DHABALESWAR INSTITUTE OF POLYTECHNIC,ATHGARH,CUTTACK

E-COMMERCE LECTURE NOTES

6TH SEMESTER

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NAME OF THE FACULTY: MANAS RANJAN MOHANTY

Chapter-1

1.1 Electronic Commerce:

- Electronic commerce, commonly known as E-commerce is trading in products or services using computer networks, such as the Internet.
- Electronic commerce draws on technologies such as mobile commerce, electronic funds transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems, and automated data collection systems.
- Modern electronic commerce typically uses the World Wide Web for at least one part of the transaction's life cycle, although it may also use other technologies such as e-mail.

Definition of E-commerce:

Sharing business information, maintaining business relationships and conducting business transactions using computers connected to telecommunication network is called E-Commerce.

1.2 E-Commerce Categories:

1. Electronic Markets

Present a range of offerings available in a market segment so that the purchaser can compare the prices of the offerings and make a purchase decision.

Example: Airline Booking System

2. Electronic Data Interchange (EDI)

- It provides a standardized system
- Coding trade transactions
- Communicated from one computer to another without the need for printed orders and invoices & delays & errors in paper handling
- It is used by organizations that a make a large no. of regular transactions

Example: EDI is used in the large market chains for transactions with their suppliers

3. Internet Commerce

- It is use to advertise & make sales of wide range of goods & services.
- This application is for both business to business & business to consumer transactions.

Example: The purchase of goods that are then delivered by post or the booking of tickets that can be picked up by the clients when they arrive at the event.

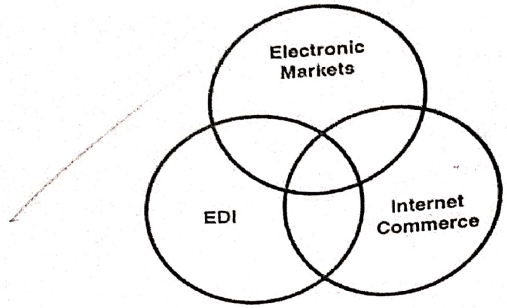


Fig. 1.1 The three categories of e-Commerce.

1.3 Advantages Of E-commerce:

- Buying/selling a variety of goods and services from one's home or business
- Anywhere, anytime transaction
- Can look for lowest cost for specific goods or service
- Businesses can reach out to worldwide clients - can establish business partnerships
- Order processing cost reduced
- Electronic funds transfer faster
- Supply chain management is simpler, faster, and cheaper using ecommerce
  - Can order from several vendors and monitor supplies.
  - Production schedule and inventory of an organization can be inspected by cooperating supplier who can in-turn schedule their work

1.4 Disadvantages Of E-commerce:

- Electronic data interchange using EDI is expensive for small businesses
- Security of internet is not very good - viruses, hacker attacks can paralise e-commerce
- Privacy of e-transactions is not guaranteed

- E-commerce de-personalises shopping

1.5 Threats of E-commerce:

- Hackers attempting to steal customer information or disrupt the site
- A server containing customer information is stolen.
- Imposters can mirror your ecommerce site to steal customer money
- Authorised administrators/users of an ecommerce website downloading hidden active content that attacks the ecommerce system.
- A disaffected employee disrupting the ecommerce system.
- It is also worth considering where potential threats to your ecommerce site might come from, as identifying potential threats will help you to protect your site. Consider:
- Who may want to access your ecommerce site to cause disruption or steal data; for example competitors, ex-employees, etc.
- What level of expertise a potential hacker may possess; if you are a small company that would not be likely to be considered a target for hackers then expensive, complex security may not be needed.

1.6 Features of E-Commerce:

- **Ubiquity**

  Internet/Web technology is The marketplace is extended beyond traditional available everywhere: at work, at home, and boundaries and is removed from a temporal and elsewhere via mobile devices, anytime. geographic location. "Marketspace" is created; shopping can take place anywhere. Customer convenience is enhanced, and shopping costs are reduced.

- **Global reach**

  The technology reaches Commerce is enabled across cultural and across national boundaries, around the earth. national boundaries seamlessly and without modification. "Marketspace" includes potentially billions of consumers and millions of businesses worldwide.

- **Universal standards**

  There is one set of There is one set of technical media standards technology standards, namely Internet across the globe.

- **Richness**

  Video, audio, and text messages Video, audio, and text marketing messages are are possible. integrated into a single marketing message and consuming experience.

- **Interactivity**

  The technology works Consumers are engaged in a dialog that through interaction with the user. dynamically adjusts the experience to the individual, and makes the consumer a co-participant in the process of delivering goods to the market.

- **Information density**

  The technology Information processing, storage, and reduces information costs and raises quality. communication costs drop dramatically, while currency, accuracy, and timeliness improve greatly. Information becomes plentiful, cheap, and accurate.

- **Personalization/Customization**

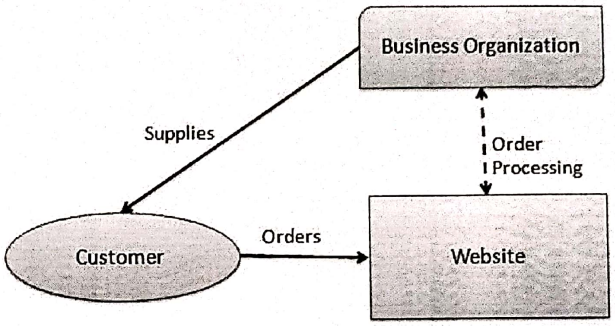
  The Personalization of marketing messages and technology allows personalized messages to customization of products and services are be delivered to individuals as well as groups. based on individual characteristics.

1.7 Business models of e-commerce:

There are mainly 4 types of business models based on transaction party.

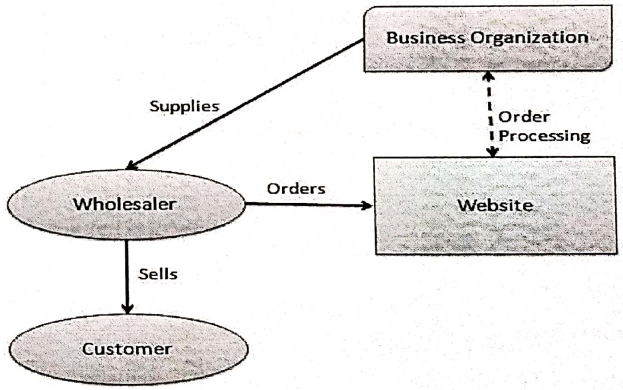
Business-to-Consumer (B2C)

In a Business-to-Consumer E-commerce environment, companies sell their online goods to consumers who are the end users of their products or services. Usually, B2C E-commerce web shops have an open access for any visitor, meaning that there is no need for a person to login in order to make any product related inquiry.



Business-to-Business (B2B)

In a Business-to-Business E-commerce environment, companies sell their online goods to other companies without being engaged in sales to consumers. In most B2B E-commerce environments entering the web shop will require a log in. B2B web shop usually contains customer-specific pricing, customer-specific assortments and customer-specific discounts.



