The Evolution Of Networked Computing Utilities

Joe Weinman

What's driving the emerging world of distributed utility computing, and where is it headed?

The traditionally separate spheres of IT and networking are converging, driving a potential transformation. Much more than just unified communications or net-sourced applications, entirely new distributed processing architectures are emerging that complement centralized datacenters with content and application distribution to support environments ranging from online gamers' multimode experiences to Web and mobile users' rich Internet applications.

One implication of this shift is that enterprises need to take a more nuanced approach to datacenter consolidation and centralization. While this is a key initiative for many enterprises, there is a legitimate benefit to dispersing some resources and functions to maximize flexibility, reliability, scalability and performance while minimizing latency and cost.

Beyond dispersion, these next-generation architectures are also leveraging the latest developments in dynamic resource allocation and virtualization. In addition, customers are becoming able to further minimize cost and enhance scalability by leveraging emerging utility (also sometimes referred to as pay-for-use, pay-per-use, or pay-as-you-go) pricing models.

The potential exists for this new paradigm of a networked computing utility to enable globally optimal architectures—or to create chaos if management and control are insufficient.

Networked Utilities

Utilities are everywhere: electricity, cable, water, gas and telephony. Many definitions of a utility exist, focusing on aspects such as ubiquity, commodity, public access, service reliability and government regulation. For the purposes of this article, we will define a utility as a business that combines flexible use with variable pricing.

A utility can therefore be characterized by what is flexible, and how that impacts the price. In an electric utility, the quantity of kilowatt-hours used is flexible, and the price is based on total quantity consumed. In a hotel utility, price is determined by the quantity of rooms rented by a customer, the duration of the stay and the quality of service, defined by type of room, e.g., standard, deluxe, junior suite or the presidential suite.

In a networked IT utility, similar characteristics hold. Thus, a storage utility might be priced based on quantity (Gigabytes), duration (months) and quality (e.g., enterprise class synchronously mirrored). A network utility might be priced based on quantity (bandwidth), duration (seconds or years), and quality (latency, jitter, packet loss).

Other pricing and payment models are possible and exist in other industries, such as the pay-for-value model inherent in a lawyer's contingency fee based on awarded damages, or a pay-for-process model with progress payments. However, evolving networked computing utilities are almost universally based on payment for resource use or allocation. Like a hotel room or rental car, a customer pays for one or more resources that will be allocated to and/or used by them.

A utility grid—or networked utility—takes a utility pricing model even further by placing resources at geographically dispersed locations, linking them, and allowing customer demand to be balanced or otherwise optimized at these locations. Hotel and rental car chains and the electric grid are commonplace examples of networked utilities. Another is the emerging world of distributed, hosted utility computing resources tied together by a global data network.

The network is an essential component to value creation in such a utility. Dispersion alone is insufficient. Abstractly, this need not be a communications network: a car rental company that allows drop-off at a location different from pick-up, an electric utility that is part of a larger grid, or a hotel chain that awards points regardless of property, all would qualify.

Networked IT utilities are beginning to appear, based on work in standards bodies, hardware vendor and service provider initiatives, and the Internet and private data networks as enablers. But important questions are beginning to arise: What are the essential enablers for a service provider-networked utility? Are there key technologies,

Joe Weinman is responsible for emerging services and business development for a large global telecommunications service provider. He has a BS and MS in Computer Science from Cornell University and UW-Madison respectively, and has completed Executive Education at the International Institute for Management Development in Lausanne. He has been awarded 10 U.S. and international patents, and is a frequent industry speaker globally on strategy and emerging technologies. The views expressed are his own. He can be reached at joeweinman@gmail.com.

components and architectures? How much value is created? How is it created? And are there inevitable trends that a networked IT utility will predictably follow?

Networked Utility Requirements

There is a logically derivable set of requirements for a networked utility. Although this article is focused on networked computing, requirements and capabilities are not that different than for hotel or rental car utilities. At a high level, these include:

