클라우드 컴퓨팅과 AI서비스 (고가용성)

융합학과 권오영

oykwon@koreatech.ac.kr

KOREATECH
한국기술교육대학교

고가용성이란

What Does “HA” Means?

- HA is actually relatively recent technology
- FT (Fault Tolerant)
  - Referred to systems that used various H/W techniques to make a single, stand-alone, piece of computer H/W (Tandem)
  - expensive FT system crashed nearly every day at recent HA survey
    - ✓ H/W was operating just fine
    - ✓ S/W buggy state caused repeated crashes (not tolerating user’s fault)
- Traditional H/W FT system
  - Rely on special H/W with unique to those systems
  - Special designed, low volume and expensive H/W
    - ✓ Tri- or bi-module redundancy, inter-module comparator, reliable voting logic, intensive internal error checking/correction, automatic H/W-based checkpoint/restart, robust electronic design
- HA Cluster
  - High volume, low cost, common off-the-self system H/W
  - Why use commodity component
    - ✓ Cost, performance scaling, HA cost

How High is “High”

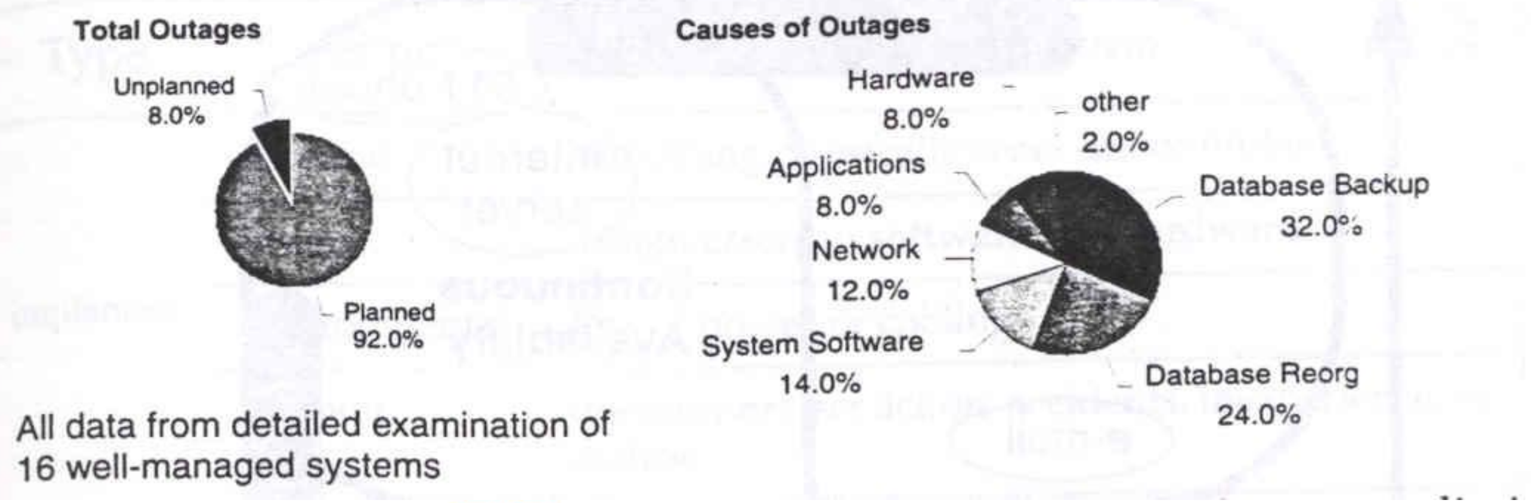
- System is available for use X % of the time
- How much total non-operation time(Outage) per year
- Availability class(# of 9’s)
  - Class 3, 4, 5: HA

Class	Availability	Total Outage	Type
1	90~99	more than a month ~ under 4 days	Campus wide LAN
2	99.9	under 9 hours	stand-alone, non-clustered open/commodity system
3	99.99	about an hour	open system-based cluster system tradition mainframe
4	99.999	a little over 5 minutes	
5	99.9999	about half a minute	telephone switch, IBM Parallel Sysplex
6	99.99999	about 3 second	in-flight aircraft computer

Facets of Availability(1)

- Unreasonable to count the time which not operate as “outage”
- Kind of Outage
  - Unplanned : something break, or operating incorrectly when it was unexpected (8%)
  - Planned : know ahead of time such as system maintenance (92%)
  - For mainframe system, planned outages are actually a much larger cause of down time than unplanned outages

Figure 104 Sources of Outage in IBM S/390 Systems (1990)