Consumer-to-Business (C2B)

In a Consumer-to-Business E-commerce environment, consumers usually post their products or services online on which companies can post their bids. A consumer reviews the bids and selects the company that meets his price expectations.

UNIT-2

e-Business Integration (Patterns)

e-Business Integration occurs in as many forms as there are e-Businesses. At first glance, integration problems and the corresponding solutions are seldom identical. Yet, upon closer examination, you discover that integration solutions can actually be classified into common categories. Each of these categories describes both a "type" of integration problem as well as a solution method. These categories are called integration patterns. Integration patterns help you understand the different methods available to you for a given type of integration problem. They allow you to take a step back and understand the differences in the various scenarios and appreciate the different approaches to integration. Finally, they allow you to view "integration in the big picture." You can learn to break down what may be a complex integration into conceptual categories and understand which technologies to apply.

What Are Integration Patterns?

A pattern is commonly defined as a reliable sample of traits, acts, tendencies, or other observable characteristics. In software development, you may be familiar with the idea of design patterns or process patterns. Design patterns systematically describe object designs that can be employed for a common set of problems. Similarly, process patterns describe proven methods and processes used in software development. In practice, patterns are simply a logical classification of commonly recurring actions, techniques, designs, or organizations. What are integration patterns? Integration patterns emerge from classification of standard solutions for integration scenarios. They are not patterns of design or code. Nor are they patterns of operational processes for an integration project. Instead, each integration pattern defines a type of integration problem, a solution technique, as well as parameters applied for e-Business Integration.

Following are seven common e-Business Integration patterns. They are not meant to be comprehensive, but they cover most of the common integration scenarios implemented today. They encompass both EAI scenarios as well as B2Bi scenarios:

- **EAI (intra-enterprise) Patterns**
  1. Database Replication
  2. Single-Step Application Integration
  3. Multi-Step Application Integration
  4. Brokering Application
- **B2Bi (inter-enterprise) Patterns**
  5. Application-to-Application B2Bi
  6. Data Exchange B2Bi
  7. B2B Process Integration

The EAI Patterns represent patterns commonly applied within a corporate enterprise, whereas the B2Bi Patterns represent the different methods in conducting integrated B2B transactions. The following sections provide a closer look at each of these patterns and discuss some of the details.

Database Replication

The Database Replication pattern may be the most prevalent pattern of EAI integration today. Database replication involves managing copies of data over two or more databases, resulting in redundant data. Companies engage in database replication for numerous reasons. One reason is that many organizations are becoming more distributed in their operations, requiring multiple copies of the same data over several physical locations. Replication is also a means of data recovery. In many organizations, an active secondary database is maintained for data recovery purposes. In the event that the production database needs to be recovered, the secondary replicated database can be used. This also applies for "high availability" systems. In these situations, a redundant copy of "live" data is

maintained to ensure that if the first system is not available, the redundant database system is activated. The two general categories for database replication are synchronous and asynchronous replication.

Single-Step Application Integration

The Single-Step Application Integration (SSAI) pattern extends the asynchronous database replication pattern. Instead of focusing on data consistency between two databases, the SSAI pattern integrates data between applications, moving data from one context to another. It does so by translating data syntax of the source message and reformatting data elements into a new target message. It is "single step" because it requires an intermediary broker to map source messages to target messages. Typically, it is an extension of the asynchronous replication technology, in that it utilizes Message Queuing Middleware such as MQ Series. It is just as likely to be implemented with the less sophisticated FTP in a batch mode. In either case, the point is that it does more than simply move data from point A to point B for consistency's sake. Whereas, in the replication pattern both the source and target data models are likely similar, if not identical at times, this is not necessarily the case for the SSAI pattern. The objective here is not data consistency, but application data integration.

Multi-Step Application Integration

The Multi-Step Application Integration (MSAI) pattern is an extension of the SSAI pattern. MSAI enables the integration of n (source) to m (target) applications. It addresses many-to-many integration, which SSAI cannot, by providing what is known as sequential logical processing. In other words, steps in this pattern are processed sequentially, and rules applied are Boolean logical in nature. Like the single-step pattern, MSAI requires an intermediary to broker the transaction of data between applications. It is often built around an asynchronous event-based system and typically is implemented through the use of Message Queuing Middleware as well. The asynchronous eventbased approach creates loose coupling. Although each system is physically independent, they are logically dependent. In other words, interdependencies exist between the application events that can be expressed in terms of transformations and data integration rules. Data elements from one application can drive the retrieval or processing of messages in another application. The simplest multi-step example in Figure 3.3 involves three applications in which a message from application A is combined with a message from application B that is reformatted for a target application C. It is common for a data element from application A to act as a key to drive the request for information from application B.

Brokering Application

At times integrating two applications is not principally a matter of integrating data, but integrating business logic. The Brokering Application pattern addresses the use of intermediary application logic to link together two or more applications. In plain terms, it means that custom application code is written containing logic to broker interactions between the disparate applications. This custom brokering application sits in the middle as an intermediary for processing requests from different applications

The use of this solution pattern is particularly applicable in the scenarios below:

- Applications Need to Reuse Logic
- Applications Linked by Complex Logic
- Applications Unified Through User Interface

Application-to-Application B2Bi

Now you're ready to move beyond EAI to learn about Application-to-Application B2Bi, extending integration beyond the corporate enterprise. I will describe four additional patterns related specifically to B2B integration, beginning first with the Application-to-Application B2Bi pattern. The Application-to-Application pattern is the logical extension of what occurs in EAI. When EAI vendors tout their products as being B2Bi, this specific pattern is what they have in mind. However, as you will discover, this is not the only pattern and very likely not even the primary pattern for B2Bi. Application-to-Application B2Bi, which is often referred to as inter-enterprise integration, involves corporate

E-Business Strategies

What is an E-Business Strategy?

- E-Business has triggered new business models, strategies and tactics that are made possible by the internet and other related technologies.
- In order to compete in the marketplace, it is essential for organizations to establish strategies for the development of an e-business.
- E-Business strategy can be viewed via two different viewpoints, which are explained below.
- One view defines strategy as plans and objectives adopted to achieve higher-level goals.
- In that sense, a strategy is developed to achieve a goal like implementing organizational change, or a large software package such as an ERP-system.
- Strategy may also relate to plans concerning the long-term position of the firm in its business environment to achieve its organizational goals.
- Based on the above, we arrive at a common definition for an e-Business Strategy.
- An e-Business strategy is the set of plans and objectives by which applications of internal and external electronically mediated communication contribute to the corporate strategy.
- Strategic planning comprises a distinct class of decisions (a plan is a set of decisions made for the future) and objectives, and has to be positioned next to tactical planning (structuring the resources of the firm) and operational planning (maximizing the profitability of the current operations).
- Strategy is concerned with changes in the competitive environment that may trigger strategic changes for the individual firm and so affect its roles and functions in the market.
- Reassessment of strategy may occur due to:
  - New Products
  - Changing customer preferences
    - Flowers: Roses / Carnations -> Orchids
    - A few years back when people went to the florist, they generally picked up Roses or Carnations etc. Now, they prefer Orchids. This is an example of changing customer preferences. A global notion is that a customer does not realize the utility of feel the need for a product until it is offered to him / her.
  - Changing demand patterns
  - New competitors
- The frequency, dynamics and predictability of the above changes dictate the intensity of the strategic planning activity of the firm.
- So, e-Business strategy (revised) is:
  - The set of plans and objectives by which applications of internal and external electronically mediated communication contribute to the corporate strategy.
- E-Business strategy may be implemented for:
  - Tactical purposes: Mail -> EDI ->XML-FDI
  - Achieving corporate strategy objectives
- E-Business is strategic in nature.
  - The idea is to create a preferably sustainable & competitive position for the company.
    - This is achieved by integration of the Internet and related technologies in its primary processes.
- E-Business must not only support corporate strategy objectives but also functional strategies (SCM, Marketing)
- **Supply Chain Management Strategy**