- **Demand Multiplexing**—For a utility, having dedicated resources without the possibility for sharing may be uneconomical to the provider of the service. If a hotel company dedicated each different room to a specific customer for the life of the building, but didn't charge a cent if that person never showed up, the hotel chain would quickly go out of business. In an IT utility, the ability to map and bind different customers to different individual or clustered resources is key.
- **Dynamic Resource Allocation**—If resources are time-division multiplexed, there must be mechanisms for allocating and de-allocating them to different customers over time. In a hotel utility, this involves checking in and checking out. In an IT utility, virtual server hypervisors, operating systems, middleware, data and applications must be loaded onto or configured for access by physical servers or de-allocated via a control layer.
- **Dynamic Partitioning**—If resources are space-division multiplexed, there must be mechanisms to enable partitions to be created and resized. These may be below the level of a CPU or core, e.g., a virtual server using 20 percent of a processor's capacity, which is similar to partitioning a hotel ballroom. Or, multiple components may be clustered to behave as a single symmetric multiprocessor, the same way that two hotel rooms may be joined into one by unlocking the door between them.
- **Virtualization**—If resources are not dedicated, and may vary over time, there must be decoupling between the logical construct and the physical, together with a binding that is specified during dynamic resource allocation. For example, a non-smoking room with a king size bed and pool view is a logical construct that gets mapped to a specific room at or before check-in. A virtual server running a particular application may be mapped to a particular physical server, and this mapping may change over time.
- **Network-Based Access/Delivery**—Getting these virtualized, dynamically allocated partitioned resources connected with a customer requires some sort of network. Whether it is a network of highways and hallways enabling a customer to access the hotel or their room, or WANs, MANs and LANs in a distributed computing utility, the network is key.
- **Security**—Given the multi-tenant environment, it is essential to ensure that simultaneous users of the environment do not interfere with each other, as well as that a prior occupant does not interfere with a subsequent one or vice versa. Door locks keep simultaneous customers of a hotel utility from entering each other's rooms; various security elements such as firewalls, VLANs and IP or MPLS VPNs accomplish the same thing in a networked utility. Housekeeping cleans rooms between stays, to prevent a later occupant from digging through the trash for a prior occupant's data, or to keep a prior occupant from bugging the room. Data shredding and version and driver auditing achieve the same objective in an IT utility.
- **Management**—Flexibility without control goes by another name: chaos. Like a front-desk manager faced with a difficult situation, managing the activities involved in secure dynamic allocation and de-allocation of resources during shifting demand, disasters or other difficult conditions may be challenging. Utility environments, more than the traditional configuration and patch management, or break-fix, must move to a high degree of not just complex orchestration but also optimization, both to reduce internal costs and provide a differentiated customer value proposition.
- **Global Interconnection**—Treating the physically dispersed resources as a single logical pool is essential to delivering customer value and maximizing brand and scale economies. Scalability, business continuity and latency reduction are additional key benefits of such an architecture. This is not limited to IT. If there are multiple hotels in a chain, when one is fully committed, the front desk can refer you to another nearby hotel in the chain. If one suffers a disaster, other nearby hotels can accept the reservation. And, although we don't tend to use the term "latency reduction" in such a context, that is exactly what happens when one has business in Chicago and stays in a hotel there rather than staying in another hotel in that chain located in Hong Kong.

Networked IT Utilities

With the above criteria in mind, we can see that the converging IT and telecom industry is moving towards a highly utilitized and distributed set of capabilities. Although there is substantial hype, there has also been real progress: in terms of technology, carrier-class operationalization of this technology, and real business benefits such as enhanced flexibility and scalability at reduced cost:

- **Wide Area Networks**—The WAN is becoming utilitized and virtualized. For example, AT&T has introduced Optical Mesh Service, which provides dynamic resource allocation (e.g., bandwidth in STS-1 increments or decrements), with rapid response time. In fact, a 4xSTS-1 connection can be enlarged to become a 7xSTS-1 connection in seconds. MPLS networks enable QOS flexibility. And control capabilities, such as AT&T's Intelligent Routing Service Control Point, enable a high

The WAN is becoming utilitized and virtualized

Utility desktops are now possible

degree of route control, using a control plane to inject forwarding tables into routers.

Next-generation technology such as Reconfigurable Optical Add-Drop Multiplexers (ROADMs), coupled with optical cross-connects, enable large-scale utility bandwidth and route control for paths at 2.5 Gbps, 10 Gbps, and beyond. Other elements now entering the domain of what had formerly been transport only, such as network-based firewalls, are also now virtualized and shared, and ultimately will become utilitized.

■ **Local Area Networks**—Although Inkra Networks, one of the innovators in creating virtualized LAN resources, is no longer in business, a number of its intellectual property assets have been acquired by major network equipment vendors such as Cisco and Nortel. Capabilities pioneered by Inkra are being incorporated into network vendor products for virtualized and dynamically partitionable and customizable elements such as Layer 2 switches, SSL accelerators, load balancers and firewalls.

■ **Servers**—Dynamically allocatable physical and virtual servers are available as turnkey hardware systems, proprietary software that runs on open hardware, and from hosting service providers, e.g., USi's USiPinnacle service platform. Virtual servers appear to the customer to be real servers, but are emulated by a construct typically referred to as a hypervisor or kernel. Hypervisors may be implemented in software, such as VMware's ESX, or in firmware providing the partitioning support. Some companies such as 3Leaf Systems are even working with chip manufacturers to build in silicon-level hypervisors.

Service providers are engaged as well. For example, in one business model, such as AT&T's Managed Utility Computing for Sun servers, dedicated servers are variably priced based on average CPU utilization. In another model, they are priced based on duration of allocation. In yet another model, they are priced based on how much of the server is allocated: number of CPUs, cores, or percentage of a CPU in the partition. Ultimately, like electricity or water, extremely fine-grained metering will be deployed that counts actual CPU cycles allocated to a running workload.

■ **Storage**—Utility disk and tape storage has been available since the turn of the millennium from pure-play storage service providers such as Arsenal Digital as well as hosting providers such as USi. Capacity-on-demand models also exist from hardware vendors. Overbooking is supported by virtualization and "thin provisioning," pioneered by 3PAR and now implemented by vendors such as EMC, HP, Hitachi Data Systems and Sun. With thin provisioning, the amount of storage used on disk is based on how much real data is actually being stored, rather than the size of an allocation. Storage virtualization also plays a role in disguising actual enterprise array characteristics while creating virtual volumes of any size.