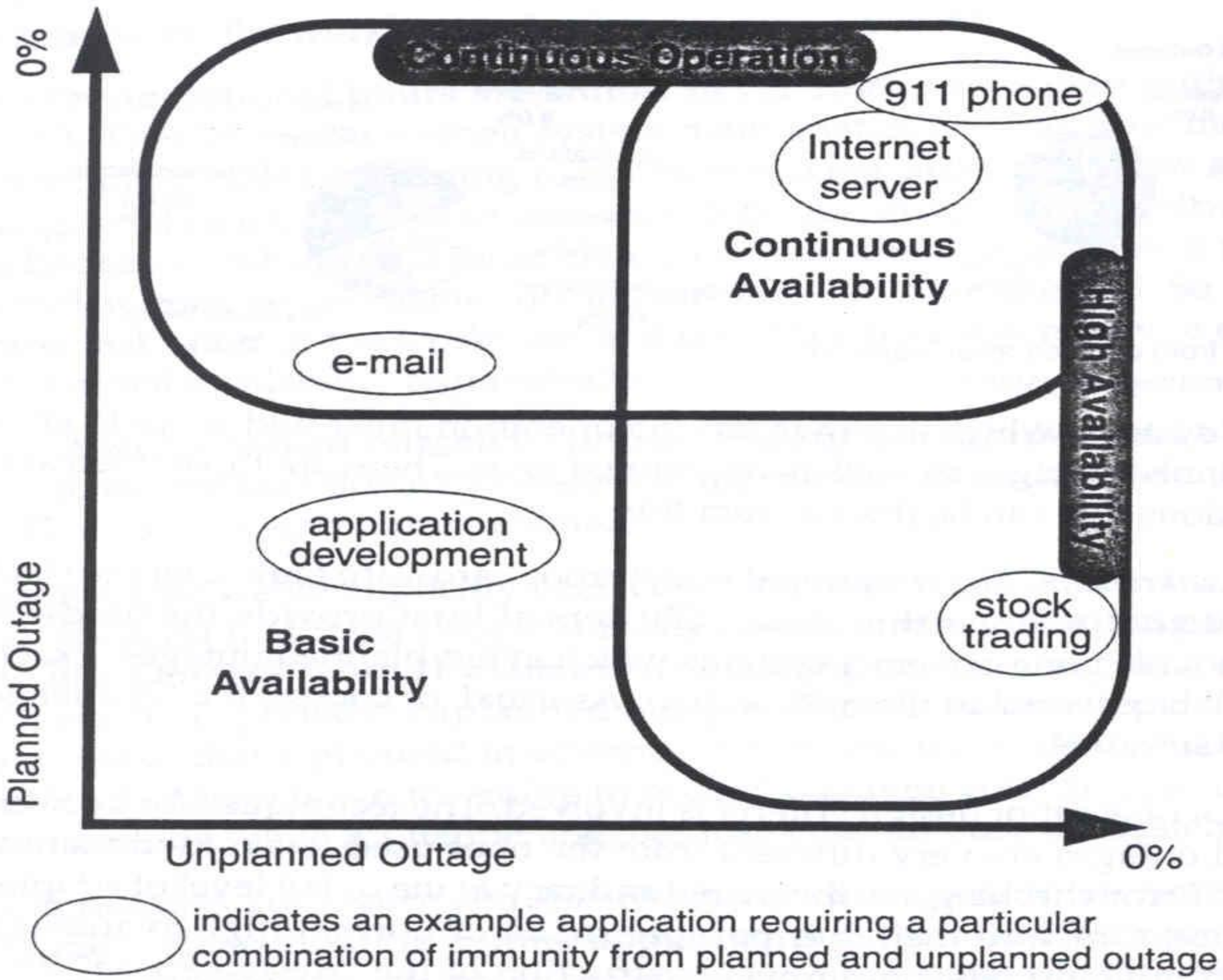


Facets of Availability(2)

- Many users, such as ATM Banks, grocery store, and Internet, want to eliminate the planned and unplanned outage
  - 24 hours a day and 365 day per year service (24x365)
- 3 implications for 24X365 operations
  - clusters rule : provide the H/W basis on which one can erect systems
    which avoid planned outage,
    and **challenge is in the S/W**.
  - new set of design criteria is involved.
    - ✓ HA: avoiding unplanned outage
    - ✓ Continuous Operation : avoiding planned outage
    - ✓ Continuous Availability : HA + CO
  - Divide the causes of the outages
    - ✓ better highlight the fact that different kinds of actions are required to avoid them

Facets of Availability(3)

Figure 105 An Availability Design Space



indicates an example application requiring a particular combination of immunity from planned and unplanned outage

Facets of Availability(4)

Type	Name	Description
Unplanned	Physical	Something physically wore out or break
	Design	Design errors in software and hardware
	Environmental	Loss of power or cooling
	Operator	Operator or user action: accidents, inexperience or malice
Planned	Upgrades	Software or hardware upgrades
	Maintenance	Preventative or deferred maintenance
	Regulations	Government/policy regulations
disaster	Natural	Natural disasters like hurricanes, earthquakes, or floods; or more localized “disasters” like accident tally setting off a file sprinkler system
	forced	“unnatural,” human-caused disasters such as terrorist activity

Facets of Availability(5)

- The expected causes of failures of open and commodity systems running competent vendor-supplied system S/W (informal study)
  - loss of power > application S/W > OS, Subsystem S/W, H/W with moving parts (fan, disks, tape, printer), I/O adapter > memory > CPU, cache
  - IDC survey of 200 users conducted by Harvard Research: They agreed with the ordering above.
- A major source of system outage is S/W
  - H/W doesn't break very often, but when it does, you're in really big trouble
    - ✓ longer to find the parts and repair the H/W
  - S/W failure is extremely difficult to track down and fix
    - ✓ S/W doesn't wear out and physically break
    - ✓ causes of failure are design flaws
    - ✓ unrepeatable, transient errors that occur due to a particular combination of inputs -> Heisenbugs (named after Heisenburg's uncertainty priciple)

So, OK, What Really is HA?

- Single point of failure: a single element of H/W or S/W brings down the entire computer system
  - highly undesirable for HA

- A computer-system is HA if it has two properties:
  - No replaceable piece is no single point of failure.
    - ✓ need to configure out system as clusters to satisfy the first
  - It is sufficiently reliable that you are overwhelmingly likely to be able to repair or replace a broken part before something else breaks
    - ✓ basic reliability of most system components today is high enough

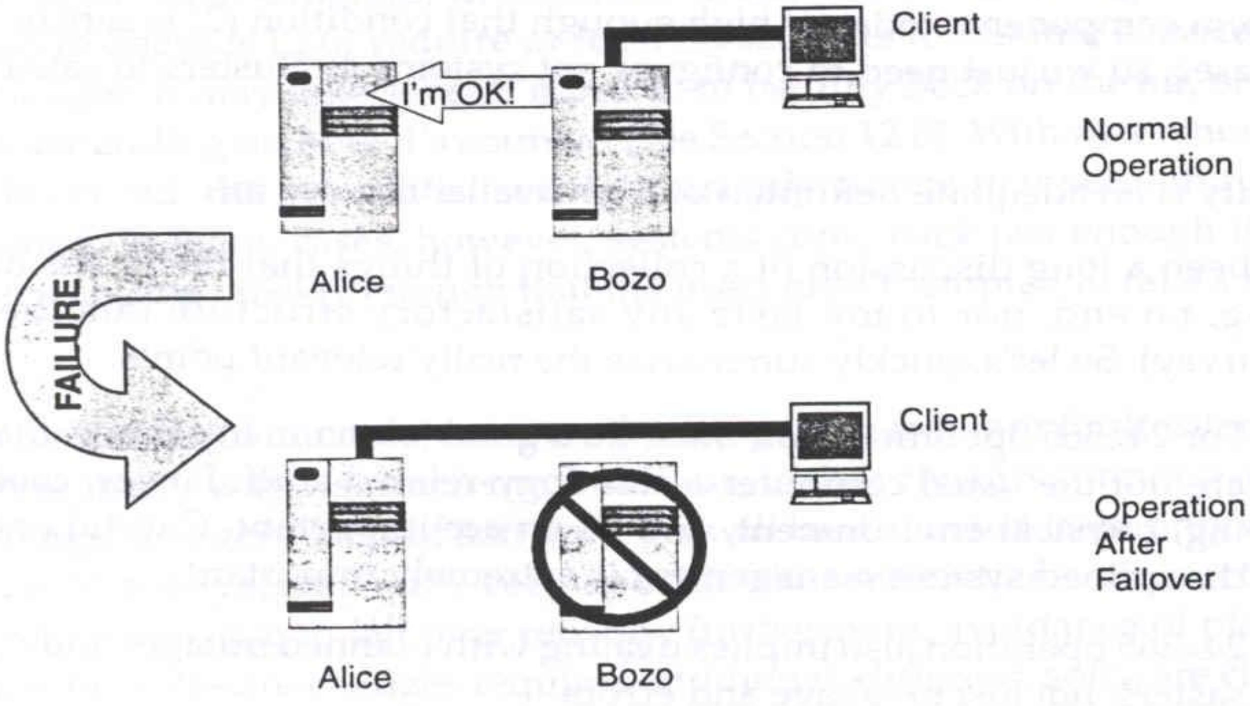
So, OK, What Really is HA?