- Based on value chain analysis for decomposing an organization into its individual activities and determining value added at each stage.
- Gauge efficiency in use of resources at each stage.

- **Marketing Strategy**
  - Is a concerned pattern of actions taken in the market environment to create value for the firm by improving its economic performance.
  - Focused on capturing market share or improving profitability via brand-building etc.
  - Operates on CURRENT AS WELL AS FUTURE projections of customer demand.
- **Information Systems Strategy**
  - How to leverage information systems in an organization to support the objectives of an organization in the long run.

- **E-Business strategy is based on corporate objectives.**

Strategic Positioning

Strategic positioning means that a firm is doing things differently from its competitors in a way that delivers a unique value to its customers. There are 6 fundamental principles a firm must follow to establish and maintain a distinctive strategic position:

1. Start with the right goal: superior long term ROI.
2. Strategy must enable it to deliver a value proposition different from competitors.
3. Strategy must be reflected in a distinctive value chain.
4. Accept tradeoffs for a robust strategy.
5. Strategy must define how all elements of what a firm does fit together.
6. Strategy must involve continuity of direction.

Levels of e-Business Strategies

Strategies will exist at different levels of an organization. Strategic levels of management are concerned with integrating and coordinating the activities of an organization so that the behavior is optimized and its overall direction is consistent with its mission. Ultimately e-Business is about communication, within business units and between units of the enterprise as well as organizations.

1) **Supply Chain or Industry Value Chain level**
   - E-Business requires a view of the role, added value, and position of the firm in the supply chain.
   - Important issues that need to be addressed at this level are:
     i. Who are the firm's direct customers?
     ii. What is the firm's value proposal to the customers?
     iii. Who are the suppliers?
     iv. How does the firm add value to the suppliers?
     v. What is the current performance of the Supply Chain in terms of revenue and profitability, inventory levels etc?
     vi. More importantly, what are the required performance levels?
     vii. What are the current problems in the chain?
   - This sort of analysis give insight into in upstream (supplier side) and downstream (customer side) data and information flows.
2) **The Line of Business or (Strategic) Business Unit level**
   - Understanding the position in the value chain is a starting point for further analysis of how Internet-related technologies could contribute to the competitive strategy of a business.
   - This is the level where competitive strategy in a particular market for a particular product is developed (Strategic Positioning).

   - There are four generic strategies for achieving a profitable business:
     I. **Differentiation:** This strategy refers to all the ways producers can make their product unique and distinguish them from those of their competitors.
     II. **Cost:** Adopting a strategy for cost competition means that the company primarily competes with low cost; customers are interested in buying a product as inexpensively as possible. Success in such a market implies that the company has discovered a unique business model which makes it possible to deliver the product or service at the lowest possible cost.
     III. **Scope:** A scope strategy is a strategy to compete in markets worldwide, rather than merely in local or regional markets.
     IV. **Focus:** A focus strategy is a strategy to compete within a narrow market segment or product segment.
3) **The Corporate or Enterprise level**
   - This level comprises a collection of (strategic) business units.
   - This level addresses the problem of synergy through a firm-wide, available IT infrastructure.
   - Common e-Business applications throughout the organization are needed for two basic reasons.
   - From efficiency point of view, having different applications for the same functionality in different areas of business is needlessly costly.
   - From an effectiveness point of view, there is the need for cross Line of Business communication and share-ability of data.
   - The emphasis in the business plans is on the customer, not the final product.
   - These all become subjects of an enterprise-wide e-Business policy.

The changing competitive Agenda: Business & Technology Drivers

Business Drivers:

- Shift in economies from supply driven to demand driven
  - Causes a shift in intent of service and quality programs, the impetus for product development & the structure of the organization itself
  - One to One marketing
  - Mass Customization

Technological Drivers:

- Internet
  - Pervasiveness
  - Interactive Nature
  - Virtual Nature

Strategic Planning Process

- The strategic planning process has the following steps:
- The strategic planning process starts with the establishment of the organization's mission statement.
  - The mission statement is a basic description of detailing the fundamental purpose of the organizations existence and encompasses strategy development, including determination of the organization's vision and objectives.
  - It is developed at the highest level of the organizations management, and provides a general sense of direction for all decision making within the firm.
- Strategic Analysis
  - This involves situation analysis, internal resource assessment, and evaluation of stakeholder's expectation.

- It will include
  - Environmental Scanning
  - Industry or market research
  - Competitor Analysis
  - Analysis of Marketplace Structure
  - Relationships with trading partners and suppliers
  - Customer Marketing Research
- Information is derived from the analysis of both internal and external factors.
- Internal Factors:
  - Human resources
  - Material resources
  - Informational resources
  - Financial resources
  - Structure
  - Operational Style
  - Culture
- External Factors:
  - Socio-cultural forces
  - Technological forces
  - Legal and regulatory forces
  - Political forces
  - Economic forces
  - Competitive forces
- Any realistic new plan will have to reflect the reality of both the external world and the internal dynamics of the corporation.

☑ Strategic Choice

- It is based on the strategic analysis and consists of four parts:
  - Generation of strategic options
  - Highlighting possible courses of action
  - Evaluation of strategic options on their relative merits
  - Selection of strategy
- Strategic choice results in Strategic Planning, which is concerned with the organizing and detailing of all the strategies that will be undertaken throughout the organization.
- Planning includes strategy specification and resource allocation.
- It commences with corporate-level planning that determines the overall direction for the organization.
- This drives Division (or Strategic Business Unit) level planning which deals with groups of related products offered by the organization.
- These plans in turn become the starting point for operating (or functional) level planning, which involves more local plans within specific departments of the organization.

☑ Implementation

- This relates to the actual tasks that must be executed in order to realize a plan and translates strategy into action.
- It includes monitoring, adjustment, control as well as feedback.

Enterprise Application Integration

Enterprise Application Integration (EAI) is defined as the use of software and computer systems architectural principles to integrate a set of enterprise computer applications. Enterprise Application Integration (EAI) is an integration framework composed of a collection of technologies and services which form a middleware to enable integration of systems and applications across the enterprise. Supply chain management applications (for managing inventory and shipping), customer relationship management applications (for managing current and potential customers), business intelligence applications (for finding patterns from existing data from operations), and other types of applications (for managing data such as human resources data, health care, internal communications, etc) typically cannot communicate with one another in order to share data or business rules.

