Cloud Computing Patterns

http://www.cloudcomputingpatterns.org

Native Cloud Applications

Two-Tier Cloud Application (290)
Presentation and business logic is bundled to one stateless tier that is easy to scale. It is separated from the data tier that is harder to scale.

Three-Tier Cloud Application (294)
Presentation, business logic, and data handling are realized in separate tiers to scale them according to their individual requirements.

Content Distribution Network (300)
Applications component instances and data handled by them are globally distributed to meet access performance requirements.

Hybrid Cloud Applications

Hybrid User Interface (304)
Varying workload from a user group interacting asynchronously with an application is handled in an elastic environment.

Hybrid Processing (308)
Processing functionality is hosted in an elastic cloud while the remainder of an application resides in a static environment.

Hybrid Data (311)
Data of varying size is hosted in an elastic cloud while the remainder of an application resides in a static environment.

Hybrid Backup (314)
Data is periodically extracted from an application to be archived in an elastic cloud for disaster recovery purposes.

Hybrid Backend (317)
Backend functionality (data-intensive processing and storage) is experiencing varying workloads and is hosted in an elastic cloud.

Hybrid Application Functions (320)
Some application functionality provided by user interfaces, processing, and data handling is hosted in an elastic cloud.

Hybrid Multimedia Web Application (323)
Website content is mainly served from a static environment. Multimedia files are served from an elastic high-performance environment.

Hybrid Development Environment (326)
A production runtime environment is replicated and mocked in an elastic environment where applications are developed and tested.

Application Workloads

Static Workload (26)
IT resources with an equal utilization over time experience static workload.

Periodic Workload (29)
IT resources with a peaking utilization at reoccurring time intervals experience periodic workload.

Once-in-a-lifetime Workload (33)
IT resources with an equal utilization over time disturbed by a strong peak occurring only once experience once-in-a-lifetime workload.

Unpredictable Workload (36)
IT resources with a random and unforeseeable utilization over time experience unpredictable workload.

Continuously Changing Workload (40)
IT resources with a utilization that grows or shrinks constantly over time experience continuously changing workload.

Cloud Service Models & Cloud Deployment Types

Infrastructure as a Service (IaaS) (45)
Physical and virtual hardware IT resources are shared between customers to enable self-service, rapid elasticity, and pay-per-use pricing.

Platform as a Service (PaaS) (49)
An application hosting environment is shared between customers to enable self-service, rapid elasticity, and pay-per-use pricing.

Software as a Service (SaaS) (55)
Human-usable application software is shared between customers to enable self-service, rapid elasticity, and pay-per-use pricing.

Public Cloud (62)
IT resources are provided as a service to a very large customer group in order to enable elastic use of a static resource pool.

Private Cloud (66)
IT resources are provided as a service exclusively to one customer in order to meet requirements on privacy, security, and trust.

Community Cloud (71)
IT resources are provided as a service to multiple customers trusting each other in order to enable collaborative elastic use of resources.

Hybrid Cloud (75)
Different clouds and static data centers are integrated to form a homogeneous hosting environment.

Fundamental Architecture Styles

Loose Coupling (156)
A broker encapsulates concerns of communication partner location, implementation platform, time of communication, and data format.

Distributed Application (160)
A cloud application divides provided functionality among multiple application components that can be scaled out independently.

Application Components

Stateful Component (168)
Multiple instances of a scaled-out application component synchronize their internal state to provide a unified behavior.

Stateless Component (171)
State is handled external of application components to ease scaling-out and to make the application more tolerant to component failures.

Multi-Component Image (206)
Virtual servers host multiple components that may not be active at all times to reduce provisioning and decommissioning operations.

User Interface Component (175)
Customizable user interfaces are accessed by humans. Application-internal interaction is realized asynchronously to ensure loose coupling.

Processing Component (180)
Processing functionality is handled by elastically-scaled components. Functionality is made configurable to support different requirements.

Batch Processing Component (185)
Requests are delayed until environmental conditions make their processing feasible.

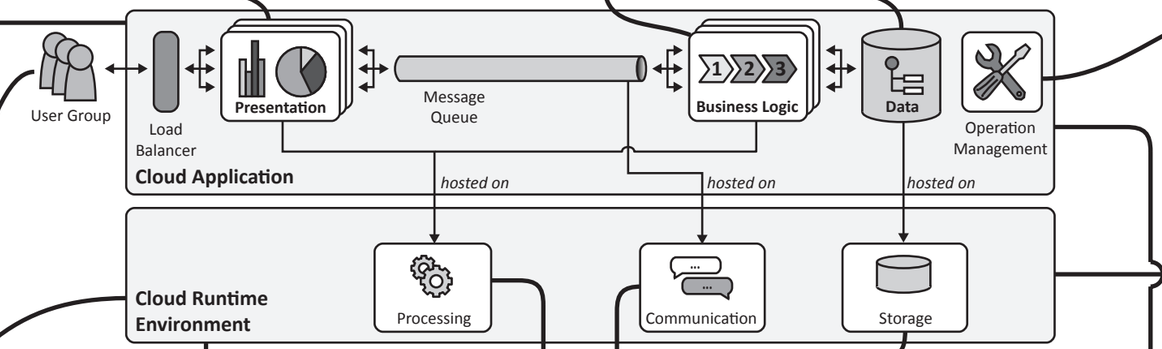
Timeout-based Message Processor (204)
Clients acknowledge message processing to ensure that messages are processed. If a message is not acknowledged it is processed again.