- Summarize the really relevant points
  - For 24x365 operation, you must do a good job on many things that are not the usual computer-technology-related staff (power, cooling, etc). Careful and disciplined system management is extremely important.
  - 24x365 operation also implies dealing with planned outage and disasters, not just breakage and errors.
  - S/W causes the largest number of outages, after power failure.
  - The worst, meaning longest, unplanned outages are caused as much by H/W as S/W (again, after power failure).
  - Configure the system to avoid single points of failure
  - Clusters can help with planned outages and unplanned errors in both H/W and S/W, but don’t do so for all S/W.
  - Pure H/W-based FT “ fails over” instantaneously, but doesn’t help with S/W errors or planned outages.

The Basic Idea : Failover

- Failover is the basic technique used in cluster to achieve HA
  - One computer(Alice) watches another computer(Bozo)
    - ✓ If Bozo dies, Alice takes over Bozo’s work

Figure 106 Basic Failover



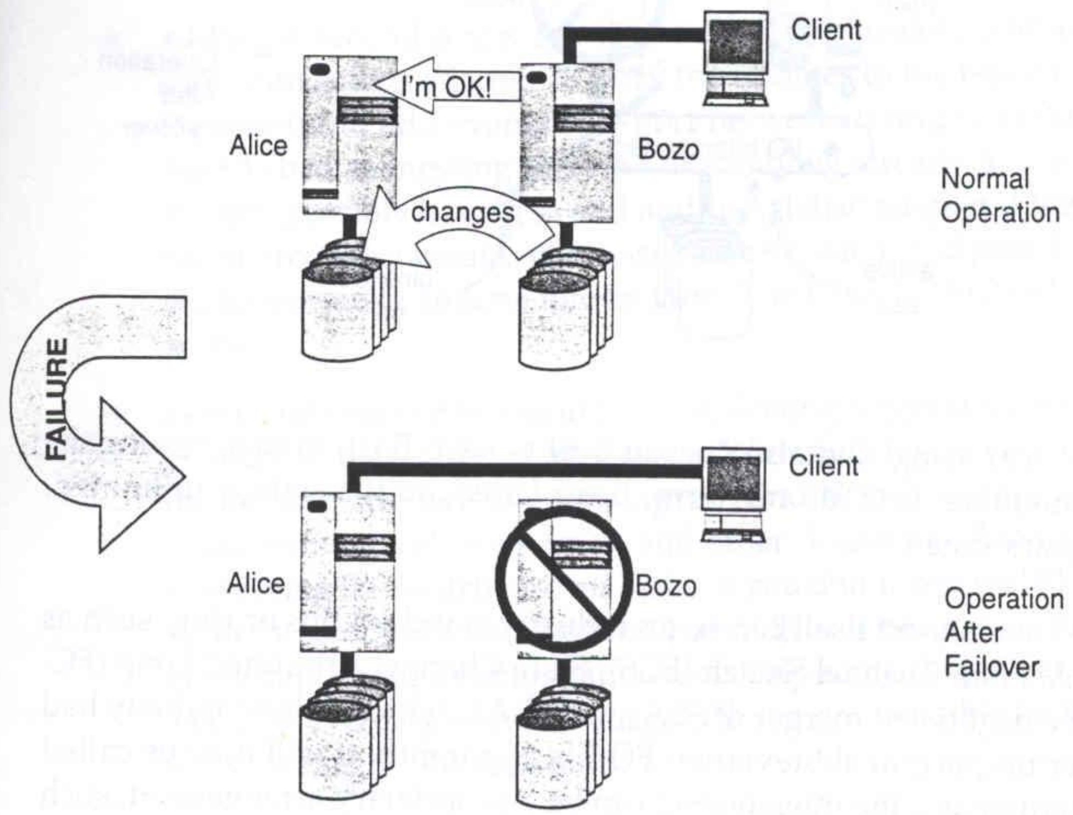
Failing Over Data

- Several issues must be confronted when a target take over data from source
  - How does the data get there?
  - Is it good data-consistent and correct?
  - How do you rebalance the workload?
  - How much time does this all take and how can it be made faster?
- Replication and Switchover
- Avoiding Toxic Data : Transactions

Replication and Switchover(1)

- Replication
  - keep its own independent copy
  - in order to be kept up to data,
    a node must ship every change
    over to another
  - if in file system,
    each write must be sent
  - if in DB, ship DB log