For this reason, such applications are sometimes referred to as islands of automation or information silos. This lack of communication leads to inefficiencies, wherein identical data are stored in multiple locations, or straightforward processes are unable to be automated. Enterprise application integration (EAI) is the process of linking such applications within a single organization together in order to simplify and automate business processes to the greatest extent possible, while at the same time avoiding having to make sweeping changes to the existing applications or data structures. In the words of the Gartner Group, EAI is the "unrestricted sharing of data and business processes among any connected application or data sources in the enterprise."

One large challenge of EAI is that the various systems that need to be linked together often reside on different operating systems, use different database solutions and different computer languages, and in some cases are legacy systems that are no longer supported by the vendor who originally created them. In some cases, such systems are dubbed "stovepipe systems" because they consist of components that have been jammed together in a way that makes it very hard to modify them in any way.

Purposes of EAI

EAI can be used for different purposes:

- Data (information) Integration: Ensuring that information in multiple systems is kept consistent. This is also known as EII (Enterprise Information Integration).
- Vendor independence: Extracting business policies or rules from applications and implementing them in the EAI system, so that even if one of the business applications is replaced with a different vendor's application, the business rules do not have to be re-implemented.
- Common Facade: An EAI system could front-end a cluster of applications, providing a single consistent access interface to these applications and shielding users from having to learn to interact with different software packages.

EAI patterns

Integration patterns

There are two patterns that EAI systems implement:

- Mediation: Here, the EAI system acts as the go-between or broker between (interface or communicating) multiple applications. Whenever an interesting event occurs in an application (e. g., new information created, new transaction completed, etc.) an integration module in the EAI system is notified. The module then propagates the changes to other relevant applications.
- Federation: In this case, the EAI system acts as the overarching facade across multiple applications. All event calls from the 'outside world' to any of the applications are front-ended by the EAI system. The EAI system is configured to expose only the relevant information and interfaces of the underlying applications to the outside world, and performs all interactions with the underlying applications on behalf of the requester.

Both patterns are often used concurrently. The same EAI system could be keeping multiple applications in sync (mediation), while servicing requests from external users against these applications (federation).

Access patterns

EAI supports both asynchronous and synchronous access patterns, the former being typical in the mediation case and the latter in the federation case.

Lifetime patterns

An integration operation could be short-lived (e. g., keeping data in sync across two applications could be completed within a second) or long-lived (e. g., one of the steps could involve the EAI system interacting with a human work flow application for approval of a loan that takes hours or days to complete).

EAI topologies

There are two major topologies: hub-and-spoke, and bus. Each has its own advantages and disadvantages. In the hub-and-spoke model, the EAI system is at the center (the hub), and interacts with the applications via the spokes. In the bus model, the EAI system is the bus (or is implemented as a resident module in an already existing message bus or message-oriented middleware).

Technologies

Multiple technologies are used in implementing each of the components of the EAI system:

Bus/hub

This is usually implemented by enhancing standard middleware products (application server, message bus) or implemented as a stand-alone program (i. e., does not use any middleware), acting as its own middleware.

Application connectivity

The bus/hub connects to applications through a set of adapters (also referred to as connectors). These are programs that know how to interact with an underlying business application. The adapter performs two-way communication, performing requests from the hub against the application, and notifying the hub when an event of interest occurs in the application (a new record inserted, a transaction completed, etc.). Adapters can be specific to an application (e. g., built against the application vendor's client libraries) or specific to a class of applications (e. g., can interact with any application through a standard communication protocol, such as SOAP or SMTP). The adapter could reside in the same process space as the bus/hub or execute in a remote location and interact with the hub/bus through industry standard protocols such as message queues, web services, or even use a proprietary protocol. In the Java world, standards such as JCA allow adapters to be created in a vendor-neutral manner.

Data format and transformation

To avoid every adapter having to convert data to/from every other applications' formats, EAI systems usually stipulate an application-independent (or common) data format. The EAI system usually provides a data transformation service as well to help convert between application-specific and common formats. This is done in two steps: the adapter converts information from the application's format to the bus's common format. Then, semantic transformations are applied on this (converting zip codes to city names, splitting/merging objects from one application into objects in the other applications, and so on).

Integration modules

An EAI system could be participating in multiple concurrent integration operations at any given time, each type of integration being processed by a different integration module. Integration modules subscribe to events of specific types and process notifications that they receive when these events occur. These modules could be implemented in different ways: on Java-based EAI systems, these could be web applications or EJBs or even POJOs that conform to the EAI system's specifications.

Support for transactions

When used for process integration, the EAI system also provides transactional consistency across applications by executing all integration operations across all applications in a single overarching distributed transaction (using two-phase commit protocols or compensating transactions).

Communication architectures

Currently, there are many variations of thought on what constitutes the best infrastructure, component model, and standards structure for Enterprise Application Integration. There seems to be consensus that four components are essential for modern enterprise application integration architecture:

1. A centralized broker that handles security, access, and communication. This can be accomplished through integration servers (like the School Interoperability Framework (SIF) Zone Integration Servers) or through similar software like the Enterprise service bus (ESB) model that acts as a SOAP-oriented services manager.
2. An independent data model based on a standard data structure, also known as a Canonical data model. It appears that XML and the use of XML style sheets has become the de facto and in some cases de jure standard for this uniform business language.
3. A connector or agent model where each vendor, application, or interface can build a single component that can speak natively to that application and communicate with the centralized broker.
4. A system model that defines the APIs, data flow and rules of engagement to the system such that components can be built to interface with it in a standardized way.

Although other approaches like connecting at the database or user-interface level have been explored, they have not been found to scale or be able to adjust. Individual applications can publish messages to the centralized broker and subscribe to receive certain messages from that broker. Each application only requires one connection to the broker. This central control approach can be extremely scalable and highly evolvable. Enterprise Application Integration is related to middleware technologies such as message-oriented middleware (MOM), and data representation technologies such as XML. Other EAI technologies involve using web services as part of service-oriented architecture as a means of integration. Enterprise Application Integration tends to be data centric. In the near future, it will come to include content integration and business processes.

EAI Implementation Pitfalls

In 2003 it was reported that 70% of all EAI projects fail.

Most of these failures are not due to the software itself or technical difficulties, but due to management issues. Integration Consortium European Chairman Steve Craggs has outlined the seven main pitfalls undertaken by companies using EAI systems and explains solutions to these problems.

- **Constant change**

  The very nature of EAI is dynamic and requires dynamic project managers to manage their implementation.

- **Shortage of EAI experts**

  EAI requires knowledge of many issues and technical aspects.

- **Competing standards**

  Within the EAI field, the paradox is that EAI standards themselves are not universal.