■ **Middleware**—AT&T and BEA have jointly created what appears to be the industry's first and possibly only utility middleware technology architecture and pricing scheme. Covering BEA WebLogic and AquaLogic middleware offered in AT&T hosting facilities, this utility pricing scheme enables two separate approaches. In one, CPU utilization for the month is averaged, and pricing is based on percentage points of average utilization. In the other model, pricing is based on duration of instance activation down to granularity as fine as minutes. In the first model, a customer would pay less if running at, say, 16 percent utilization for the month than at 17 percent utilization. In the duration model, a customer would pay less running WebLogic Server for, say 16 minutes, than running it for 17 minutes.

■ **Applications**—Software as a Service (SaaS) is a huge focus today, thanks to providers such as Microsoft, Google, Yahoo! and SalesForce.com. Besides pure-play providers, there are SaaS aggregators/marketplaces and platform/technology providers such as JamCracker and Sphera (recently acquired by SWsoft). Some providers, such as USi, play in more than one arena via hosted managed messaging in addition to application management. And providers of mash-up technology, such as IBM with their QEDWiki mashup maker, enable user-defined services.

■ **Desktops**—Utility desktops are now possible due to the emergence of WAN-enabled remote desktops, wherein a pool of "virtual desktops" can be allocated to numerous users. Each user has a hardware thin client, such as that available from Wyse, which, rather than running a PC operating system runs a lightweight environment that mostly processes display graphics, two-way audio, and keyboard and mouse inputs.

In one architecture, exemplified by HP's Consolidated Client Infrastructure or ClearCube's PC Blade, remote processors have a one-to-one connection with each user. These processors may have WAN graphics acceleration via technologies such as those available from Teradici. In a more virtualized approach with potentially greater payoff, rack-mount or blade servers such as HP Proliants or an IBM BladeCenter are virtualized with software such as VMware Virtual Desktop Infrastructure. Each processor then serves from a few to several dozen desktops, each of which may be dynamically allocated at time of use by a connection broker. And these hardware architectures are now being offered as a service through companies such as Desktone.

Although similar approaches never quite caught on, a new generation of technology coupled with utility services enhances security, extends distance to the processor complex, reduces capital expenditures, and simplifies desktop management—and is now getting traction, with some customer deployments reaching tens of thousands of desktops. They complement related

FIGURE 1a Simple Demand Model

For many situations, dedicated and utility models are best combined

approaches, such as Sun's Global Desktop and Citrix Presentation Server.

The Utility Value Paradox

One of the counterintuitive observations about utilities is that even though they may cost more, they still save money.

For example, consider a car you are financing at $300 per month. That works out to $10 per day. It would be hard to find a rental car company that would rent you that same make and model for $10 per day. One might expect to pay $30 to $50 instead. So, on a "car per day" basis, customers pay 3 to 5 times as much for a utility.

The reason it makes sense to pay extra for a utility is that a dedicated asset costs money (depreciation, operating expenses, managed services charges and/or leases), whether it is in use or not. A utility resource may cost more when used, but costs less, typically zero, when not used.

Let's say that a utility is priced at twice the average rate of a dedicated asset when used, but there is no charge when it is not used. If it is used less than 50 percent of the time, the premium for use the minority of the time is more than made up for by the savings the majority of the time.

In fact, this breakeven point is easy to calculate. Let the average utilization be A, and the peak utilization be P. For fixed resources, assume that the base rate per time period is B, and let the premium for the utility be U, such that the premium utility rate is therefore B * U. Then having an environment engineered for a peak that uses the lower flat base rate would have a cost of P * B. However, a utility environment would charge the higher rate of B * U, but for fewer resources, on average only A, for a total cost of A * (B * U). The utility is financially attractive when A * B * U < P * B. Eliminating the B from both sides, this happens when U < (P / A), that is, when the utility premium is less than the Peak-to-Average rate.

Hybrids Can Be Optimal

In fact, rather than either/or, a hybrid comprising various pricing strategies may be optimal in many cases. For example, a dedicated environment or portion of an environment to serve the baseline demand and/or for minimal business continuity can take advantage of the typically lower non-utility flat rate. However, the remainder of the (spiky) demand is best served by a utility model.

Consider an application requiring 100 servers at peak. This application is run from 9–5, Monday through Friday. Within those timeframes, utilization is assumed to be uniformly distributed. A quick calculation shows how low the average utilization is: 40 hours a week out of 168 possible hours in each week is only 23.8 percent. But since the utilization during these periods of use is assumed to be uniformly distributed, the expected utilization is half of that, or only 11.9 percent.

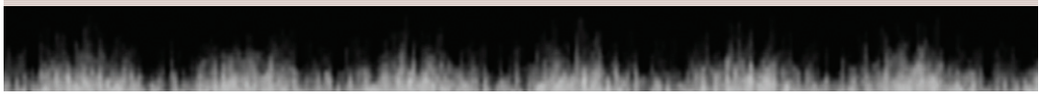
A model of this application is shown in Figure 1a from ComplexModels.com. It shows a snapshot of this application running over the course of a year. A 365-day year has 8,760 hours in it. If the allocated cost for a server is $2 an hour, this will cost $2/server/hour * 8,760 hours/year * 100 servers = $1.752 million.