Transaction-based Processor (201)
Components receive messages or read data and process the obtained information under a transactional context to ensure processing.

Idempotent Processor (197)
Application functions detect duplicate messages and inconsistent data or are designed to be immune to these conditions.

Data Access Component (188)
Access to data is handled by components that isolate complexity, enable additional consistency, and ensure adjustability of data elements.

Data Abstractor (194)
Data is altered to inherently support eventually consistent data storage through the use of abstractions and approximations.



Cloud Environments

Elastic Infrastructure (87)
Hosting of virtual servers, disk storage, and configuration of network connectivity is offered via a self-service interface over a network.

Elastic Platform (91)
Middleware for the execution of applications, their communication, and data storage is offered via a self-service interface over a network.

Node-based Availability (95)
A cloud provider guarantees the availability of individual nodes, such as virtual servers, middleware, or hosted application components.

Environment-based Availability (98)
A cloud provider guarantees the availability of the environment hosting individual nodes, such as virtual servers or application components.

Processing Offerings

Hypervisor (101)
To enable the elasticity of clouds, the time required to provision and decommission servers is reduced through hardware virtualization.

Execution Environment (104)
To avoid duplicate implementation of functionality, applications are deployed to a hosting environment providing common functionality.

Map Reduce (106)
Large data sets to be processed are divided into smaller data chunks and distributed among processing application components.

Application Management

Provider Adapter (243)
Provider interfaces are encapsulated to separate concerns of interactions with the provider from application functionality.

Managed Configuration (247)
Application components use a centrally stored configuration to provide a unified behavior that can be adjusted simultaneously.

Elasticity Manager (250)
The utilization of IT resources on which an application is hosted is used to adjust the number of required application component instances.

Elastic Load Balancer (254)
The number of synchronous accesses is used to adjust the number of required application component instances.

Elastic Queue (257)
The number of accesses via messaging is used to adjust the number of required application component instances.

Watchdog (260)
Applications cope with failures automatically by monitoring and replacing faulty application component instances.

Elasticity Management Process (267)
Application component instances are added and removed automatically to cope with increasing or decreasing workload.

Feature Flag Management Process (271)
If the cloud cannot provide required resources in time, some application features are degraded in order to keep vital features operational.

Update Transition Process (275)
When a new application component version becomes available, running application components are updated seamlessly.

Standby Pooling Process (279)
Application component instances are kept on standby to increase provisioning speed and utilize billing time-slots efficiently.

Resiliency Management Process (283)
Application components are checked for failures and replaced automatically without human intervention.

Cloud Integration

Restricted Data Access Component (222)
Data provided to clients from different environments is adjusted based on access restrictions.

Message Mover (225)
Messages are moved automatically between different cloud providers to provide unified access to application components using messaging.

Application Component Proxy (228)
An application component is made available in an environment from where it cannot be accessed directly by deploying a proxy.

Compliant Data Replication (231)
Data is replicated among multiple environments. Data is automatically obfuscated and deleted to meet laws and security regulations.

Integration Provider (234)
Integration functionality such as messaging and shared data is hosted by a separate provider to enable integration of hosting environments.

Multi-Tenancy

Shared Component (210)
A component is accessed by multiple tenants to leverage economies of scale.

Tenant-isolated Component (214)
A component avoids influences between tenants regarding assured performance, available storage capacity, and accessibility.

Dedicated Component (218)
Components providing critical functionality are provided exclusively to tenants while still allowing other components to be shared.

Storage Offerings

Block Storage (110)
Centralized storage is integrated into servers as a local hard drive to enable access to this storage via the local file system.

Blob Storage (112)
Data is provided in form of large files that are made available in a file system-like fashion.

Relational Database (115)
Data is structured according to a schema that is enforced during data manipulation and enables expressive queries of handled data.

Key-Value Storage (107)
Semi-structured or unstructured data is stored with limited querying support but high-performance, availability, and flexibility.

Strict Consistency (123)
Data is stored at different locations to improve response time and failure resiliency while consistency of replicas is ensured at all times.

Eventual Consistency (126)
Performance and availability of data are increased by ensuring data consistency eventually and not at all times.

Communication Offerings

Virtual Networking (132)
Networking resources are virtualized to enable customers to configure networks, firewalls, and remote access using a self-service interface.

Message-oriented Middleware (136)
Asynchronous communication is made robust and flexible by hiding the complexity of addressing, routing, or data formats.

Exactly-once Delivery (141)
The messaging system ensures that each message is delivered exactly once by filtering possible message duplicates automatically.

At-least-once Delivery (144)
In case of failures that lead to message loss, messages are retransmitted to assure they are delivered at least once.

Transaction-based Delivery (146)
Clients retrieve messages under a transactional context to ensure that messages are received by a handling component.

Timeout-based Delivery (149)
Clients acknowledge message receptions to ensure that messages are received properly.

© The Authors 2014. All Rights Reserved. For more information visit: http://www.cloudcomputingpatterns.org