UNIT-4

E-Commerce Infrastructure

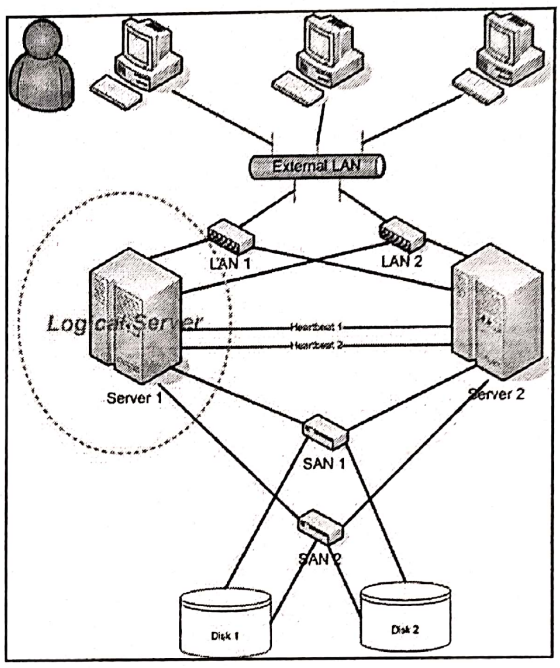
Cluster of Servers

A computer cluster is a group of linked computers, working together closely thus in many respects forming a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks.

Clusters are usually deployed to improve performance and/or availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

Cluster Categorizations

- High-availability (HA) clusters
  - High-availability clusters (also known as Failover Clusters) are implemented primarily for the purpose of improving the availability of services that the cluster provides.
  - They operate by having redundant nodes, which are then used to provide service when system components fail.
  - The most common size for an HA cluster is two nodes, which is the minimum requirement to provide redundancy.
  - HA cluster implementations attempt to use redundancy of cluster components to eliminate single points of failure.
  - There are commercial implementations of High-Availability clusters for many operating systems.
  - One such implementation is the Gridlock platform from http://www.obsidiandynamics.com.



  - The Linux-HA project is one commonly used free software HA package for the Linux operating system.
  - The LanderCluster from Lander Software can run on Windows, Linux, and UNIX platforms.
- Load-balancing clusters
  - Load-balancing is when multiple computers are linked together to share computational workload or function as a single virtual computer.
  - Logically, from the user side, they are multiple machines, but function as a single virtual machine.
  - Requests initiated from the user are managed by, and distributed among, all the standalone computers to form a cluster.
  - This results in balanced computational work among different machines, improving the performance of the cluster systems.
- Compute Clusters
  - Often clusters are used primarily for computational purposes, rather than handling IO-oriented operations such as web service or databases.
  - For instance, a cluster might support computational simulations of weather or vehicle crashes.
  - The primary distinction within compute clusters is how tightly-coupled the individual nodes are.
  - For instance, a single compute job may require frequent communication among nodes - this implies that the cluster shares a dedicated network, is densely located, and probably has homogenous nodes.

- This cluster design is usually referred to as Beowulf Cluster.
- The other extreme is where a compute job uses one or few nodes, and needs little or no inter-node communication.
- This latter category is sometimes called "Grid" computing.
- Tightly-coupled compute clusters are designed for work that might traditionally have been called "supercomputing".
- Middleware such as MPI (Message Passing Interface) or PVM (Parallel Virtual Machine) permits compute clustering programs to be portable to a wide variety of clusters.

- Grid Computing
  - Grids are usually computer clusters, but more focused on throughput like a computing utility rather than running fewer, tightly-coupled jobs.
  - Often, grids will incorporate heterogeneous collections of computers, possibly distributed geographically, sometimes administered by unrelated organizations.
  - Grid computing is optimized for workloads which consist of many independent jobs or packets of work, which do not have to share data between the jobs during the computation process.
  - Grids serve to manage the allocation of jobs to computers which will perform the work independently of the rest of the grid cluster.
  - Resources such as storage may be shared by all the nodes, but intermediate results of one job do not affect other jobs in progress on other nodes of the grid.
  - An example of a very large grid is the Folding@home project.
  - It is analyzing data that is used by researchers to find cures for diseases such as Alzheimer's and cancer.
  - Another large project is the SETI@home project, which may be the largest distributed grid in existence.
  - It uses approximately three million home computers all over the world to analyze data from the Arecibo Observatory radiotelescope, searching for evidence of extraterrestrial intelligence.
  - In both of these cases, there is no inter-node communication or shared storage.
  - Individual nodes connect to a main, central location to retrieve a small processing job.
  - They then perform the computation and return the result to the central server.
  - In the case of the @home projects, the software is generally run when the computer is otherwise idle.

A cluster of servers allows servers to work together as computer cluster, to provide failover and increased availability of applications, or parallel calculating power in case of high-performance computing (HPC) clusters (as in supercomputing).

A server cluster is a group of independent servers working together as a single system to provide high availability of services for clients. When a failure occurs on one computer in a cluster, resources are redirected and the workload is redistributed to another computer in the cluster.

You can use server clusters to ensure that users have constant access to important server-based resources. Server clusters are designed for applications that have long-running in-memory state or frequently updated data. Typical uses for server clusters include file servers, print servers, database servers, and messaging servers.

A cluster consists of two or more computers working together to provide a higher level of availability, reliability, and scalability than can be obtained by using a single computer. Microsoft cluster technologies guard against three specific types of failure:

- Application and service failures, which affect application software and essential services.

- System and hardware failures, which affect hardware components such as CPUs, drives, memory, network adapters, and power supplies.
- Site failures in multisite organizations, which can be caused by natural disasters, power outages, or connectivity outages.

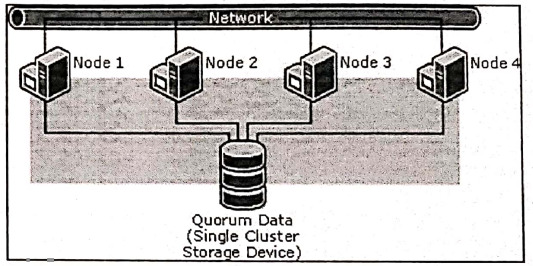
The ability to handle failure allows server clusters to meet requirements for high availability, which is the ability to provide users with access to a service for a high percentage of time while reducing unscheduled outages.

In a server cluster, each server owns and manages its local devices and has a copy of the operating system and the applications or services that the cluster is managing. Devices common to the cluster, such as disks in common disk arrays and the connection media for accessing those disks, are owned and managed by only one server at a time. For most server clusters, the application data is stored on disks in one of the common disk arrays, and this data is accessible only to the server that currently owns the corresponding application or service.

Server clusters are designed so that the servers in the cluster work together to protect data, keep applications and services running after failure on one of the servers, and maintain consistency of the cluster configuration over time.

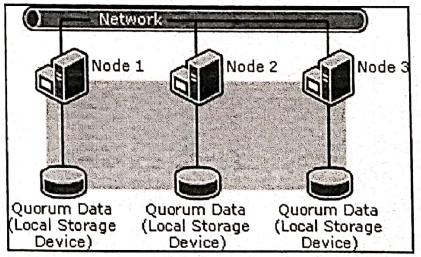
Types of Server Clusters

Single Quorum Device Cluster



The most widely used cluster type is the single quorum device cluster, also called the standard quorum cluster. In this type of cluster there are multiple nodes with one or more cluster disk arrays, also called the cluster storage, and a connection device, that is, a bus. Each disk in the array is owned and managed by only one server at a time. The disk array also contains the quorum resource. The following figure illustrates a single quorum device cluster with one cluster disk array. Because single quorum device clusters are the most widely used cluster, this Technical Reference focuses on this type of cluster.

