbitrage opportunity and watching a competitor get there first.

A number of studies have empirically shown the business value of faster response times. For example, a substantial fraction of people will abandon desktop or mobile websites if they take more than a few seconds to load. Longer times are correlated with a decrease in "conversions," such as a prospect making a purchase, a decrease in customer satisfaction, fewer page views, fewer return visits, and so on.[1] Results from Google showed that a half-second delay in presenting search results led to 20 percent fewer click-throughs.[2] When your business revenue is highly correlated with clicks, this is substantive.

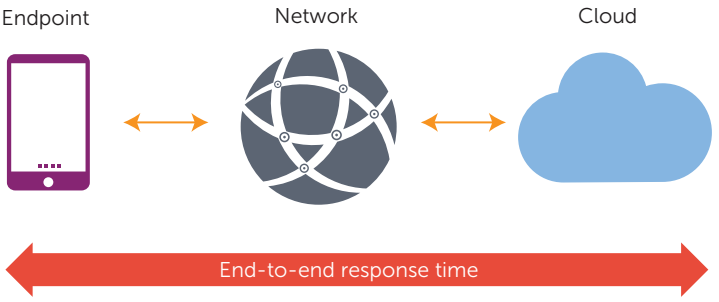

FIGURE 1. End-to-end response time depends on the endpoint, the network, and the cloud.

Users, whether customers or employees, or whether in other categories such as partners, shareholders, or citizens, represent one major class of interactivity, in terms of users interacting both with systems, such as querying a Web search service, or with each other, as in immersive videoconferencing or "telepresence." Other computer applications, servers, storage systems, and physical devices making up the Internet of Things are another class.

For example, consider motor vehicle collision detection. Although in-car systems are becoming increasingly autonomous and sophisticated, there are limits to the technology. For example, an in-vehicle system can help ensure that the car stays within lane boundaries, and can detect an impending collision with a car ahead of it, seemingly reducing the importance of external capabilities. But, consider two vehicles racing toward the same intersection in a city, say one heading north and the other west. Because of a building, the northbound car can't see the westbound and vice versa. Only a common system that remotely alerts each car of the other's presence can help avoid a collision, and a projection of trajectories and alerting in near real time is critical.

Numerous other examples exist where time is critical for connected things, such as coordinating the movements of cranes at a construction site; a surgeon conducting telesurgery; vehicular coordination, as between a grain combine, which reaps the grain, and a grain cart, which follows alongside and then carries the grain to a silo; or remote operations, such as of a mine or rescue robot.

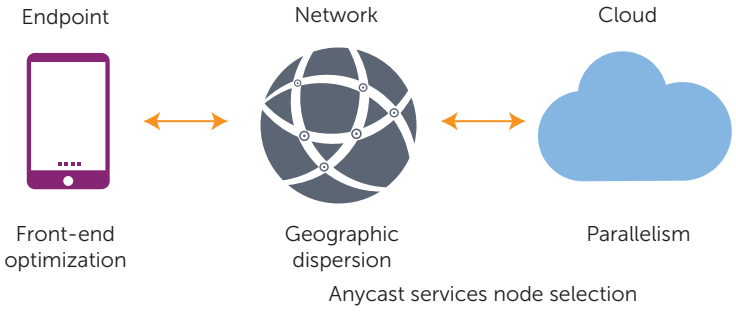

FIGURE 2. Generic approaches to user experience enhancement.

End-to-End Optimization

To optimize the user experience or device functionality and reliability, there are many possible points of attack that involve the cloud to a greater or lesser extent, as Figure 2 shows. As examples of the former, consider elements of webpage, application, or mobile app design that enhance navigability, improve aesthetics, reduce complexity, or provide an element of surprise or delight. For example, Google's homepage (www.google.com) is famous for its sparse aesthetic, its A/B testing of numerous color choices, and Google "doodles"—the sometimes whimsical illustrated or animated variations on Google's logo. One important category of user experience enhancement is "front-end optimization," a broad technique that applies tactics such as reducing the size and number of objects to speed rendering and thus enhances experience. It works because the time to load a webpage is partly a function of the number of objects in it and thus the time needed to request and receive the objects. It's also a function of image processing at the endpoint: a large image that is resized to be smaller by specifications in the HTML takes more time to transmit to the endpoint and takes additional CPU or GPU cycles to resize and render.

In addition, for a global audience, the geographically dispersed nature of the cloud can be used to reduce the physical distance and thus network propagation delay between the endpoint device (machine, sensor, or user device) and a processing node. Second, parallelism (within a datacenter) can be used to reduce the interval between starting a task such as a query and completing it. Third, optimized selection of the "best" service node among many capable of providing a service can improve average response times.

