4.3. Azure:

In 2008, Microsoft launched a Windows Azure platform to meet the challenges in cloud computing. This platform is built over Microsoft data centers. Figure 4.22 shows the overall architecture of Microsoft's cloud platform. The platform is divided into three major component platforms. Windows Azure offers a cloud platform built on Windows OS and based on Microsoft virtualization technology. Applications are installed on VMs deployed on the data-center servers. Azure manages all servers, storage, and network resources of the data center. On top of the infrastructure are the various services for building different cloud applications.

Live service Users can visit Microsoft Live applications and apply the data involved across multiple machines concurrently.

- .NET service This package supports application development on local hosts and execution on

cloud machines.

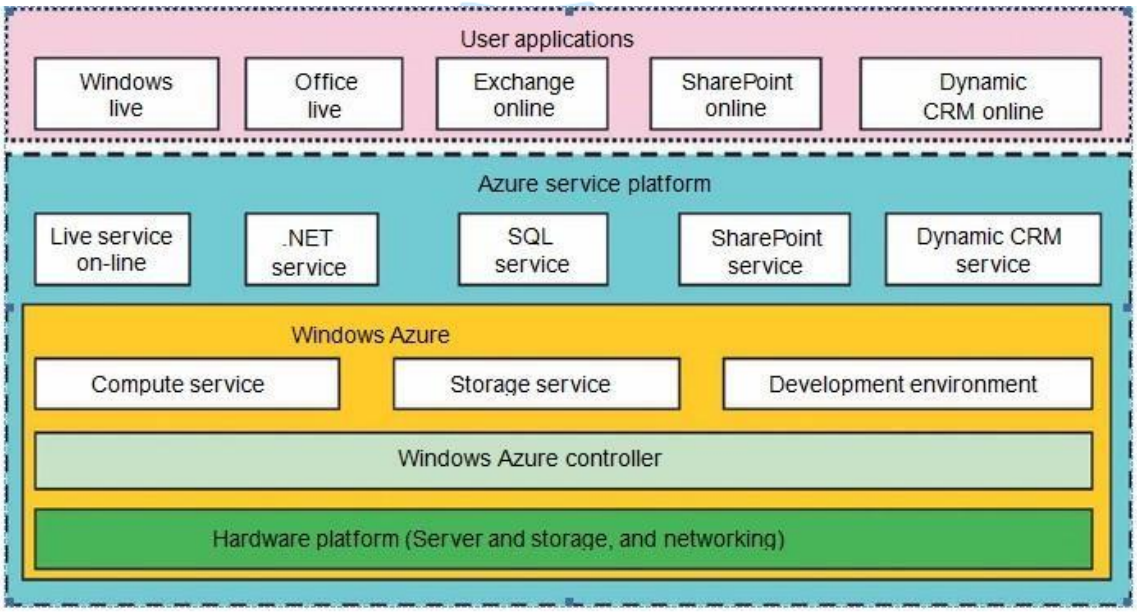


FIGURE 4.22

Microsoft Windows Azure platform for cloud computing.

SQL Azure This function makes it easier for users to visit and use the relational database associated with the SQL server in the cloud.