However, even with a premium of 50 percent for utility servers, say $3 an hour, if they are used to handle the average requirement of only 11.9 servers, there is a savings of more than 80 percent, since $3/server/hour * 8,760 hours/year * 11.9 servers = $312,732. In this simple model, demand is highly spiky, and a pure utility environment is dramatically less expensive than a pure fixed environment or even a hybrid environment.

Figure 1b shows a more complex and realistic environment, from a simulation run at ComplexModels.com, composed of 100 servers of uniformly distributed 24/7 demand, 100 servers of uniformly distributed 9–5 demand, and 100 servers of uniformly distributed monthly cyclical demand. In any given hour during the year (for this simulation run), demand drops as low as .58 of a server, but also gets as high as 266.02 servers. Average utilization is 86.61 servers.

A pure dedicated flat-rate environment engineered to peak capacity would cost $2.00/server/hour * 8,760 hours * 300 servers, or $5,256,000. Even if the capacity planner guessed the peak of 266.02 correctly, it would still cost $4,660,761 for the environment. Using pure utility resources, the price would drop to $2,275,980. As it turns out however, by recognizing that this utilization curve has a substantial baseline processing demand on top of which there are spiky peaks, a hybrid solution is optimal. In this case, a

FIGURE 1b More Realistic Environment



Allocation of resources must be dynamic to match demand

minimal cost of $1,942,202 is reached via a combination of 63 fixed servers with the remaining capacity met on demand using utility servers.

Service Provider View Of Dedicated vs. Utility Economics

Service providers also benefit from utilities. For the same resources, service providers can charge a premium, due to the economic customer value proposition of flexible resources.

Of course, whether the utility environment actually generates profitable revenue depends on the extent to which the provider can maximize capacity utilization. So, for the service provider, the revenue enhancement factor from utility pricing is U * C, where U, again, is the utility premium and C is the average capacity utilization. For example, if resources offered on a utility basis can be priced at double that of fixed resources, but utilization of the environment is below 50 percent, it would have been better to sell that capacity the old-fashioned way: on a flat rate, dedicated basis.

Also, in real environments, many other factors come into play, e.g., the additional cost of utility-enabling infrastructure, overbooking or SLA penalties, dynamic pricing to maximize yield, etc. However with sound capacity management and provisioning, both the service provider and the customer gain.