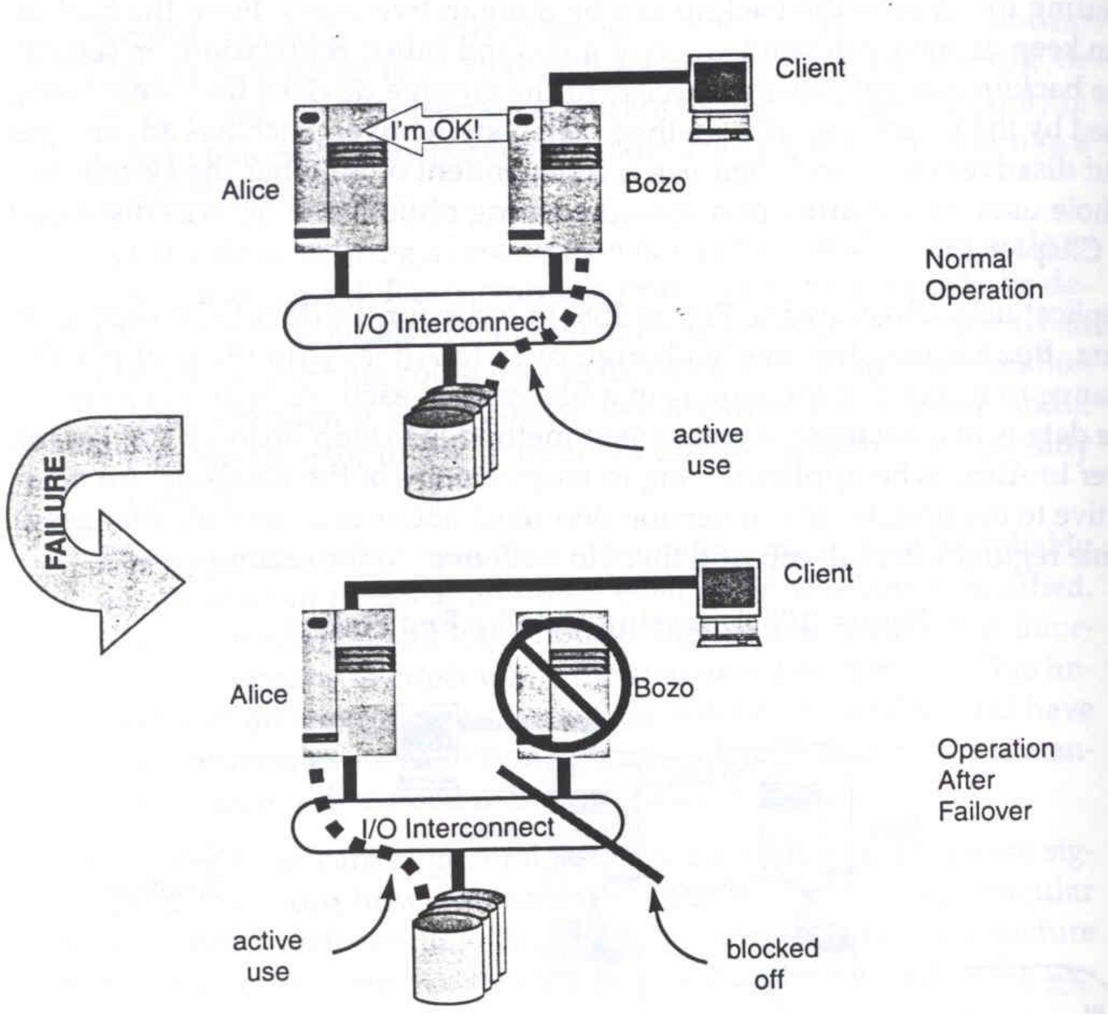
Figure 108 Accessing Data by Replication



Replication and Switchover(2)

- Switchover
  - some form of external storage interconnect provides a path to storage units from both systems
  - source failed, so it’s state is unknown
  - I/O interconnect
    - ✓ industry standard bus or ring : SCSI,
      Fibre Channel Switch(FCB),
      Fibre Channel Arbitrated Loop(FCAL),
      negotiated merger of SSA and FC-EL
    - ✓ proprietary arrangement :
      Digital’s broadcast-based Star Coupler,
      IBM’s S/390 switch-based ESCON director,
      Tandem’s ServerNet SAN
    - ✓ designed not just for failover
      but for active sharing of data