Majority Node Set Cluster

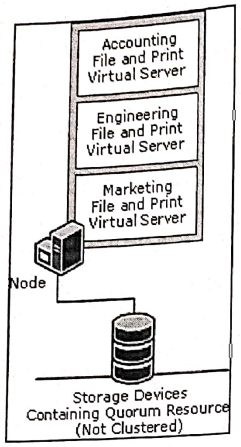


Windows Server 2003 supports another type of cluster, the majority node set cluster. In a majority node set cluster, each node maintains its own copy of the cluster configuration data. The quorum resource keeps configuration data consistent across the nodes. For this reason, majority node set clusters can be used for geographically dispersed clusters. Another advantage of majority node set clusters is that a quorum disk can be taken offline for maintenance and the cluster as a whole will continue to operate.

The major difference between majority node set clusters and single quorum device clusters is that single quorum device clusters can operate with just one node, but majority node set clusters need to have a majority of the cluster nodes available for the server cluster to operate. The following figure illustrates a majority node set cluster. For the cluster in the figure to continue to operate, two of the three cluster nodes (a majority) must be available.

Local Quorum Cluster

A local quorum cluster, also called a single node cluster, has a single node and is often used for testing. The following figure illustrates a local quorum cluster.



Virtualization

Server virtualization is the masking of server resources, including the number and identity of individual physical servers, processors, and operating systems, from server users. The server administrator uses a software application to divide one physical server into multiple isolated virtual environments. The virtual environments are sometimes called virtual private servers, but they are also known as guests, instances, containers or emulations.

Virtualization was invented more than thirty years ago to allow large expensive mainframes to be easily shared among different application environments. As hardware prices went down, the need for virtualization faded away. More recently, virtualization at all levels (system, storage, and network) became important again as a way to improve system security, reliability and availability, reduce costs, and provide greater flexibility.

Approaches to Virtualization

There are three popular approaches to server virtualization:

1. The virtual machine model,
2. The paravirtual machine model,
3. Virtualization at the operating system (OS) layer.

Virtual Machines

Virtual machines are based on the host/guest paradigm. Each guest runs on a virtual imitation of the hardware layer. This approach allows the guest operating system to run without modifications. It also allows the administrator to create guests that use different operating systems. The guest has no knowledge of the host's operating system because it is not aware that it's not running on real hardware.

It does, however, require real computing resources from the host -- so it uses a hypervisor to coordinate instructions to the CPU. The hypervisor is called a virtual machine monitor (VMM). It validates all the guest-issued CPU instructions and manages any executed code that requires addition privileges. VMware and Microsoft Virtual Server both use the virtual machine model.

Para-virtual Machine

The paravirtual machine (PVM) model is also based on the host/guest paradigm -- and it uses a virtual machine monitor too. In the paravirtual machine model, however, The VMM actually modifies the guest operating system's code. This modification is called porting. Porting supports the VMM so it can utilize privileged systems calls sparingly. Like virtual machines, paravirtual machines are capable of running multiple operating systems.

Virtualization @ OS Level

Virtualization at the OS level works a little differently. It isn't based on the host/guest paradigm. In the OS level model, the host runs a single OS kernel as its core and exports operating system functionality to each of the guests. Guests must use the same operating system as the host, although different distributions of the same system are allowed.

This distributed architecture eliminates system calls between layers, which reduces CPU usage overhead. It also requires that each partition remain strictly isolated from its neighbors so that a failure or security breach in one partition isn't able to affect any of the other partitions.

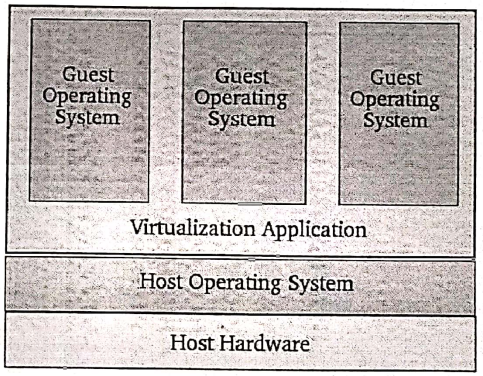
In this model, common binaries and libraries on the same physical machine can be shared, allowing an OS level virtual server to host thousands of guests at the same time.

Virtualization Techniques

Guest Operating System Virtualization

In this scenario the physical host computer system runs a standard unmodified operating system such as Windows, Linux, Unix or MacOS X. Running on this operating system is a virtualization application which executes in much the same way as any other application such as a word processor or spreadsheet would run on the system. It is within this virtualization application that one or more virtual machines are created to run the guest operating systems on the host computer.

The virtualization application is responsible for starting, stopping and managing each virtual machine and essentially controlling access to physical hardware resources on behalf of the individual virtual machines. The virtualization application also engages in a process known as binary rewriting which involves scanning the instruction stream of the executing guest system and replacing any privileged instructions with safe emulations.



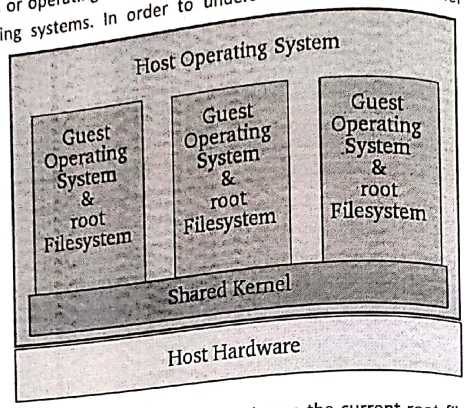
This has the effect of making the guest system think it is running directly on the system hardware, rather than in a virtual machine within an application.

As outlined in the above diagram, the guest operating systems operate in virtual machines within the virtualization application which, in turn, runs on top of the host operating system in the same way as any other application. Clearly, the multiple layers of abstraction between the guest operating systems and the underlying host hardware are not conducive to high levels of virtual machine performance.

This technique does, however, have the advantage that no changes are necessary to either host or guest operating systems and no special CPU hardware virtualization support is required.

Shared Kernel Virtualization

Shared kernel virtualization (also known as system level or operating system virtualization) takes advantage of the architectural design of Linux and UNIX based operating systems. In order to understand how shared kernel virtualization works it helps to first understand the two main components of Linux or UNIX operating systems. At the core of the operating system is the kernel. The kernel, in simple terms, handles all the interactions between the operating system and the physical hardware. The second key component is the root file system which contains all the libraries, files and utilities necessary for the operating system to function. Under shared kernel virtualization the virtual guest systems each have their own root file system but share the kernel of the host operating system. This structure is illustrated in the following architectural diagram.