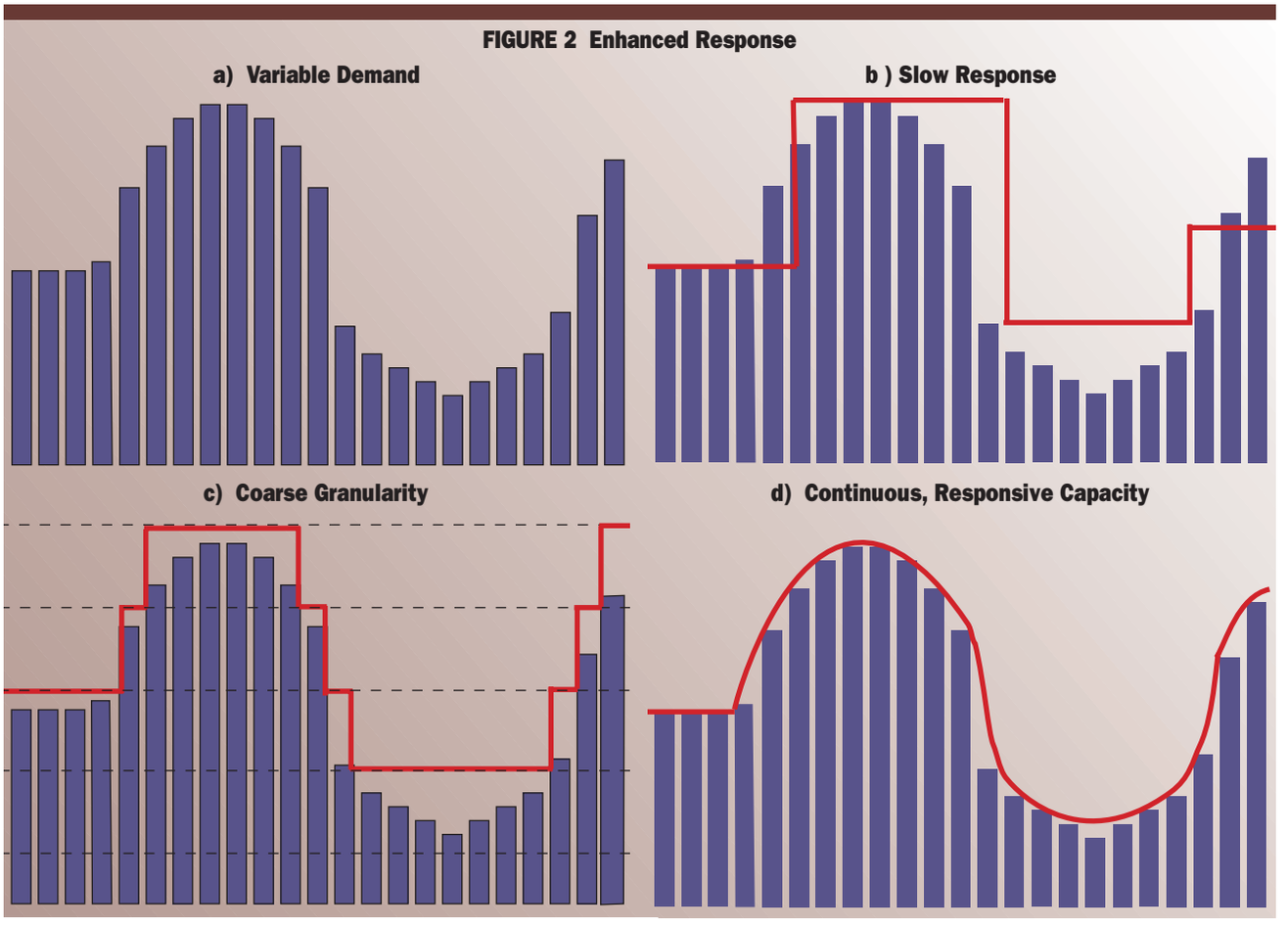
The Service Provider Advantage

Service providers have an inherent and fundamental economic advantage in delivering utility models, since they time-division- or space-division-multiplex multiple customers into a shared, multi-user, multi-application environment. This sets up a natural three-tier value chain structure for utility delivery: hardware/software vendors, service providers and customers.

Although many hardware vendors offer customer premises-based "utility" products, with names such as "Instant Capacity on Demand" or "On-Off Capacity on Demand," these models can only provide limited value to customers because they don't multiplex demand across customers, applications, locations—or at least not to the same extent that a service provider can. This is why rental car service providers are different than car manufacturers, and hospitality service providers are different than concrete, plumbing fixture and mattress manufacturers.

Hotels can be booked solid 7 days a week, by booking conventions and individual business travelers during the week and weddings, vacationers, etc., during the weekend. Similarly, a networked IT service provider can get high utilization by running employee and customer applications Monday through Friday during the day, then running consumer, ecommerce, gaming and batch transactions at night and on weekends.

FIGURE 2 Enhanced Response



Utility Evolution I: Dynamically Responsive Capacity

Figure 2 illustrates the end-state objective for a simple resource-based utility. Figure 2a shows an example variable-demand curve: flat for a while, then with increasing demand, then decreasing demand, then another growth period. Figure 2b shows what happens when allocation and de-allocation of resources is too slow. There are periods of insufficient capacity, where demand will either not be met or have unacceptably long response times, and then periods of overcapacity when the system doesn't shed excess capacity quickly enough. Even with excellent dynamic response, Figure 2c shows what happens when the capacity increments are too coarse. The system instantly adjusts its capacity, but always has too much capacity. If the demand is continuously variable, the expected amount of overcapacity is 1/2 the capacity increment. Figure 2d shows the desired end-state, a system with very fast response times and very fine-grained, ideally continuous, capacity increments.

Speeding up response time is the responsibility of the utility management system. Ideally, it not only can respond reactively in extremely short time frames, but can be proactive and predictive. Decreasing the granularity of capacity is the responsibility of a virtualization layer. To get granularity below one server, one enterprise array, or one OC-12 requires virtualization with fine-grained partition sizes.

For example, one server product from a large vendor provides 10 percent micro-slices of a CPU using a firmware hypervisor. A software-based virtualization product provides single percentage points of partitioning. Ultimately, server virtualization has, as its end-state, granularity of individual CPU cycles, and utilities based on that will depend on either a virtual server kernel to count cycles or a workload management agent.

Similarly, in the WAN, we are seeing finer granularity. For example, provisioning options have traditionally been restricted to OC-3c, OC-12c, or OC-48c. Now, Virtual Concatenation (VCAT) enables finer-grained STS-1 increments. Beyond that, one equipment startup claims to be able to adjust SONET bandwidth allocations within one 72,000th of a second, and in granularity measured in frame-sized increments.

Utility Evolution II: Geographic Dispersion

Complementing the evolution in pricing models is increased geographic spread:

1. Single Location—A single location is the minimal baseline for processing.

2. Secondary Location—Evolving to two locations has a compelling business continuity advantage. If a single site has availability of 99 percent then two sites provide 99.99 percent (four nines) availability, assuming independent root causes of unavailability, since the total system availability is now 1 – ((1-.99) * (1–.99)). From a network architecture perspective, the analog of dual sites is diverse routing, and dual independent access (dual laterals and dual risers).

3. Tertiary Location—Three sites have the advantage of achieving a five 9s or higher availability architecture (depending on individual site availabilities and correlations). In general, more sites and greater geographical separation protect better against physical disasters.

4. Dispersed Locations—In addition to business continuity concerns, latency for real-time and interactive applications is helped by geographical dispersion. Networks have become lower-latency due to longer-haul optics, smarter routing, fewer hops, WAN acceleration, and faster ASICs, and also higher bandwidth due to next generation 40-Gbps wireline optics and evolving wireless capabilities such as 3GPP's HSPA (High Speed Packet Access) and ultimately LTE (Long Term Evolution). However, physical propagation delays will always exist, driving a dispersion requirement for interactive and real-time services.