Cloud Dispersion

Geographic dispersion can help reduce processing time by reducing the distance and thus the propagation delay between a user or machine endpoint and the cloud-based service node. Large cloud and colocation providers have global footprints; so do large content delivery networks. This principle reduces both expected and worst-case network transport latency.

There are some details to consider. At a macroscopic level, the point-to-point shortest path distance isn't always the path that a data packet will take. For example, Spread Networks (www.spreadnetworks.com) has created a near-optimal fiber network between New York and Chicago, with a path that is close to the shortest possible and with a network architecture that minimizes delays due to amplification and regeneration. On the other hand, routes from, say, Key West to Seattle start off heading in the wrong direction (Key West to Miami!) and then zig-zag either north or west rather than directly northwest due to how fiber routes are laid out. At a more detailed level, there are issues such as packet loss, router hops, and latency induced by limited bandwidth, network outages, or congestion.

However, generally speaking, latency is proportional to distance. Therefore, it's easy to calculate the relationship between the number of service nodes and worst-case latency on a plane. The area A covered by a service is proportional to the number of service nodes n and the radius of coverage r based on the area of a circle with radius r. Simply put, $A \propto n\pi r^2$. The exact constant of proportionality depends on the packing density η. This in turn means that for a given area, if worst-case latency l is proportional to maximum distance from a service node, since π and A are constants, then $l \propto r \propto 1/\sqrt{n}$. There are some additional complexities since we're concerned with latency on a sphere (planet Earth) rather than a plane, necessitating a slight trigonometric adjustment,[3] but this inverse square root function is fine for our purposes. It means that huge improvements in network latency are possible by adding relatively few service nodes to an existing small number. Unfortunately, it also means that when there are many service nodes already in place, eking out further gains is prohibitively expensive. Consequently, merely dozens or scores of geographically dispersed nodes can suffice to offer very good response time

for many types of computing tasks to a global audience, and this is exactly what the large cloud providers appear to be pursuing in terms of footprint. As costs and minimum required footprints come down, more distributed architectures such as microcells, edge, and fog can lead to further latency reductions.

For enterprises pondering the relevance of the cloud either in pure form or as part of a hybrid architecture, in addition to cost, the improvement in user experience is also relevant. Moreover, dispersion can largely be accomplished without worsening cost. A service that utilizes 100 servers, VMs, or container instances in a single location can be offered for roughly the same cost using five servers, VMs, or container instances in each of 20 locations. There are additional pros and cons, such as bandwidth cost savings, data replication costs, volume discounts, statistical multiplexing effects, and the like, but the point is clear.[4]

Cloud Parallelism

Although dispersion helps reduce time for network transport to and from a service node, elastic cloud resources can help reduce the time needed at the service node through parallelism. For example, a task that might take 100 seconds on a single processor might take only 10 seconds if run on 10 processors, 1 second on 100 processors, or 100 milliseconds on 1,000 processors. Tasks that are "embarrassingly parallel" exhibit an exactly inversely proportional relationship. This category generally includes map-reduce processing such as running a search query against index shards. The idea for such a task is that if it takes time T on a single processor or core, it will take time T/p on p processors or cores.

This is a high-level observation, because there are many different types of parallelism. At one extreme, tasks might be distributed across a wide geographic area, but interlocation latency would make interprocess communication prohibitively expensive. At the other extreme, one might consider parallelism within a GPU, or even, conceptually, within a quantum computer. Here, we're referring to parallelism in a scale-out architecture within a single physical location—that is, a cloud datacenter.

Algorithms typically have sequential and parallel elements. If such code is equally divided, say, running on two processors will not reduce the time by half, but only by a quarter, since only half the code is sped up. Put simply, if the serial portion takes time S and the parallel portion takes time T when run without parallelism, the time taken on p processing elements will be $S + (T/p)$.

In the real world, such tasks also require interprocess communication, and the associated overhead can reduce the actual speedup to less than the theoretical maximum, making the total time required more than T/p. For example, one analysis of the use of Hadoop suggests that actual performance depends on the specific hardware, network performance, how the cluster is set up, and the specific algorithm and use of mapping and reducing components.[5]

Interestingly, the cloud not only offers elasticity (that is, nearly immediate access to nearly "infinite" resources), it also offers pay-per-use pricing. This means that, subject to minimum billing increments or sustained-use discounts, speedup through parallelism (within the same datacenter) can be essentially free. For scale-out resources with linear pricing, 100 server or VM hours cost just as much whether it's 1 unit for 100 hours or 100 units for 1 hour, or 1,000 units for 6 minutes, and so on. This calculus changes somewhat if there is nonlinear pricing and depending on exactly how a parallel algorithm utilizes resources.

