Compare three options for delivering your cloud applications

Learn the best uses for bare metal, VM, and container-based delivery

Chuck Calio

August 20, 2015

The growth of cloud computing is changing how applications and other IT services are delivered to users. Learn how to choose the most effective delivery solution for your needs, whether through bare metal, virtual machines (VMs), or container technologies.

Most cloud-based services available today are delivered through either bare metal or virtual machines (VMs), however container technologies offer a powerful third choice for cloud services application delivery.

With the emergence of cloud computing, many organizations established cloud first policies that specify cloud, rather than traditional on-premises, as the priority choice for new application delivery. Three options for application delivery are popular:

• **Bare metal environments**: Applications are installed, run, and delivered from a base of dedicated cloud infrastructures and OS platforms. A key attribute of bare metal environments is the use of dedicated and specific hardware and operating system configurations.

• **Virtual machines (VMs)**: Applications are installed, packaged, and run under the control of a hypervisor that virtualizes the hardware environment and also includes dedicated copies of an operating system and other required software. VM environments share hardware resources.

• **Containers**: Only essential parts of the application (cloud services) and its dependencies are included. Multiple containers can run within a single instance of an operating system, which is either Linux® or Microsoft® Windows®. Docker is an example of a container runtime engine, along with a registry of popular Docker containers called DockerHub. Containers are used extensively by many next-generation cloud Platform as a Service (PaaS) developers. IBM® Bluemix™ is a PaaS cloud platform where you can use IBM Containers to run Docker containers.

Most popular cloud computing companies offer various permutations and combinations of bare-metal Infrastructure as a Service (IaaS) cloud application delivery options (for example, IBM SoftLayer). In addition, a large percentage of today’s cloud implementations take advantage of
the efficiencies and effectiveness gained by using hardware virtualization (VMs). The third option, container technology, is open source and makes it less costly.

Knowing the pros and cons of the most popular cloud application delivery methods helps you make the best choice. While most cloud-based services are delivered through bare metal or by using virtualization (and VMs), a popular third choice for application delivery is Docker containers. A combination of bare metal, VMs, and containers made available in a unified cloud platform might be the best of all possible options.

A few basics on cloud application delivery choices

One of the first areas to explore is mapping the required cloud application to a specific and optimal hardware, operating system, and hypervisor.

- **Bare-metal cloud environments** can be built across a broad range of combinations of hardware, operating systems, middleware, and applications that are optimized to run with dedicated hardware resources. For example, you can build bare-metal environments with IBM AIX®, IBM i, Linux on Power, IBM z Systems™, and x86 Linux and Windows operating systems. Bare metal environments can also support a broad range of internal and external storage subsystems and networking gear.

- **Virtualized (VM-based)** applications support environments that are tied to a specific OS, hypervisor, and hardware combination (for example, the Linux operating system and KVM hypervisor, or the Windows Operating system and VMware, or Windows and Hyper-V), and are often enabled to support highly shared and virtualized operating environments on specific hardware. With VMs, you have the flexibility to run different kernels or OS levels in different guests or VMs on the same physical hardware.

- **Containers** run on Linux (x86, Power, or z Systems) or Windows systems. IBM AIX also supports container technology. When using containers, many clients try to use the same kernel/Linux distribution as much as possible. You might want to investigate whether the container and Linux environment being targeted supports the internal and external disk subsystems that store the data to be accessed by the applications running in the Docker containers.

Key advantages of bare metal, VMs, and containers

Bare metal cloud environments can be built to support large number of OS/hardware combinations. Resources can be dedicated and tailored to an individual client or an application. Many people prefer bare metal configurations because they deliver consistent and often high levels of application performance due to their dedicated, optimized, and specialized focus and construction. Bare metal environments can also provide applications with full access to system resources, if required. In bare metal environments, you can deploy a specific application on optimized combinations of specialized hardware and operating systems versions. This deployment can be useful for applications that need access to specific hardware features (for example, graphics processor accelerators, Field Programmable Gate Array) and operating systems environments. In some cases, bare metal cloud environments can be the only option available to provide cloud applications with the required level of performance. Regulatory requirements can dictate that certain applications and data be on specific bare metal cloud environments.
VMs enable IT systems to share resources much more efficiently than dedicated or bare metal approaches, and can also provide a robust level of isolation across application OS and hypervisor pairings. VMs can optimize cloud applications to accommodate fluctuations in demand (elasticity), which is a powerful value proposition of cloud. VMs are enabled by hypervisors, which support a broad range of OS and hardware operating environments. It is relatively easy to refactor existing traditional monolithic software applications into virtual appliances that run in a VM, and then deploy the monolithic software as a cloud service or cloud pattern. In essence, VMs represent a fast path to cloud for monolithic software. Also, migrating VMs across systems is easy.

Containers are much smaller, faster to start, and require less resources to deploy when compared to VMs. Containers enable the highest level of cloud application density. Many cloud microservices developers prefer containers as the standard for the next generation of DevOps environments, and they are also optimized to run well on scale-out systems. A large supporting base ecosystem of containers (for example, Docker Hub) exists to accelerate developer productivity. The large ecosystem of cloud containers can accelerate agility and deployment for the next generation of cloud services.