Figure 109 Accessing Data by Switchover



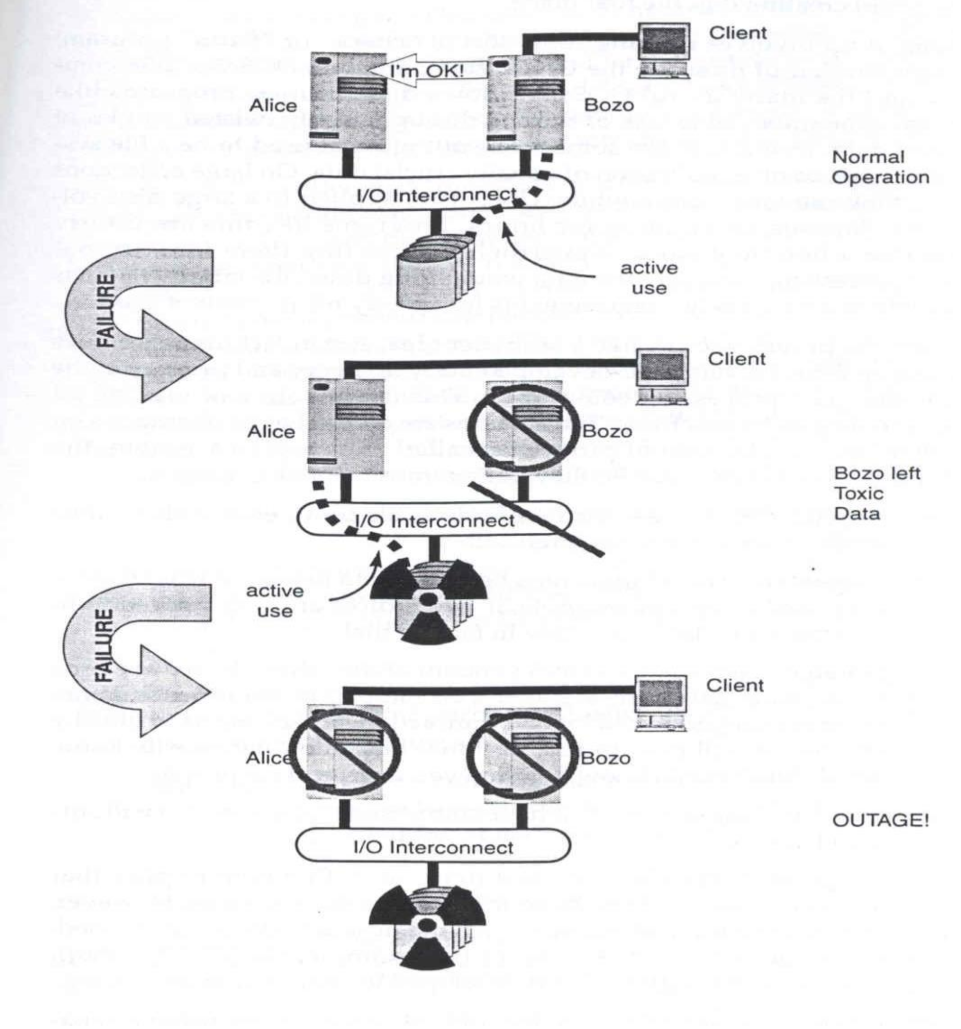
Replication vs Switchover

	Replication	Switchover
Adding a second node	easier to add	harder to add; alter existing cabling
Synchronizing configuration	easier to configure; systems are fairly independent	harder to configure; disk configuration must be synchronized
distance	can be a distance apart	nodes must be physically close and disks are in one place
adapter/controller issues	can use old I/O adapters and controllers	actually implement their specification
storage units	can use simple storage unit	must use hardened storage such as RAID
one-to-many backup	limited on the how busy the backup unit become	limited only by how many systems can be attached to the same interconnection
duplicate storage	another copy of storage	share a single copy of storage
overhead	use CPU and I/O capacity in normal operation	no overhead in normal operation
synchronization	tight synchronization causes performance loss; loose synchronization can lose data at failure	natural complete synchronization without performance loss
failback	present new problems of re-synchronization	no different from failover

Toxic Data

- Toxic data syndrome
  - Alice now has physical access
    to the data which was last written by Bozo
    who is dead
  - Bozo may well have been deranged
    before kicking the bucket
  - Alice crashes?

Figure 110 Toxic Data Syndrome



Adding Toxic Data : Transactions(1)

- Two way avoid this problem
  - clean it up
    - ✓ running some sort of “ check” or “ fixup” program : fsck, 찬아 (running for a long time)
  - avoid creating it => make all changes to data as transactions
- Transaction
  - defined to be operations on data that have a collection of properties taken together (ACID properties)
  - ACID defines transactional semantics (no implementation)
    - ✓ Atomic : changes are made as indivisible unit
    - ✓ Consistent : the relationships between data items are what they’re supposed to be
    - ✓ Isolated : each change is independent of the others
    - ✓ Durable : transactions that have completed will survive failures
  - there are wide variety of techniques for transaction processing

Adding Toxic Data : Transactions(2)

- Log file : sharing
  - a separate file on stable storage used to implement transactional semantics
  - sequentially written record of everything done to the data:
    the values before it was changed, with a unique id of this change; then the values after it is changed, with the same id; and then, crucially, as a single atomic write, a short item with identifier and the notation “did it”. (committed)
  - application doing the changes has to note when it’s starting the change
  - consistent data can be recreated by scanning through log
- 2-phase commit : shared-nothing
  - phase one : gathers commitment from everyone
  - phase two : tells them all to complete
- Checkpointing
  - backup to the last consistent state

Failing Over Communications

- IP takeover
  - Alice (target) resets one of its communication adapters to respond to the IP address that Bozo(source) was using
  - Lost session information
  - Re-log-on of several thousands of user => time consuming work
- Solutions
  - persistence sessions on the server
    - ✓ session state data is kept on stable storage and updated with transactional semantics
  - intelligence of client workstation
    - ✓ keeps session data on the client side (cookie, Lotus Notes)
    - ✓ holds the communication address of multiple servers

HEARTBEAT 등

Heartbeats, Events, and Failover Processing

- Heartbeat
  - realtime task with hard, externally imposed deadlines
  - In a conventional OS, heartbeat send and receive processes can be pinned into dedicated memory that's never swapped out
- Well designed heartbeat subsystem
  - Get maximum information about a cluster's health by running multiple heartbeats that couple the cluster's nodes through every possible way
    - ✓ normal communication (LAN)
    - ✓ all the cluster acceleration gear available (Myrinet)
    - ✓ slightly less normal paths (RS-232 links)
    - ✓ even signals sent across the I/O interconnect from node to node (SCSI)

Heartbeats, Events, and Failover Processing

- Heartbeat has two implications
  - somebody knows what all those orifices are, to say nothing of how they're connected and what all the nodes are (resource information)
  - the immediate response of the system to loss of a heartbeat operation is wait. The first heartbeat loss dose not provide enough information to conclude that a node is down.
- Funeral rite sequence
  - establish a new heartbeat chain that excludes Bozo
  - inform parallel subsystems that were running on Bozo, such as database, of what has occurred and is about to happen
  - fence Bozo off from its resources (e.g. disks)
  - form a cluster-wide, consistent plan defining Bozo's resources and workload should be redistributed among the remaining nodes
  - execute the plan : actually move the resources
  - inform subsystems that the resource reallocation process has been completed
  - resume normal operation

Heartbeats, Events, and Failover Processing

- Rejoin (Bozo comes back to life)
  - initiated by a message from Bozo to an advertised port listening for such message
  - the step fencing Bozo off is not performed.

- Liveness check
  - active system resources are sent messages, thus giving an indication that the resource is still functional.
    - ✓ Are you awake?
    - ✓ I am awake. (heartbeat)
  - response time to a liveness check can vary radically

Link to Scaling : Cost

- Scaling makes availability affordable
  - The more nodes there are in a cluster, the less you pay for HA