5. Connected Grid—A connected grid that enables resource-sharing across locations—creating a single logical pool of resources from formerly unconnected dispersed pools—can create substantial value. Table 1 (p. 42) shows 10 locations over a 24-hour period. Each has randomly generated, uniformly distributed demand ranging from 0 to 100 servers worth of capacity.

As can be seen, sometimes the demand is close to zero, but virtually all servers come close to requiring peak capacity at some point during the day. Consequently, without sharing across locations, the actual capacity required to meet peak demand would be 954 servers. However, if all the servers are part of the same logically centralized pool, the peak aggregate demand never rises above 658.19 servers (in this sample run).

Conceptually then, a reduction in capacity from 954 to 659, or 31 percent would be possible. In practice, one would not engineer capacity that tightly, and this is only one set of data, but it gives a sense of possibilities. And, in general, the more locations being connected, the smoother the aggregate curve; and the larger the set of samples across more and more hours, the greater the likelihood that any particular site will experience a time slot with demand arbitrarily close to the upper bound of the distribution.

6. Inter-Provider Grid—Once intra-provider connections occur, a next step is inter-provider resource pooling. If a hotel is booked up, the first thing they will try to do is send you to another local property in the chain. Failing that, however, they will send you to a competing chain. In the same way, if networked utility computing capacity from a given provider is sold out, it may be preferable to send the demand to a competitor rather than provide poor service or pay out on an SLA.

Using multiple locations enhances availability

TABLE 1 The Value of Pooling Dispersed Resources

	Loc 1	Loc 2	Loc 3	Loc 4	Loc 5	Loc 6	Loc 7	Loc 8	Loc 9	Loc 10	Aggregate Demand
Time 1	74.55	68.10	15.18	32.82	23.99	30.75	82.58	61.31	28.19	9.77	427.24
Time 2	5.96	2.64	40.99	36.33	36.05	43.14	12.60	6.19	61.21	26.26	271.38
Time 3	47.27	98.00	90.47	17.92	82.00	71.68	32.44	47.25	54.79	30.38	572.20
Time 4	98.35	7.35	86.79	28.78	16.41	53.07	89.13	71.73	33.02	71.59	556.22
Time 5	27.91	10.80	64.47	26.26	6.20	1.66	24.90	96.89	61.34	35.15	355.58
Time 6	82.85	63.88	57.24	25.90	63.23	37.43	70.46	79.48	96.53	3.18	580.18
Time 7	93.27	74.32	60.86	37.29	25.68	83.08	63.84	54.55	65.36	79.70	637.95
Time 8	0.79	52.61	32.27	40.31	16.17	27.60	66.18	44.30	38.37	57.54	376.13
Time 9	74.75	12.17	0.32	71.44	12.37	55.95	86.89	4.26	49.27	36.67	404.10
Time 10	29.75	58.73	36.96	85.87	4.13	5.91	61.14	93.01	41.94	38.14	455.60
Time 11	92.46	45.00	43.50	18.11	28.55	27.56	25.80	40.14	60.13	53.83	435.09
Time 12	59.69	9.04	81.36	34.57	50.14	22.47	64.56	44.23	94.70	51.84	512.59
Time 13	59.56	49.88	73.99	78.72	88.48	28.90	78.78	72.35	51.84	23.97	606.46
Time 14	73.94	71.66	77.28	14.19	51.65	33.30	30.20	25.36	48.62	9.90	436.10
Time 15	95.20	64.72	6.04	99.72	58.37	25.08	45.10	9.40	90.67	55.84	550.14
Time 16	79.84	75.26	45.72	97.07	53.25	21.60	55.68	55.14	38.60	8.31	530.46
Time 17	80.68	92.99	1.89	61.60	34.02	2.45	92.76	85.53	50.62	19.29	521.83
Time 18	36.09	17.40	32.71	18.22	22.40	2.09	52.09	19.72	94.89	69.21	364.82
Time 19	55.24	81.18	9.99	58.40	26.21	48.40	40.35	73.23	55.30	30.06	478.35
Time 20	93.11	65.60	90.34	62.60	44.13	69.09	70.04	7.76	76.68	22.64	601.99
Time 21	96.03	97.61	66.93	49.77	66.24	91.79	32.63	55.54	87.75	3.92	648.20
Time 22	45.49	10.16	65.15	18.28	44.82	42.74	38.41	88.30	90.36	51.99	495.69
Time 23	37.15	62.06	5.30	52.83	80.29	83.39	98.20	59.43	84.07	95.47	658.19
Time 24	89.13	20.83	25.07	59.81	4.48	12.25	2.93	53.72	47.57	57.71	373.50
Peak cap req./loc	98.35	98.00	90.47	99.72	88.48	91.79	98.20	96.89	96.53	95.47	
Total capacity required w/o sharing						953.90	Peak Capacity Required Across Locations				658.19

7. Co-Generation—Not only may capacity be shared between service providers, but also between customers and service providers. In the world of computing, this is made more difficult due to application security requirements, but technologies are emerging for securely running jobs in a computational sandbox immune from eavesdropping by the owner of the processor.

8. Aggregator/Resellers—Once jobs may be run anywhere, capacity aggregators, resellers, agents and other intermediaries are likely to arise.