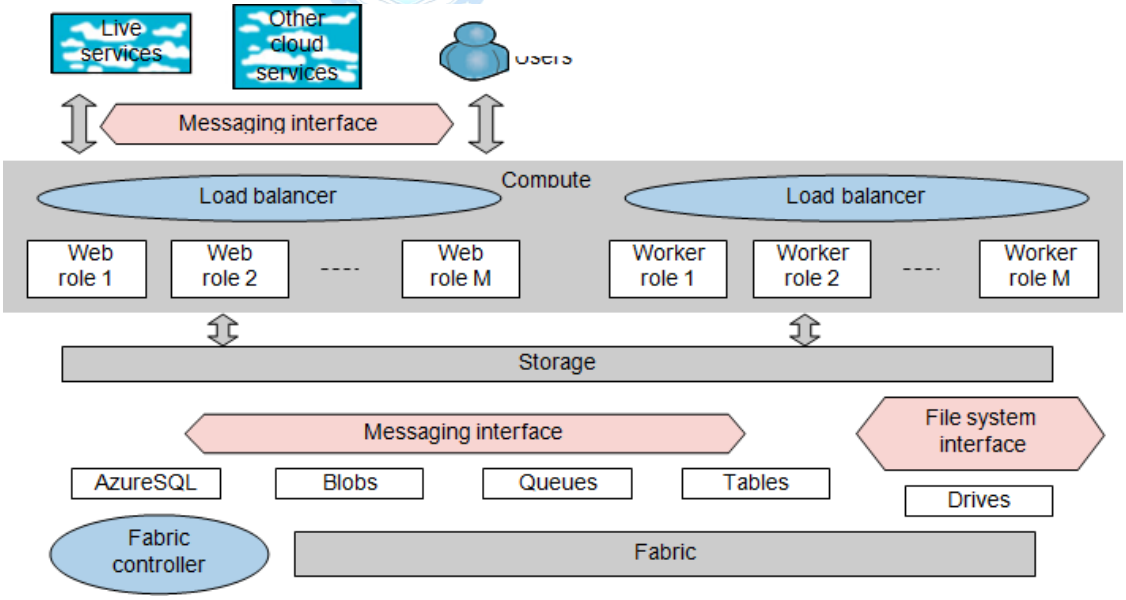
- SharePoint service This provides a scalable and manageable platform for users to develop their special business applications in upgraded web services.

- Dynamic CRM service This provides software developers a business platform in managing CRM applications in financing, marketing, and sales and promotions.

✓ All these cloud services in Azure can interact with traditional Microsoft software applications, such as Windows Live, Office Live, Exchange online, SharePoint online, and dynamic CRM online.

✓ The Azure platform applies the standard web communication protocols SOAP and REST. The Azure service applications allow users to integrate the cloud application with other platforms or third-party clouds.

✓ You can download the Azure development kit to run a local version of Azure. The powerful SDK allows Azure applications to be developed and debugged on the Windows hosts.



Features of the Azure cloud platform.

SQLAzure

Azure offers a very rich set of storage capabilities, as shown in Figure 6.25. All the storage modalities are accessed with REST interfaces except for the recently introduced Drives that are analogous to Amazon EBS discussed in above (AWS Methods), and offer

a file system interface as a durable NTFS volume backed by blob storage. The REST interfaces are automatically associated with URLs and all storage is replicated three times for fault tolerance and is guaranteed to be consistent in access.

The basic storage system is built from blobs which are analogous to S3 for Amazon. Blobs are arranged as a three-level hierarchy: Account → Containers → Page or Block Blobs.

Containers are analogous to directories in traditional file systems with the account acting as the root. The block blob is used for streaming data and each such blob is made up as a sequence of blocks of up to 4 MB each, while each block has a 64-byte ID.

Block blobs can be up to 200 GB in size. Page blobs are for random read/write access and consist of an array of pages with a maximum blob size of 1 TB. One can associate metadata with blobs as <name, value> pairs with up to 8 KB per blob.

Azure Tables

The Azure Table and Queue storage modes are aimed at much smaller data volumes. Queues provide reliable message delivery and are naturally used to support work spooling between web and worker roles. Queues consist of an unlimited number of messages which can be retrieved and processed at least once with an 8 KB limit on message size.

Azure supports PUT, GET, and DELETE message operations as well as CREATE and DELETE for queues. Each account can have any number of Azure tables which consist of rows called entities and columns called properties.

There is no limit to the number of entities in a table and the technology is designed to scale well to a large number of entities stored on distributed computers. All entities can have up to 255 general properties which are <name, type, value> triples.

Two extra properties, PartitionKey and RowKey, must be defined for each entity, but otherwise, there are no constraints on the names of properties—this table is very flexible! RowKey is designed to give each entity a unique label while PartitionKey is designed to be shared and entities with the same PartitionKey are stored next to each other; a good use of PartitionKey can speed up search performance. An entity can have,

at most, 1 MB storage; if you need large value sizes, just store a link to a blob store in the Table property value. ADO.NET and LINQ support table queries.