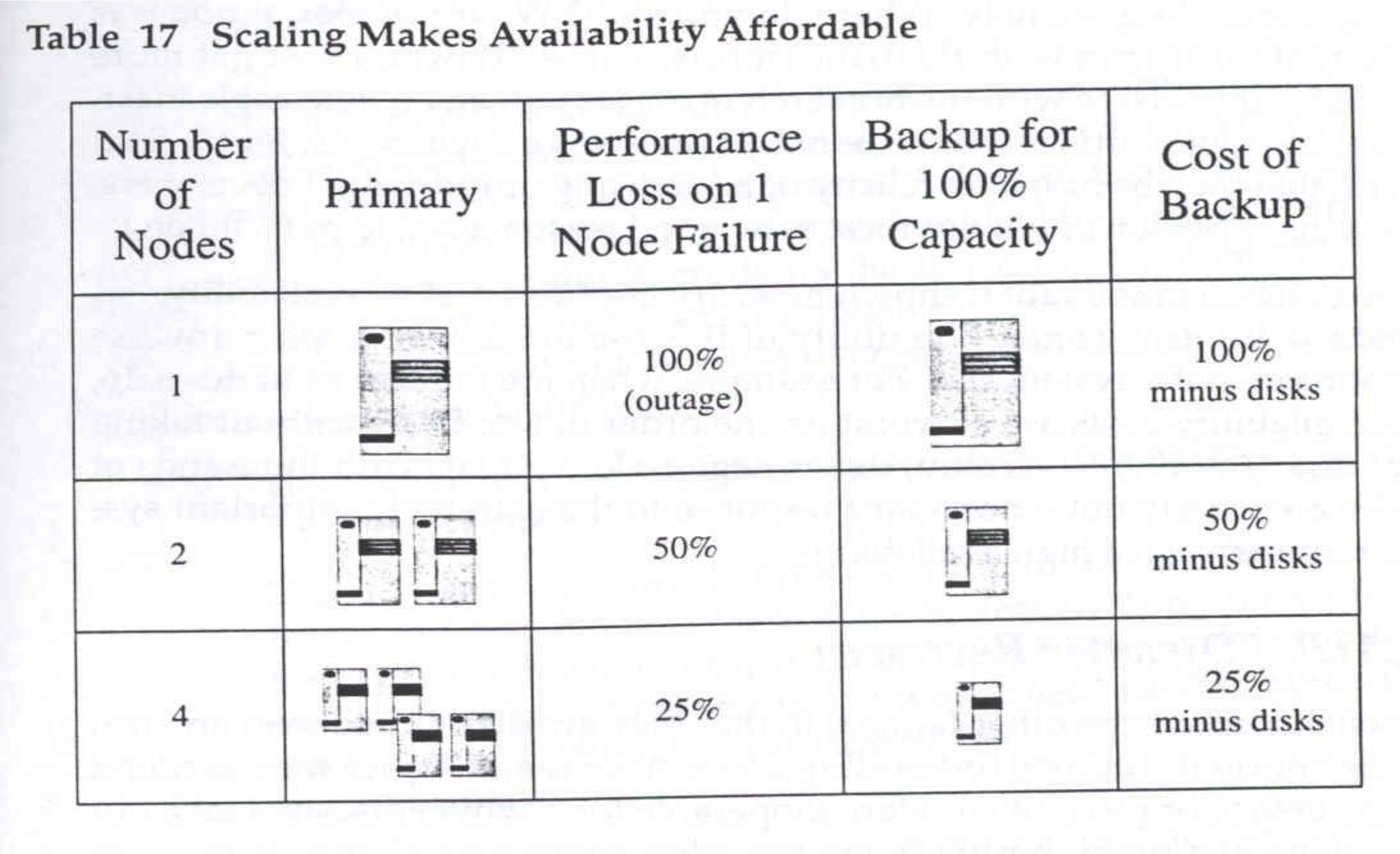


Table 17 Scaling Makes Availability Affordable

Number of Nodes	Primary	Performance Loss on 1 Node Failure	Backup for 100% Capacity	Cost of Backup
1		100% (outage)		100% minus disks
2		50%		50% minus disks
4		25%		25% minus disks

Link to Scaling : Cost

- Scaling makes availability affordable
  - As the number of nodes increased, performance loss on node failure becomes less and less severe
  - the cost of backup facilities to maintain 100% capacity becomes less and less
  - cost is substantially less

- Different target for scaling effect
  - human hunger for power : no defensible maximum
  - cost of availability: The utility of scaling drops in inverse proportion to the system size.

DISASTER RECOVERY

Disaster Recovery

- With disaster-caused outages of less than six days, 25 % of the companies studied immediately went bankrupt, 40% were bankrupt within 2 years, and less than 7% were still in business after five years

  -> Disaster can really hurt in businesses

- Disaster recovery
  - data replication (backup)

Disaster Recovery

- seven tiers of disaster recovery
  - Tier 0 obviously contains nothing worth discussing
  - in Tier 1, backup all data, load it on a truck, and take it somewhere else
  - Tier 2 is like Tier 1, except that you drive the truck to your own duplicate system site
  - in Tier 3, replace Tier 2’s pickup truck with wires
  - Tier 4, active secondary, send data electronically to a duplicate site
  - Tier 5 makes sure the secondary is continuously up to date by doing 2-phase commit for key data
  - Tier 6 is the highest tier, ultimate zero data loss situation