9. Global Optimization—Given the numerous options on where to run compute jobs, and given the dynamically varying costs for compute capacity and network transmission capacity to deliver data to compute nodes or results from nodes to users, global optimization becomes the final challenge. In a complex, dynamically variable environment, it may be very difficult to optimally reduce total cost, including not only compute, storage and network costs, but costs associated with latency, job delays or potential disaster scenarios in the event a node fails just as a long-running job was about to complete. But heuristics may be able to deliver "good enough" scheduling and allocation.

The Consolidation Conundrum

Many companies today are consolidating datacenters to reduce operating and personnel costs. However, the logic above argues for greater geographic dispersion.

Which approach is correct? The answer is: both. Consolidating datacenters for back-office tasks and batch processing makes sense. However, increasing dispersion for real-time and interactive processing also makes sense. An architecture composed of capabilities appropriately deployed in the core data processing center, at the network edge, and also in intelligent devices appears to be optimal for most tasks.

Similarly, as described above in the analysis of the demand shown in Figure 1B, a hybrid of utility and flat-rate resources may be optimal, in the same way that owning, financing, or leasing a car at one's primary residence but renting a car or using a taxi when traveling is the lowest cost.

This leads to a hybrid strategy, combining the best of both worlds for the typical complex set of applications an enterprise needs to run: on the one hand, legacy, internal, predictable, core, throughput-intensive, consolidated, owned/dedicated, flat-rate tasks may be best run in a customer datacen-

ter. On the other hand, Web-oriented, customer-facing, spiky, real-time/interactive, latency-sensitive tasks may be best run in a dispersed, shared/multi-tenant utility environment.

Utility Evolution III: Pricing

As defined in the introduction, the key aspects of utility for our purposes are flexible use combined with variable pricing. Understanding the evolution of pricing is key to shaping customer value propositions as well as predicting trends in pricing evolution and market opportunities.

1. Dedicated Resource, Flat Rate—The baseline model is a dedicated resource with flat-rate pricing. Such a flat rate is based on straight-line depreciation or a fixed monthly recurring charge for a lease or managed service.

2. Dedicated Resource, Non-Usage Sensitive Variable Pricing—Non-flat-rate pricing does not necessarily imply usage-sensitive pricing. For example, pricing could decline as technology ages and newer technology arrives at different price points, following Moore's Law or Gilder's Law.

3. Dedicated Resource, Usage-Sensitive Pricing—This is the first stage that utility pricing occurs. For example, a dedicated CPU may have variable pricing based on average CPU utilization, or a dedicated network pipe may have variable pricing based on average bandwidth utilization.

4. Dynamically Allocated Resource, Usage-Sensitive Pricing—This is the model that we usually think of for a utility-priced service such as a rental car, a cab or a hotel room. A virtualized entity (king size hotel room) is mapped to a dynamically allocated physical entity (room 327), which is then billed based on usage (3 nights). Similarly, pricing storage based on Gigabyte-months or CPUs on core-minutes or networks on STS-1-minutes provides true usage-sensitive pricing for these dynamically allocated resources.

5. Differential Pricing—Similar environments can have substantially different cost structures. This can be due to different ages, prices, and depreciation of physical equipment resources; different operations costs such as floor space, power, cooling and system administration; different acquisition costs due to shipping, installation, volume purchasing contracts, vendor pricing and promotions, and myriad other factors. Consequently, similar resource pools may be priced differently. Of course, a service provider may also decide to make subtle differences transparent.

6. Dynamic Pricing—Resource-based utilities of perishable capacity, whether airplane seats or CPU cycles, ultimately evolve to dynamic pricing. This is an inevitable consequence of free markets as well as intentional yield management, which maximizes resource revenue and therefore profitability by raising prices when capacity is scarce and lowering prices when excess capacity exists. Such variation may be seasonal, e.g., resort pricing is higher during holiday periods and lower during hurricane season, and similarly, capacity to support on-line retailing may be higher towards the end of the year. It also may follow daily cycles, encouraging batch jobs to run overnight when capacity is not in use for interactive applications. Ultimately, a real-time market will evolve with prices varying just as quickly as in the stock market as demand ebbs and flows.

7. Derivatives—Dynamic pricing introduces pricing risk into markets. To hedge that risk, derivatives such as long-term supply contracts, futures, options and exchanges enter a market. These exist today in equities (e.g., stock exchanges and options) and commodities (e.g., pork bellies). To the extent that CPU cycles become commoditized, dynamically priced and tradable, one can predict derivative instruments will become commonplace.

Utility Evolution IV: Control

With all of the complexity inherent in tomorrow's hybrid fixed-resource and utility networked IT architectures, the final requirement is for enhanced control. The evolution of control starts at awareness but ultimately requires global end-to-end optimization.

Various vendor products and capabilities currently in the market provide intelligent correlation for accelerated root cause analysis of troubles, support SLA objective compliance, provide some degree of policy-based orchestration based on Boolean rules, recommend environment changes, or otherwise support the evolution below. However, neither the management software industry nor the service provider industry yet fully provides end-to-end optimization for complex distributed multi-layer architectures with differential and dynamically priced utility elements:

1. Awareness—Whether through auto-discovery or manual element and topology entry, awareness of the elements in the environment and their relationships is a key first step.

2. Monitoring—Monitoring status and troubles is the next step providing minimal service assurance and enabling subsequent capabilities.