**Potential disadvantages of each approach**

<table>
<thead>
<tr>
<th>IT job roles</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IT solution architects, systems integrators, and business partners</strong> should be skilled in these three cloud application delivery choices. These users should be able to clearly articulate how to connect together hybrid cloud environments and applications that span delivery across bare metal, VMs, and containers.</td>
</tr>
<tr>
<td><strong>IT operations professionals</strong> are probably already up to speed on the tools and techniques that are needed to run applications that are delivered in bare metal or VM-based cloud. However, they might need to learn how containers can be integrated into the organization’s cloud runtime environments.</td>
</tr>
</tbody>
</table>

Applications running in bare metal environments use dedicated hardware, which can quickly become inefficient and costly. Dedicated, bare metal hardware and software resources can be under-utilized and costly if not managed properly. Bare metal environments are typically static and it can be difficult to service erratic workloads that benefit from hardware elasticity. Installing bare metal applications is time consuming. Applications running on bare metal systems often are not fully optimized for cloud pay-as-you-go pricing, metering, billing, and elasticity. Workload migrations can be difficult in bare metal environments due to their customized nature.

VMs can take a long time to create and activate. Applications running in VMs tend to be built as distinct software stacks and have to be redone for minor (or major) changes, which can be time consuming. VMs take longer to start when compared to containers. Start time can become an issue. Hypervisor management software can be expensive compared to containers, which are free. Finally, VMs can introduce high levels of performance overhead, which can be unacceptable.
to performance or latency-sensitive applications. Hypervisor overhead can be high, and licensing costs that are associated with hypervisors can range from low (KVM) to high (VMware).

Containers run only on Linux or Windows. The business case for refactoring existing, older monolithic software to run in containers might not be viable. Many large, traditional IT shops and IT business people do not have a good understanding of containers. It is unclear how micro-services cloud software is licensed in a variety of fee-based container environments. Stateful services require access to the underlying host operating systems, and therefore are still hard to run in containers. Containers can also introduce some uncertainty around cloud security and compliance issues. If that is the case, users can look at running containers in VMs.

One final potential issue with delivering applications that use VMs or containers is that some software vendors might not create or optimize software licensing and support agreements that fully support virtualized (VMs) or container environments.

Choose the best option for your workload type

Deciding which option is best for you depends on your workload. This section describes the workloads that run well on each of the cloud computing delivery choices.

Bare metal workloads

- Applications that need the highest power you can give them in a single node
- Workloads with very high processor, I/O, or memory capacity requirements (for example, traditional large databases)
- Workloads that benefit from running on large, centralized scale-up systems
- Latency-sensitive services or applications
- Workloads that need predictable throughput and latency (great for guaranteed response times for a single workload; for example, performance or latency sensitive)
- Workloads that are reasonably steady and predictable. It is probably both: better for performance and cheaper to run on bare metal
- Legacy workloads that are not cloud enabled; Java or IBM WebSphere® multi-tenant environments
- Applications that need to or cannot tolerate virtualization or the noisy neighbor effect
- Applications with governance or regulatory restrictions that bind them to specific hardware

VM workloads

- Applications that benefit from the flexibility to easily provision, deploy, and de-provision from cloud services catalogs and cloud management software
- Legacy applications that need to be refactored on to a cloud deployment model
- Applications that benefit from pay-as-you-go and self-service
- Applications that do not require specific hardware bindings
- Workloads with low-to-moderate processor, I/O, or memory systems capacity requirements
- Workloads that benefit greatly from scale-out systems (lower TCO)
- Bursty workloads that can benefit from fast provisioning, deprovisioning, and virtualization resource efficiencies
• Applications that are optimized to be managed efficiently and effectively in a VM environment

Container workloads

• Standardized workloads using micro-services running in containers
• New Linux or Windows applications that are offered as cloud services
• Born-on-the-cloud micro-services developed for cloud containers and optimized for execution in containers
• Brand new applications that benefit from the large existing ecosystem of shared containers (for example, Docker Hub)
• Applications that require the fastest start and deprovisioning times
• Cloud services that can take advantage of large scale-out environments
• Clients already offering container-based cloud services

Conclusion

This article provided an introduction and guidance on the growing number of today's popular application delivery technologies associated with cloud computing. Most cloud-based services today are delivered through bare metal or virtualization (and VMs). A new, popular third choice, Docker containers, was described and recommended for the next generation of cloud services application delivery.
Related topics

- IBM Containers
- Explore IBM Bluemix
- Learn more about Docker
- Introduction to microservices
- Get an overview of IBM cloud technologies and how they all fit together.
- Learn how to install the Docker tools so that you can run containers on your local machine.
- Learn more about IBM Containers on the IBM Bluemix platform.
- "Hot Metal: How bare metal is making a comeback in cloud" examines the trend in enterprises toward bare metal implementations that's driven by performance, security, and cost concerns.
- "Docker container or VM? Canonical's LXD splits the difference" explores the new Linux Container Demon (LXD) project, which lets users work with Docker containers to deploy the functional equivalent of full-blown isolated Linux VMs, not merely individual containerized apps.
- In "Containers running on Bare-metal IaaS will Destroy the EC2 Virtualization Model of Cloud Computing," Zack Rosen explains how he believes that the VM-centric compute model will be replaced by a simpler, more efficient model of cloud computing: containers running on bare-metal infrastructure provisioned through an API.