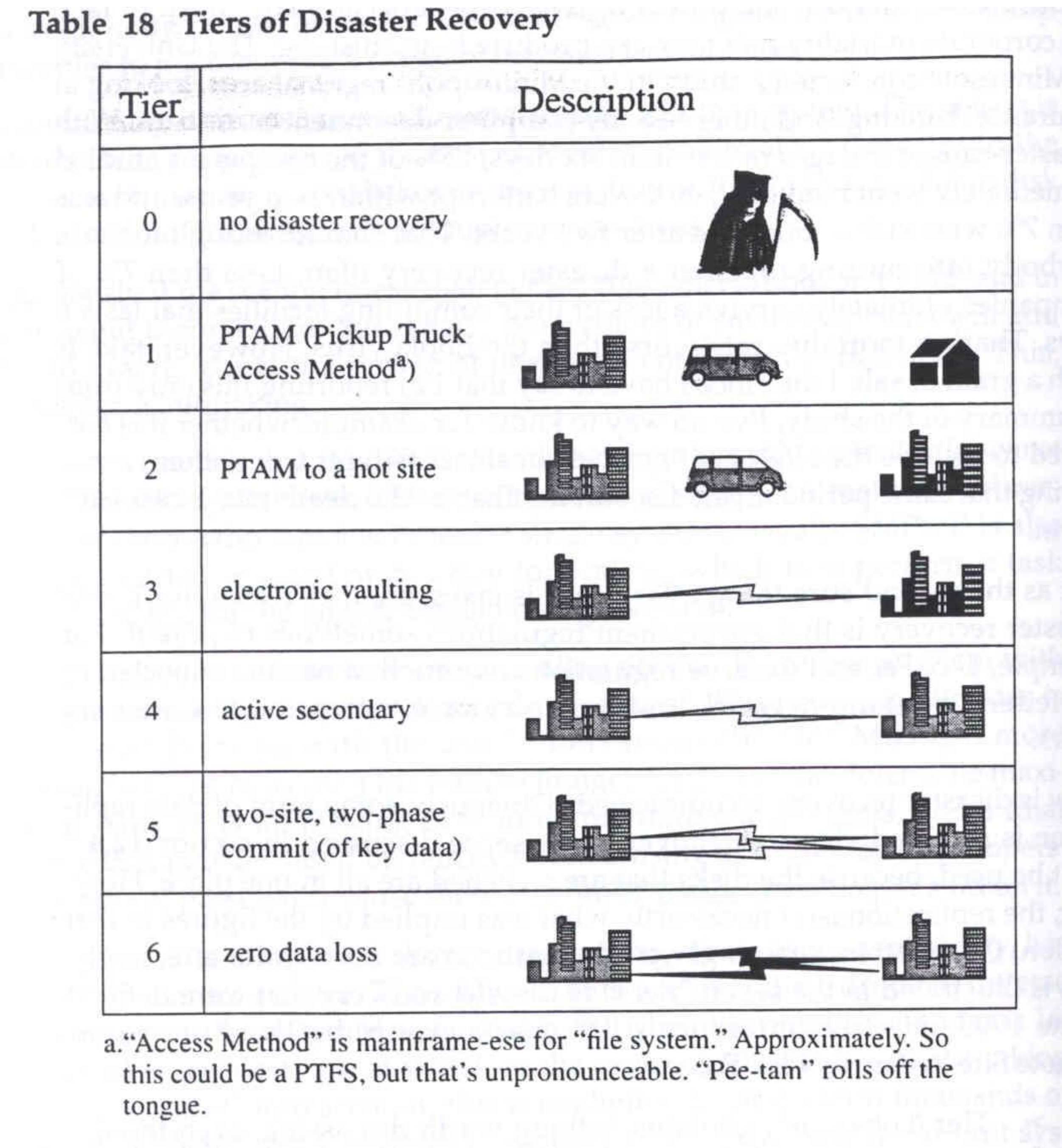


Table 18 Tiers of Disaster Recovery

Tier	Description
0	no disaster recovery
1	PTAM (Pickup Truck Access Method[a])
2	PTAM to a hot site
3	electronic vaulting
4	active secondary
5	two-site, two-phase commit (of key data)
6	zero data loss

a. "Access Method" is mainframe-ese for "file system." Approximately. So this could be a PTFS, but that's unpronounceable. "Pee-tam" rolls off the tongue.

Disaster Recovery

❖ divide the possibilities for Tiers 5 and 6 into 3 cases :

- 1-safe : primary site commits all transactions. After committing each, it asynchronously writes the log to the backup site
- 2-safe : If backup site is operational, each transaction's log records are written to the backup before committing the transaction. The primary commits only after getting a response that the log was received
- very safe : literally performs 2-phase commit between the sites

- DB vendor: 1- or 2-safe
- EMC, IBM: geographical mirroring (1-safe)

- **Netflix : chaos monkey**

CLOUD COMPUTING

클라우드 컴퓨팅 개요

1. 컴퓨팅 자원 소유 방식의 변화
   1) 기업 내 IT 자원 및 서비스의 아웃소싱 확대
   2) 분업화와 규모의 경제 실현
2. 인터넷 기반 서비스의 확대
   1) SW와 컨텐츠의 온라인 서비스화
   2) 초고속망을 통한 안정적인 서비스 전송 가능
3. 클라우드 컴퓨팅
   1) 모든 소프트웨어 및 데이터는 클라우드(IDC 등 대형컴퓨터)에 저장되고 네트워크 접속이 가능한 PC나 휴대폰, PDA 등의 다양한 단말기를 통해 장소에 구애 받지 않고 원하는 작업을 수행할 수 있는 컴퓨팅 기술
   2) 사용자는 서버, 디스크, 소프트웨어 등을 임대해서 사용하고 사용한 만큼의 요금을 서비스 회사에 지불하는 컴퓨팅 사용방식
   3) 클라우드(Cloud)라는 명칭은 IT 아키텍처 다이어그램에서 인터넷을 구름으로 표현하던 것에서 유래

클라우드 컴퓨팅 필요성

1. **Cloud 도입은 IT 리소스의 탄력적 사용으로 인한 TCO절감은 물론 Process 혁신으로 신속한 사업 추진 (Time-to-market) 가능**

1) 데이터 폭증- 빅데이터의 출현

- ✓ SNS와 Smart Phone 의 대중화로 개인 무선 데이터의 폭발
- ✓ 사물이 인터넷에 연결되는 IoT (Internet Of Things) 성장

2) 모바일 기기의 다양화 및 활성화

- ✓ Tablet PC와 NFC 등 탑재한 스마트폰, 다양한 모바일 단말의 등장
- ✓ Thin client를 넘어 Zero client 출현 – VDI, 게임 기기와 Connected TV

3) Cloud Streaming – 음악, 게임, 비디오 처리 고성능 SW가 서버에서 실행

- ✓ Content 소비가 소유 -> 접속, Streaming 기반 on-demand services

4) 녹색 성장의 사회적 정책 및 산업의 핵심 가치에 부합

- ✓ 저탄소, 고효율의 Green IT를 위해 기업 자체 IT투자 및 운용 최소화

클라우드 컴퓨팅 필요성

1. 기존 공급자 중심 방식에서 사용자 중심의 서비스 제공방식으로 변화
   1) 위치에 무관한 자원 공동 사용(가상화)
   2) 어디서나 연결 가능한 인터넷
   3) 온 디멘드(on demand) 셀프 서비스
   4) 신속한 탄력성(elasticity)
   5) 사용한 만큼 지불

클라우드 컴퓨팅 아키텍쳐

1. 서비스 유형 (서비스의 종류에 따른 분류)

서비스 유형	정의
IaaS (Infrastructure as a Service)	서버, 데스크탑 컴퓨터, 스토리지 같은 IT 하드웨어 자원을 클라우드 서비스로 빌려쓰는 형태를 말한다. 예) Amazon EC2, S3
PaaS (Platform as a Service)	소프트웨어 개발자들이 자유롭게 머물며 자신이 원하는 소프트웨어를 구현할 수 있도록 지원한다. 이는 응용 소프트웨어를 제작하기 위한 도구인 프로그래밍 언어를 제공하는 수준을 넘어서 미들웨어까지 포괄하는 개발 플랫폼을 제공한다. 예) 구글의 Apps 등
SaaS (Software as a Service)	클라우드 컴퓨팅 서비스 사업자가 인터넷으로 소프트웨어를 제공하고, 사용자가 인터넷에 원격으로 접속해 소프트웨어를 활용하는 모델이다. 소프트웨어를 주문형 서비스 형태로 제공하는 것으로, 같은 소프트웨어를 여러 고객이 공유해서 사용할 수 있도록 한다. 예) Salesforce.com, MS Office Live 등