3. Management—Patch and configuration management, break/fix, and related capabilities are the next traditional evolution of control, even for non-utility environments.

4. Orchestration—With all of the virtualization and flexibility inherent in utility environments, a control layer must be able to orchestrate this flexibility. This may be driven by customer policies and supported by workflow engines or scripts to reconfigure elements correctly and without causing outages.

5. Local Optimization—Several types of optimization can be expected in emerging environments. One is performance tuning, for example, increasing or decreasing virtual server partition sizes, adjusting database connections, changing virtual to physical server mapping, and so forth to meet performance SLAs or overcome failures or

Enhanced control is required to make it all work

Pricing models tend to swing from flat-rate to usage-based and back again

outages. Another is cost optimization, especially if differential or dynamic pricing exists.

6. Global Optimization—There are many potential costs in a complex environment including latency, resource price variation, costs of job delays, solutions to move data closer to where it is processed or selecting processing nodes closer to data, etc. Determining the globally optimal solution and updating it continuously in real time is challenging, and may even be computationally infeasible depending on the complexity of the environment and desired response time.

The Flat-Rate vs. Utility Pendulum

Flat rate and utility models each offer benefits. As shown in Figure 3, products or services may start out with simple, flat-rate pricing. This flat rate pricing is, ideally, based on making a sustainable, fair profit on the goods or services being sold. Because there is likely to be a normal (bell-shaped) or other variable usage distribution, inevitably, some infrequent users will be relatively overpaying, and frequent users will be relatively underpaying.

One competitor will introduce usage-sensitive pricing, and the infrequent users, sensing a bargain, will defect to this competitor. The average usage of the flat-rate services then increases, as infrequent users defect to usage-sensitive services. The profitability of the flat-rate service providers decreases, if rates stay constant but average usage increases. So, flat-rate service prices then increase, causing defection of a new tranche of low-usage subscribers.

In a second phase, more and more usage-sensitive providers enter the market, and more and more users shift from flat-rate services to usage-sensitive services. Now, the competition is not so much between flat-rate and usage-sensitive providers, but among usage-sensitive providers. In an attempt to innovate, they now battle among each other, offering various incentives, bundles, promotions and discounts, leading to a phase we can call complexification.

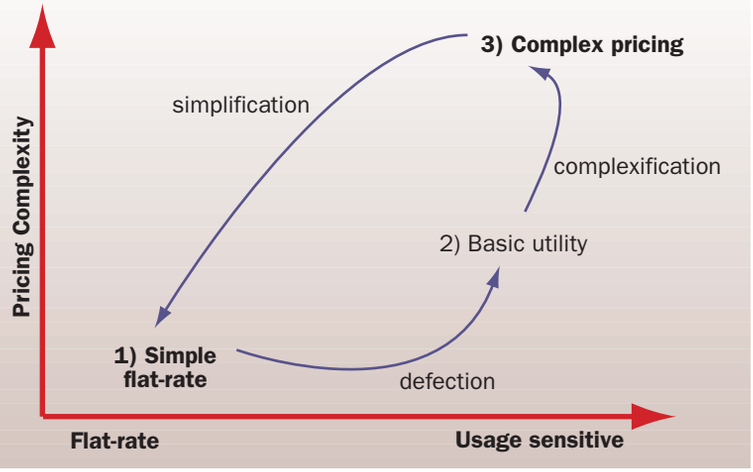
Finally, in a third phase, an innovator promises a return to simpler days, through simple, flat-rate pricing. As the hidden costs of optimizing against and arbitraging complex rate plans get factored in to total cost of service, users rebel against complexity and, in a period of simplification, return to flat-rate providers.

As we have seen, a hybrid of flat-rate and utility models is mathematically optimal for both customers and service providers for typical real-world demand scenarios. Therefore, the challenge is to ensure that service provider offers don't drive through this complexification phase, but let all participants in the market benefit from these inherent advantages.

Summary

Networked utility computing environments are becoming more pervasive and more capable. Enabled by today's dispersed content delivery and application hosting environments, coupled with global networks providing bandwidth on demand, quality of service flexibility, route control, and even packet control, they are already creating solid return on investment for customers as well as viable new business models for service providers□

FIGURE 3 Utility Market Evolution



Companies Mentioned In This Article

3Leaf Systems (www.3leafsystems.com)
3PAR (www.3par.com)
AT&T (www.att.com)
Arsenal Digital (www.arsenaldigital.com)
BEA (www.bea.com)
Cisco (www.cisco.com)
Citrix (www.citrix.com)
Clearcube (www.clearcube.com)
Desktone (www.desktone.com)
EMC (www.emc.com)
Google (www.google.com)
Hitachi Data Systems (www.hds.com)
HP (www.hp.com)
IBM (www.ibm.com)
Jamcracker (www.jamcracker.com)
Microsoft (www.microsoft.com)
Nortel (www.nortel.com)
Salesforce.com (www.salesforce.com)
Sun (www.sun.com)
SWsoft/Sphera (www.swsoft.com)
Teradici (www.teradici.com)
USi (www.usi.com)
VMware (www.vmware.com)
Wyse (www.wyse.com)
Yahoo! (www.yahoo.com)