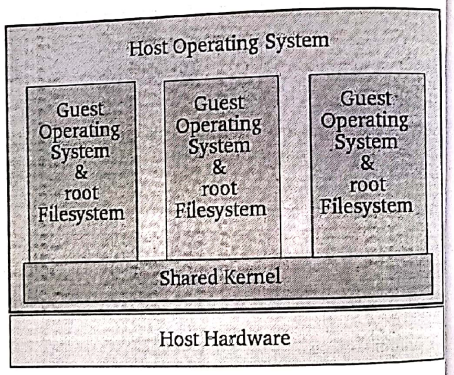


This type of virtualization is made possible by the ability of the kernel to dynamically change the current root file system (a concept known as chroot) to a different root file system without having to reboot the entire system. Essentially, shared kernel virtualization is an extension of this capability.

Perhaps the biggest single drawback of this form of virtualization is the fact that the guest operating systems must be compatible with the version of the kernel which is being shared. It is not, for example, possible to run Microsoft Windows as a guest on a Linux system using the shared kernel approach. Nor is it possible for a Linux guest system designed for the 2.6 version of the kernel to share a 2.4 version kernel.

Kernel Level Virtualization

Under kernel level virtualization the host operating system runs on a specially modified kernel which contains extensions designed to manage and control multiple virtual machines each containing a guest operating system. Unlike shared kernel virtualization each guest runs its own kernel, although similar restrictions apply in that the guest operating systems must have been compiled for the same hardware as the kernel in which they are running. Examples of kernel level virtualization technologies include User Mode Linux (UML) and Kernel-based Virtual Machine (KVM). The following diagram provides an overview of the kernel level virtualization architecture:



Cloud Computing

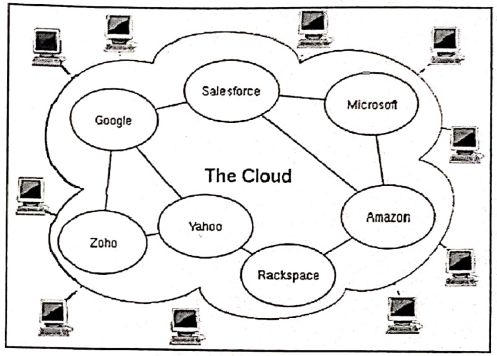
Cloud computing is Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand, like the electricity grid. Cloud computing is a paradigm shift following the shift from mainframe to client–server in the early 1980s. Details are abstracted from the users, who no longer have need for expertise in, or control over, the technology infrastructure "in the cloud" that supports them.

Cloud computing describes a new supplement, consumption, and delivery model for IT services based on the Internet, and it typically involves over-the-Internet provision of dynamically scalable and often virtualized resources. It is a byproduct and consequence of the ease-of-access to remote computing sites provided by the Internet. This frequently takes the form of web-based tools or applications that users can access and use through a web browser as if it were a program installed locally on their own computer.

NIST provides a somewhat more objective and specific definition here. The term "cloud" is used as a metaphor for the Internet, based on the cloud drawing used in the past to represent the telephone network and later to depict the Internet in computer network diagrams as an abstraction of the underlying infrastructure it represents.

Typical cloud computing providers deliver common business applications online that are accessed from another Web service or software like a Web browser, while the software and data are stored on servers. A key element of cloud computing is customization and the creation of a user-defined experience.

Most cloud computing infrastructures consist of services delivered through common centers and built on servers.



Clouds often appear as single points of access for consumers' computing needs. Commercial offerings are generally expected to meet quality of service (QoS) requirements of customers, and typically include service level agreements (SLAs).The major cloud service providers include Salesforce, Amazon and Google. Some of the larger IT firms that are actively involved in cloud computing are Microsoft, Hewlett Packard and IBM.

Cloud computing derives characteristics from, but should not be confused with:

1. Autonomic computing — "computer systems capable of self-management"
2. Client–server model – client–server computing refers broadly to any distributed application that distinguishes between service providers (servers) and service requesters (clients).
3. Grid computing — "a form of distributed computing and parallel computing, whereby a 'super and virtual computer' is composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks"
4. Mainframe computer — powerful computers used mainly by large organizations for critical applications, typically bulk data-processing such as census, industry and consumer statistics, enterprise resource planning, and financial transaction processing.
5. Utility computing — the "packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility, such as electricity";
6. Peer-to-peer – a distributed architecture without the need for central coordination, with participants being at the same time both suppliers and consumers of resources (in contrast to the traditional client–server model)

Data!

We live in the data age. It's not easy to measure the total volume of data stored electronically, but an IDC estimate put the size of the "digital universe" at 0.18 zettabytes in 2006, and is forecasting a tenfold growth by 2011 to 1.8 zettabytes.* A zettabyte is 10^{21} bytes, or equivalently one thousand exabytes, one million petabytes, or one billion terabytes. That's roughly the same order of magnitude as one disk drive for every person in the world. This flood of data is coming from many sources. Consider the following:

- The New York Stock Exchange generates about one terabyte of new trade data per day.
- Facebook hosts approximately 10 billion photos, taking up one petabyte of storage.
- Ancestry.com, the genealogy site, stores around 2.5 petabytes of data.
- The Internet Archive stores around 2 petabytes of data, and is growing at a rate of 20 terabytes per month.
- The Large Hadron Collider near Geneva, Switzerland, will produce about 15 petabytes of data per year.

So there's a lot of data out there. But you are probably wondering how it affects you. Most of the data is locked up in the largest web properties (like search engines), or scientific or financial institutions, isn't it? Does the advent of "Big Data," as it is being called, affect smaller organizations or individuals?

Arguably it does. Take photos, for example. My wife's grandfather was an avid photographer, and took photographs throughout his adult life. His entire corpus of medium format, slide, and 35mm film, when scanned in at high-resolution, occupies around 10 gigabytes. Compare this to the digital photos that my family took last year, which take up about 5 gigabytes of space. My family is producing photographic data at 35 times the rate my wife's grandfather's did, and the rate is increasing every year as it becomes easier to take more and more photos.

More generally, the digital streams that individuals are producing are growing apace. Microsoft Research's MyLifeBits project gives a glimpse of archiving of personal information that may become commonplace in the near future. MyLifeBits was an experiment where an individual's interactions—phone calls, emails, documents—were captured electronically and stored for later access. The data gathered included a photo taken every minute, which resulted in an overall data volume of one gigabyte a month. When storage costs come down enough to make it feasible to store continuous audio and video, the data volume for a future MyLifeBits service will be many times that.

The trend is for every individual's data footprint to grow, but perhaps more importantly the amount of data generated by machines will be even greater than that generated by people. Machine logs, RFID readers, sensor networks, vehicle GPS traces, retail transactions—all of these contribute to the growing mountain of data.

The volume of data being made publicly available increases every year too. Organizations no longer have to merely manage their own data: success in the future will be dictated to a large extent by their ability to extract value from other organizations' data.

Initiatives such as Public Data Sets on Amazon Web Services, Infochimps.org, and theinfo.org exist to foster the "information commons," where data can be freely (or in the case of AWS, for a modest price) shared for anyone to download and analyze. Mashups between different information sources make for unexpected and hitherto unimaginable applications.