클라우드 컴퓨팅 아키텍쳐

1. 서비스 유형 (resource의 위치 및 관리에 따른 분류)

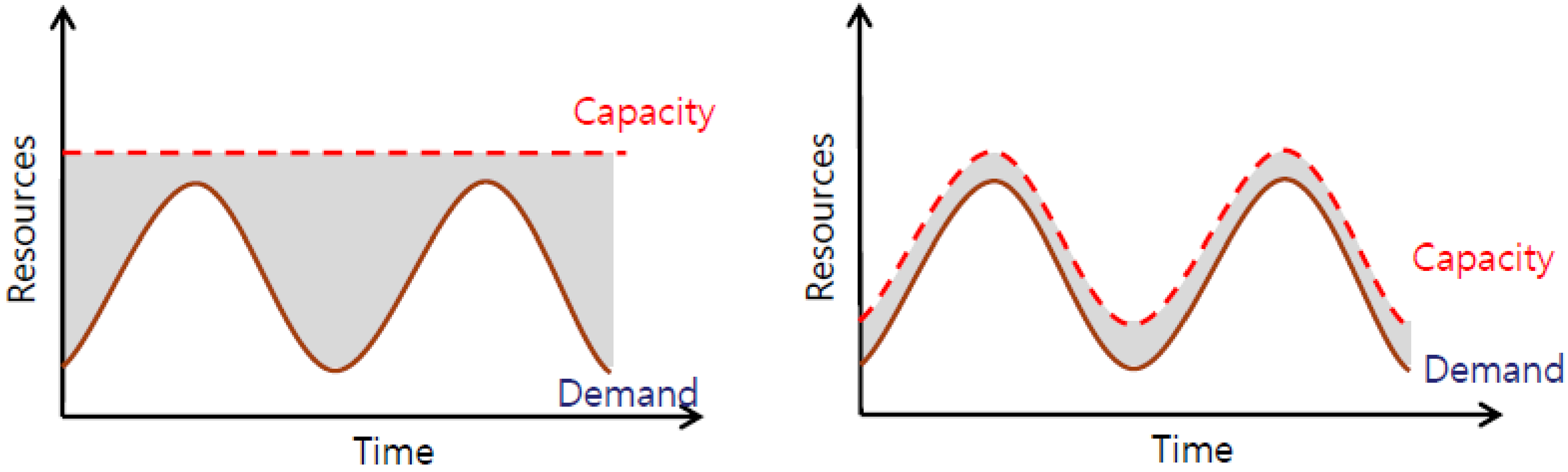
서비스 유형	정의
퍼블릭 클라우드 컴퓨팅	불특정 다수(일반개인 또는 조직)를 위한 클라우드 시스템 시설 소유 및 관리운영 주체는 서비스 제공 사업자
프라이빗 클라우드 컴퓨팅	특정조직 전용의 클라우드 시스템 관리운영 주체에 따라 직접 또는 제3자 관리운영으로 나누어짐 설치장소에 따라 조직내 또는 조직외로 나누어짐
하이브리드 클라우드 컴퓨팅	2개 이상의 클라우드 시스템이 유기적으로 연계되어 존재 전체가 하나의 시스템처럼 운용되며, 개별 시스템 간에는 표준화된 기술로 연동되고, 데이터나 응용서비스의 이동성이 확보되어야 함. 내부 중요 파일은 프라이빗 클라우드에 보관하고 그렇지 않은 데이터는 외부 데이타센터를 이용

클라우드 컴퓨팅 아키텍쳐

1. 클라우드 컴퓨팅 기술 및 서비스 모델
   1) 가상화 기술
      - ✓ 컴퓨팅 자원 가상화, 스토리지 가상화, 네트워크 가상화
   2) 관리 기술
      - ✓ 로드밸런싱, 고가용성 기술, 복구 기능, SLA
   3) 보안 기술
      - ✓ Single sign on, 다중임차자 관련 보안 기술
   4) 분산 기술
      - ✓ 그리드 컴퓨팅, 분산 데이터베이스, 분산 파일 시스템
   5) 서비스 모델
      - ✓ SaaS, PaaS, IaaS
   6) 빌링 및 프로비저닝
      - ✓ 유틸리티 컴퓨팅

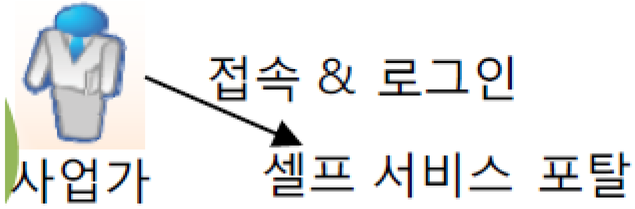
클라우드 컴퓨팅 장점

1. 사용량에 따라 IT 자원을 즉시 확장/축소하고, 사용량에 따른 과금

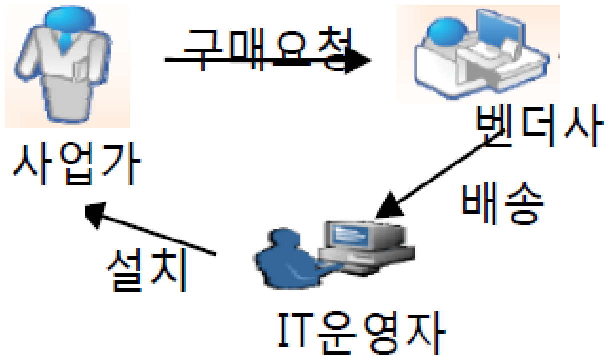


2. IT관리자의 간섭 없이 사용자가 직접 용이하게 구매, 설계, 설치 가능

웹을 통한 절차



여러 관계자를 통한 절차



클라우드 컴퓨팅 장점

1. 고객(Customer)
   1) 저비용
   2) 효율적인 운영환경
   3) 막강한 컴퓨팅 파워
   4) 필요할 때 즉시 사용

2. 서비스 제공자
   1) 규모의 경제 실현
   2) 미래성장산업을 통한 매출증대
   3) Lock-in & Retention
   4) Cloud 시장 주도