Anycast Service Optimization

Sitting at the intersection of geographic dispersion, network transport, and elasticity lies an approach that can potentially eke out further gains. Anycast services are available at multiple locations, the same way that a Mocha Frappuccino is available from many different Starbucks cafés.

If there are a number of cafés within a short distance of where you are, with perfect information you would select the one "closest" to you, accounting for highway routes versus side roads, traffic congestion at that time of day, and the one with the "fastest" service, based on the length of the line, the complexity of the intended orders of the patrons, and the skills and resources of the baristas and the café.

Similarly, with perfect information, one would select a service node with the lowest network latency, based on network bandwidth and congestion, as well as node processing time, based on raw capacity as well as other jobs sharing those resources.

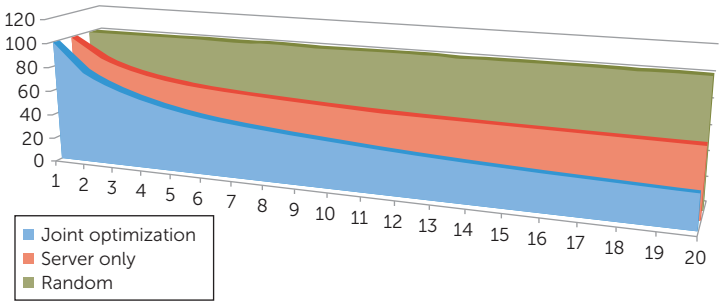

FIGURE 3. Expected value of random, server or network-based, and joint optimization.

If we assume that roundtrip network latencies are uniformly distributed on [0, 1], that the response times at a processing node such as a cloud datacenter are also uniformly distributed on [0, 1], and that the two variables are independent, then the following results apply when there are n choices as to where to run a job. If the node is randomly selected, the expected value of the network time is 1/2 and the expected value of the processing time is also 1/2. According to the theory of order statistics, for a random variable uniformly distributed on [0, 1], the expected value of the minimum item for n items sampled from the distribution is $1/(n + 1)$. Consequently, if either the best node or the best path is selected, the expected value of the total response time would be $1/2 + 1/(n + 1)$. However, when the best combination of node and path is always selected, the expected value of the total response time can be roughly approximated by $\sqrt{\pi/2n}$.[6] In other words, in an anycast environment, for users with a choice of nodes that are equivalent on average, selecting the "best" node at any given time results in a speedup over a single instance. As Figure 3 shows, as n (the number of options) increases, the denominator gets larger and the expected value of the response time decreases. Of course, such results depend on a variety of details, including the stability of network congestion, the ability to conclude a service interaction on a single node or dynamically and efficiently migrate it to another node in midstream, and so on.

THE CLOUD IS OFTEN CONSIDERED A MEANS TO REDUCE COST. However, it can be equally viewed as a mechanism for improving the user experience, and thereby either improving the customer experience and thus revenues, or the employee experience and thus labor productivity and business agility through enhanced collaboration. In addition, the performance and safety of connected things can be enhanced. Key attributes of the cloud that help achieve these gains include geographic dispersion, elastic resources, pay-per-use pricing, and increasingly, software-defined intelligence in network and computing resources.

References

1. A. Price, "Infographic: Web Performance Impacts Conversion Rates," blog, 9 Apr. 2014; http://loadstorm.com/2014/04/infographic-web-performance-impacts-conversion-rates.
2. S. Shankland, "We're All Guinea Pigs in Google's Search Experiment," CNET, 29 May 2008; www.cnet.com/news/were-all-guinea-pigs-in-googles-search-experiment.
3. J. Weinman, "As Time Goes By: The Law of Cloud Response Time," working paper, 12 Apr. 2011; http://joeweinman.com/Resources/Joe_Weinman_As_Time_Goes_By.pdf.
4. J. Weinman, *Cloudonomics: The Business Value of Cloud Computing*, John Wiley & Sons, 2012, pp. 269–272.
5. J. Kestelyn, "How-to: Tune MapReduce Parallelism in Apache Pig Jobs," blog, 16 July 2015; http://blog.cloudera.com/blog/2015/07/how-to-tune-mapreduce-parallelism-in-apache-pig-jobs.
6. H.N. Nagaraja, "Moments of Order Statistics and L-Moments for the Symmetric Triangular Distribution," *Statistics & Probability Letters*, vol. 83, no. 10, 2013, pp. 2357–2363.

JOE WEINMAN *is a frequent keynoter and the author of* Cloudonomics *and* Digital Disciplines. *He also serves on the advisory boards of several technology companies. Weinman has a BS in computer science from Cornell University and an MS in computer science from the University of Wisconsin-Madison, and has completed executive education at the International Institute for Management Development in Lausanne. He has been awarded 22 patents. Contact him at joeweinman@gmail.com.*