Take, for example, the Astrometry.net project, which watches the Astrometry group on Flickr for new photos of the night sky. It analyzes each image, and identifies which part of the sky it is from, and any interesting celestial bodies, such as stars or galaxies. Although it's still a new and experimental service, it shows the kind of things that are possible when data (in this case, tagged photographic images) is made available and used for something (image

Characteristics

In general, cloud computing customers do not own the physical infrastructure, instead avoiding capital expenditure by renting usage from a third-party provider. They consume resources as a service and pay only for resources that they use. Many cloud-computing offerings employ the utility computing model, which is analogous to how traditional utility services (such as electricity) are consumed, whereas others bill on a subscription basis.

Sharing "perishable and intangible" computing power among multiple tenants can improve utilization rates, as servers are not unnecessarily left idle (which can reduce costs significantly while increasing the speed of application development).

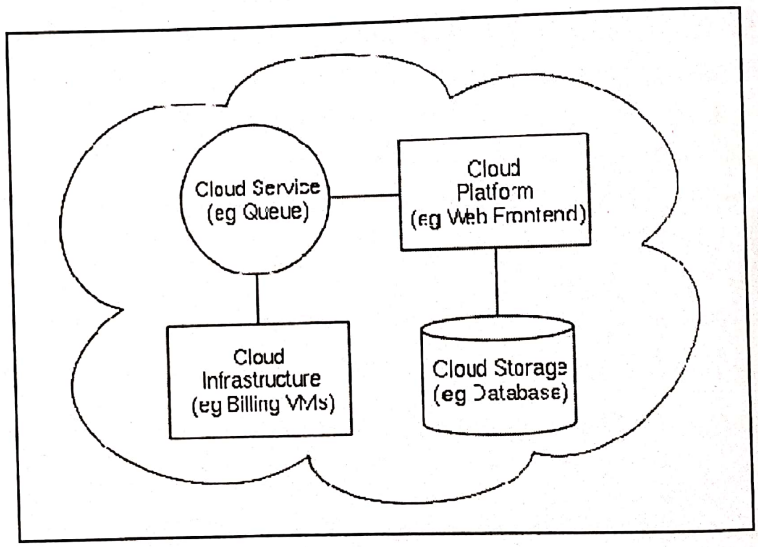
A side-effect of this approach is that overall computer usage rises dramatically, as customers do not have to engineer for peak load limits. In addition, "increased high-speed bandwidth" makes it possible to receive the same. The cloud is becoming increasingly associated with small and medium enterprises (SMEs) as in many cases they cannot justify or afford the large capital expenditure of traditional IT.

SMEs also typically have less existing infrastructure, less bureaucracy, more flexibility, and smaller capital budgets for purchasing in-house technology. Similarly, SMEs in emerging markets are typically unburdened by established legacy infrastructures, thus reducing the complexity of deploying cloud solutions.

Architecture

Cloud architecture, the systems architecture of the software systems involved in the delivery of cloud computing, typically involves multiple cloud components communicating with each other over applicati n programming interfaces, usually web services.

This resembles the Unix philosophy of having multiple programs each doing one thing well and working together over universal interfaces. Complexity is controlled and the resulting systems are more manageable than their monolithic counterparts.



The two most significant components of cloud computing architecture are known as the front end and the back end. The front end is the part seen by the client, i.e. the computer user. This includes the client's network (or computer) and the applications used to access the cloud via a user interface such as a web browser.

The back end of the cloud computing architecture is the 'cloud' itself, comprising various computers, servers and data storage devices.

Key Features

- **Agility** improves with users' ability to rapidly and inexpensively re-provision technological infrastructure resources.
- **Cost** is claimed to be greatly reduced and capital expenditure is converted to operational expenditure. This ostensibly lowers barriers to entry, as infrastructure is typically provided by a third-party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained with usage-based options and fewer IT skills are required for implementation (in-house).

It has been said that "More data usually beats better algorithms," which is to say that for some problems (such as recommending movies or music based on past preferences), however fiendish your algorithms are, they can often be beaten simply by having more data (and a less sophisticated algorithm). The good news is that Big Data is here. The bad news is that we are struggling to store and analyze it.

Data Storage and Analysis

The problem is simple: while the storage capacities of hard drives have increased massively over the years, access speeds—the rate at which data can be read from drives— have not kept up. One typical drive from 1990 could store 1370 MB of data and had a transfer speed of 4.4 MB/s,§ so you could read all the data from a full drive in around five minutes. Almost 20 years later one terabyte drives are the norm, but the transfer speed is around 100 MB/s, so it takes more than two and a half hours to read all the data off the disk.

This is a long time to read all data on a single drive—and writing is even slower. The obvious way to reduce the time is to read from multiple disks at once. Imagine if we had 100 drives, each holding one hundredth of the data. Working in parallel, we could read the data in under two minutes.

Only using one hundredth of a disk may seem wasteful. But we can store one hundred datasets, each of which is one terabyte, and provide shared access to them. We can imagine that the users of such a system would be happy to share access in return for shorter analysis times, and, statistically, that their analysis jobs would be likely to be spread over time, so they wouldn't interfere with each other too much.

There's more to being able to read and write data in parallel to or from multiple disks, though.

The first problem to solve is hardware failure: as soon as you start using many pieces of hardware, the chance that one will fail is fairly high. A common way of avoiding data loss is through replication: redundant copies of the data are kept by the system so that in the event of failure, there is another copy available. This is how RAID works, for instance, although Hadoop's filesystem, the Hadoop Distributed Filesystem (HDFS), takes a slightly different approach, as you shall see later.

The second problem is that most analysis tasks need to be able to combine the data in some way; data read from one disk may need to be combined with the data from any of the other 99 disks. Various distributed systems allow data to be combined from multiple sources, but doing this correctly is notoriously challenging. MapReduce provides a programming model that abstracts the problem from disk reads and writes, transforming it into a computation over sets of keys and values. The important point for the present discussion is that there are two parts to the computation, the map and the reduce, and it's the interface between the two where the "mixing" occurs. Like HDFS, MapReduce has reliability built-in.

This, in a nutshell, is what Hadoop provides: a reliable shared storage and analysis system. The storage is provided by HDFS, and analysis by MapReduce. There are other parts to Hadoop, but these capabilities are its kernel.

Comparison with Other Systems

The approach taken by MapReduce may seem like a brute-force approach. The premise is that the entire dataset—or at least a good portion of it—is processed for each query. But this is its power. MapReduce is a *batch* query processor, and the ability to run an ad hoc query against your whole dataset and get the results in a reasonable time is transformative. It changes the way you think about data, and unlocks data that was previously archived on tape or disk. It gives people the opportunity to innovate with data. Questions that took too long to get answered before can now be answered, which in turn leads to new questions and new insights.

For example, Mailtrust, Rackspace's mail division, used Hadoop for processing email logs. One ad hoc query they wrote was to find the geographic distribution of their users. In their words: