



KT ucloud biz를 활용한 서비스별 클라우드 구축 가이드

How to design and build your business
infrastructure on KT cloud

김인정, 김경화 지음





KT ucloud biz를 활용한 서비스별 클라우드 구축 가이드

How to design and build your business
infrastructure on KT cloud

김인정, 김경화 지음

지은이_ 김인정(pypupipo@gmail.com)



성신여자대학교 컴퓨터정보학부를 졸업하고, 포항공과대학교 시스템 소프트웨어 연구실과 미국 Columbia University에서 석사 과정을 마쳤다. 현재는 Pennsylvania State University 네트워크 연구실에서 박사 과정을 밟으며 IBM Watson 연구소에서 인턴으로 일하고 있다. 안철수 연구소에서 윈도우 커널에서의 게임과 온라인 뱅킹 보안 모듈 개발하였고, KT Cloud 플랫폼 본부에서 KT ucloud biz 클라우드 상품에 대해 국내외 기업을 대상으로 기술컨설팅과 연구, 그리고 Open API 활용 기술에 대한 지원을 담당하였다. 이동성을 지원하는 운영체제 기법에서부터 각종 클라우드 및 네트워크 분야의 논문, 특히, E-book, 온/오프라인 강의 수행 및 클라우드 컴퓨팅 관련 번역 작업도 진행하였다.

지은이_ 김경화(desp99@gmail.com)



KAIST 전산학과 학부를 졸업하고, 국내 삼성전자 무선사업부에서 일하였으며, 미국 Columbia University에서 석사 및 박사 과정을 마쳤다. 국제 학회에 무선인터넷, 클라우드 컴퓨팅, Software-defined networking 등 다양한 네트워킹 분야의 논문을 여러 편 발표하였다. 미국 벨 연구소와 IBM Watson 연구소에서 인턴으로 일하였으며, 현재는 Google 뉴욕 지사에서 소프트웨어 엔지니어로 일하고 있다.

KT에서 클라우드 관련 연구 개발과 기술 컨설팅을 하면서 터득한 지식을 다른 사람들이 좀 더 쉽게 접할 수 있길 바라는 마음에서 못매를 맞을 각오를 하고 펜을 들게 되었다. 국내 소프트웨어와 IT 기술이 더욱 널리 사랑받고 IT 업계 종사자들이 국내외에서 활약할 수 있도록 작게나마 도움이 되었으면 하는 바람이다.

이 책이 세상에 나오기까지 도움을 주신 한빛미디어 정지연 과장님과 스마트미디어팀에게 감사하다. 또한, 저술에 많은 도움을 주신 KT 송순용, 한석웅, 유필준, 원유훈, 송태연, 박종진, 백두현 님과 엔클라우드 24의 맹상영 사장님께 감사의 마음을 전한다. 저자의 지적 성장에 많은 도움을 주신 성신여자대학교 홍의석, 우종정, 유원경, 이재원 교수님, 포항공과대학교 박찬익 교수님, IBM 연구소 및 Columbia University의 Sambit Sahu 교수님, Pennsylvania State University의 Thomas La Porta 교수님께도 감사의 마음을 전한다.

마지막으로 언제나 큰 힘이 되어 주신 사랑하는 부모님과 언니 수정, 동생 효정, 그리고 남편에게 이 책을 바친다.

- 펜실베이니아의 공기 좋은 산속 마을에서 저자 김인정

클라우드 컴퓨팅^{Cloud Computing}은 차세대 컴퓨팅 자원의 유틸리티 서비스로서, 누구나 한 번쯤 들어보았거나 알게 모르게 이미 클라우드 기술을 사용하고 있을 것이다. 물론 모두가 클라우드 컴퓨팅의 상세 기술이나 특징점에 대해 깊이 있게 파악하기는 어렵다. 따라서 국내 사용자와 기업이 클라우드 환경을 도입하는 현시점에서 각자의 비즈니스 모델에 따라 클라우드 컴퓨팅을 언제 어떻게 사용할 수 있는지에 대한 구체적인 가이드 라인이 필요하다.

현재 시점에서 볼 수 있는 클라우드 컴퓨팅 서적은 대부분 클라우드 컴퓨팅에 대한 일반적인 개념을 소개하거나 미국 아마존 사의 클라우드 서비스인 아마존 웹 서비스^{AWS, Amazon Web Service}에 대해 기술하고 있다. 하지만 이 책에서는 AWS와 KT ucloud biz 서비스를 비교하여 국내 시장에서 KT ucloud biz가 우세한 점을 거론하고, 효율적인 아키텍처 방법에 대해 안내하고자 한다.

필자는 다년간 SaaS^{Software As A Service} 기술부터 IaaS^{Infrastructure As A Service} 기술에 이르는 연구와 상품 개발 및 기술 컨설팅에 집중하면서, 프라이빗 클라우드^{Private Cloud}를 고려하는 사업자와 퍼블릭 클라우드^{Public Cloud}를 이용하려는 기업의 엔지니어에게 좀 더 나은 방안을 안내하였다. 또한, 특정 서비스를 운영하기 위해 필요한 최적화 방법론부터 클라우드 서비스를 사용할 때 고려해야 할 점까지 여러 방면에 대해 연구해 왔다. 그 결과, 중복적이고 당연하게 논의되는 부분을 찾았으며 이를 공유해야겠다고 느꼈다. 물론 시시각각으로 변하는 컴퓨팅 기술에서 클라우드 컴퓨팅이라는 기술의 유효성이 얼마나 지속될지, 국내 최대 클라우드 사업자인 KT ucloud biz 서비스가 언제까지 유효하게 될지는 가늠하기 어렵다. 그렇지만 클라우드 컴퓨팅 기술의 활용 방법에 대해 더 구체적이면서도 포괄적으로, 상용 시스템에 맞도록 설명하여 많은 사람들의 궁금증이 해소되었으면 하는 바람이다.

이 책에서 다루는 내용

우선 KT 클라우드에 대한 주요 사항과 도입 방법에 대해 다루고, 서비스 종류에 따른 설계 방법과 성공 사례에 대해 소개한다. 이는 KT 클라우드를 사용하는 대기업, 공공기관, 모바일 게임 업체, 리셀러 업체에게 기술 컨설팅을 진행하면서 실시하였던 테스트, 팁, 구성 방안을 기반으로 설명하므로 실용적이고 심도 있는 내용이라 할 수 있다. 이 내용에는 KT 클라우드 매뉴얼이나 홈페이지에 공개되지 않은 내용도 있다. 또한, Open API를 사용하여 활용하는 방법에 대해 소개하여 리셀러, 오픈소스 사용자와 클라우드 자원 구성의 자동화에 관심 있는 사용자에게 용이한 정보를 제공하고자 하였다.

1부에서는 클라우드 컴퓨팅의 이론과 국내외 클라우드 제공자의 현황 및 서비스에 대해 다룬다. 1장에서는 클라우드 컴퓨팅의 기본 개념에서 사용되는 주된 자원을 알아본 뒤, 클라우드 환경에서의 아키텍처 설계, 구축, 운영 방법을 설명한다. 2장에서는 KT ucloud biz 서비스를 이용한 서비스별 아키텍처 구축 과정을 소개한다. 사용자가 가장 많고, 상세한 기능이 소개되어 있는 ucloud server에 대해 알아보고, Amazon EC2와 비교해 본다. 또한, ucloud 스토리지 및 CDN, 각종 응용 서비스에 대해 알아본다.

2부에서는 각각의 비즈니스 모델에 맞는 클라우드 아키텍처 설계 방법을 다룬다. 실제로 비즈니스가 안정 지향적인지 속도 지향적인지 확장/축소가 급변하는 서비스를 구축해야 하는지 등에 따라 적절한 아키텍처 설계 가이드를 살펴보고, 클라우드 자원에 자신의 IT 비즈니스를 수행하는 것이 아닌 리셀링하는 사업에 대해서도 알아본다.

마지막으로 3부에서는 클라우드 컴퓨팅의 관리 자동화 방법과 빅데이터^{Big Data} 기술을 클라우드 서버에 적용하는 방안을 설명한다. 웹 포털을 사용한 관리가 아닌 사용자의

요구에 맞게 Open API로 다양하게 커스터마이징된^{customized} 관리 방법을 소개하고, 하둡과 맵리듀스, 몽고디비, 도커와 같은 최신 빅데이터 기술을 클라우드에 적용하는 방법을 알아본다.

이 책의 대상 독자

대기업, 공공기관, 모바일 게임사, 클라우드 관련 연구자와 같이 기존에 KT ucloud를 사용하는 클라우드 운영자부터 ucloud biz에 대해 관심 있는 사용자, 자신의 서비스에 맞게 클라우드 환경을 운영하려는 관리자, 아마존 웹 서비스와 KT ucloud biz를 비교/검토하고 싶은 IT 업계 종사자, KT ucloud biz를 사용하여 리셀링을 하고 싶은 사업자, KT ucloud를 도입하여야 하는 KT 내부 직원까지 폭넓게 활용할 수 있도록 엮었다. 다만, 이 책을 좀 더 잘 이해하기 위해서는 인터넷과 네트워크, 컴퓨터에 대한 배경지식이 있어야 한다.



한빛 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾기도 쉽지 않습니다. 또한, 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1 eBook First - 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 좀 더 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한, 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

무료로 업데이트되는 전자책 전용 서비스입니다

2 종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3 독자의 편의를 위해 DRM-Free로 제공합니다

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 어려운 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유 권한을 표시한 문구가 없거나 타인의 소유권함을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 큼니다. 이 경우 저작권법에 따라 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한, 한빛미디어 사이트에서 구매하신 후에는 횡수에 관계없이 내려받을 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구매하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허락되지 않습니다.



Part 1

클라우드 컴퓨팅

Cloud Computing

일반 서버와 클라우드의 차이는 무엇일까? 또한 클라우드 도입을 위해서는 어떠한 절차가 필요한가? KT ucloud biz 서비스의 특징점은 어떠한 것이 있으며 아마존 AWS와 비교하였을 때 어떠한 차이점이 있을까? 1부는 이러한 질문에 대한 답으로, 1장에서는 클라우드 컴퓨팅의 정의, 클라우드 컴퓨팅에 사용되는 자원을 살펴보고, 아키텍처의 설계, 구축, 운영 방법에 대해 다루겠다. 2장에서는 이 책에서 주로 다룰 KT ucloud biz의 주요 서비스를 알아본다.

클라우드 컴퓨팅 도입에 유용한 개념

Essential Concepts and Principles to Adopt Cloud Computing

이 장에서는 클라우드 컴퓨팅이 무엇인지 정의하고, 클라우드 컴퓨팅 도입에 필요한 핵심 기술과 현황을 다룬 다음 아마존 웹 서비스^{AWS}와 KT ucloud biz를 비교한다.

1.1 클라우드 컴퓨팅

컴퓨터 전공자가 아닌 사람에게 클라우드 컴퓨팅 기술은 새로 등장한 기술처럼 여겨진다. 그러나 컴퓨터 과학 분야에서 클라우드 컴퓨팅 기술은 그리 새로운 기술이 아니다. 이는 기존에 있는 컴퓨팅 자원에 대한 가상화^{Virtualization} 기술이 응용된 형태로 볼 수 있다. VMware나 Oracle VirtualBox, Microsoft Virtual PC 등의 제품이 운영체제를 가상화하는 기술이었다면 아마존 웹 서비스나 KT ucloud biz와 같은 클라우드 컴퓨팅은 물리적인 컴퓨팅 자원을 가상화하는 기술이라 할 수 있다.

클라우드 서비스는 기본적으로 IT 자원의 사용 여부, 사용량, 시간에 따라 과금^{Pay per use}한다. 그래서 물리적으로 시스템을 구축해서 서비스를 운용할 경우 구성한 사양보다 높은 사양이 필요하거나 자원의 추가 확장이 필요할 때 주로 클라우드 서비스를 이용한다. 또는, 반대로 대부분의 자원을 사용하지 않고 방치하는 경우

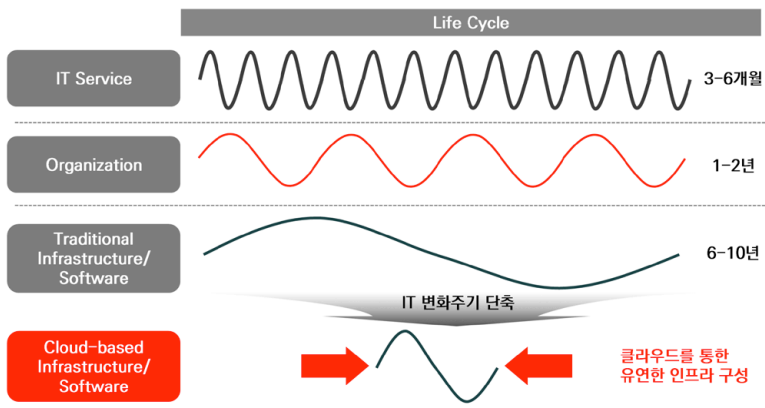
에 이용하기도 한다. 실제로 컴퓨팅 자원의 유지보수와 24시간 운용, 그리고 업그레이드에 따른 소요 비용을 따졌을 때 클라우드로 전환하는 것이 효율적이기 때문이다. 우선, 클라우드 서비스는 기민성이 뛰어나다. 다시 말해, 원하는 사양의 클라우드 자원을 선택하여 사용하기까지에 필요한 배송, 설치 및 부가적인 고려 사항(쿨링 및 서버실 증축 등)이 생략되고 수 분 내에 사용할 수 있다. 또한, 실 사용량과 시간에 따라 과금되기 때문에 사용하지 않는 유휴자원을 최소화하므로 IT 인프라 소유 비용(TCO: Total Cost of Ownership)을 줄일 수 있다. 다음은 클라우드 자원 사용의 특성이다.

- **기민한 탄력성**Rapid Elasticity: 필요에 따라 자원의 확장 및 축소를 처리할 수 있는 능력을 말한다. 사용자가 자원을 필요한 수준으로 마음대로 확장하거나 줄일 수 있어야 한다. 이런 작업을 수행하는 데 수일 이상 소요되는 것이 아니라 수 분에서 수십 분 이내로 작업할 수 있어야 한다.
- **온디맨드 셀프서비스**On-Demand Self-Service: 사용자는 클라우드 서비스 관리자의 개입 없이 원하는 시점에 서비스를 바로 사용할 수 있다. 또한, 사용하는 클라우드 자원의 양과 시간에 따라 과금하게 된다.
- **측정**Measured 가능한 서비스: 자원의 사용량이 실시간으로 수집되고 모니터링된다. 이 정보를 이용하여 사용량에 따라 과금할 수 있으며 여유 자원의 부족에 따른 자원 추가 증축 등을 하여 사용자가 원하는 시점에 원하는 만큼의 클라우드 자원을 사용할 수 있게 한다.
- **유비쿼터스 네트워크 접근**Ubiquitous Network Access 방식: 클라우드 서비스 제공자는 네트워크 기반으로 서비스에 접속하도록 제공하여 다양한 클라이언트로 접속할 수 있다.
- **자원 풀링**Resource Pooling 서비스: 물리적 또는 가상화된 자원은 풀로 관리되며 사용자의 요청에 의해 사용자에게 할당되거나 다시 풀로 반환된다. 사용자는 자원의 물리적인 위치, 크기 등에 대해서는 신경 쓰지 않게 자원을 추상화시켜 제공한다.

스마트폰, 태블릿, 노트북과 같은 모바일 단말기가 급증함에 따라 데이터가 폭발적으로 증가하고 IT 서비스의 수명(Lifetime)이 3~6개월 단위로 단축되고 있어, 조직적으로 움직이는 전통적인 IT 시스템 구축 방법으로는 이러한 트렌드를 따라가기

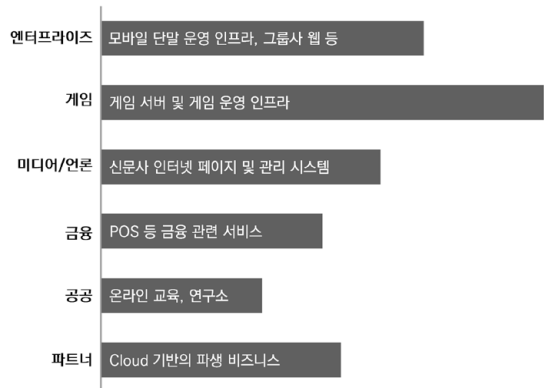
버겁다. 클라우드 인프라를 도입하여 유연한 인프라를 구성하면 투자 리스크를 줄일 수 있고 신속한 사업화가 가능해진다. 안정적으로 자원을 자동 확장할 수 있고 TCO를 절감하여 경쟁력이 높아지는 장점이 있다. 구체적으로 설명하면, 과도한 초기 투자를 방지하고 사업 종료 시의 투자 손실을 줄일 수 있다. 시장에 진출할 때 즉각적인 IT 시스템 구축이 가능하고, 돌발적인 자원 수요에 따라 신속하게 대응할 수 있으며, 기존의 IT 시스템 구축 방법과 비교하여 운용비를 30% 정도 절감할 수 있어 본업의 경쟁력에 집중할 수 있게 된다.

그림 1-1 IT 서비스의 변화 주기 단축



2014년 기준으로 클라우드 사용자를 살펴보면 [그림 1-2]와 같이 IT 자원의 유연하고 신속한 대응이 필요한 게임사와 엔터프라이즈가 1, 2위를 다투고 있고, 그 뒤로 미디어, 언론, 금융, 공공기관, 파트너사가 있다.

그림 1-2 다양한 업종의 클라우드 사용자

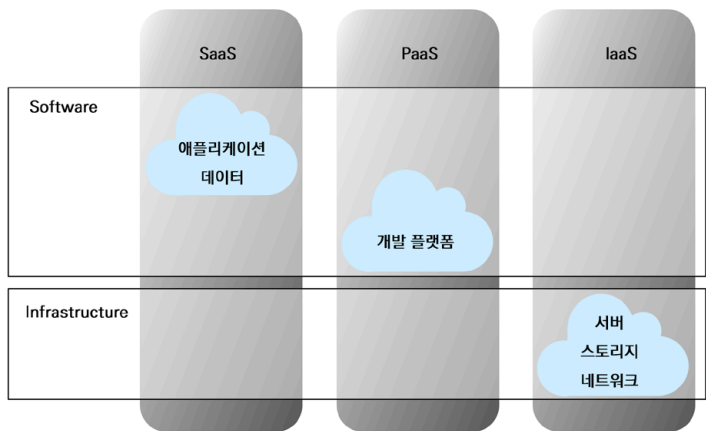


다음 절에서는 클라우드 컴퓨팅의 분류에 따른 개념과 사용되는 자원을 살펴보고, 클라우드를 사용하는 데 필요한 아키텍처 설계와 구축 및 운영 방법을 알아본다.

1.1.1 클라우드 컴퓨팅의 분류에 따른 개념

클라우드 컴퓨팅은 가상화 단계에 따라 크게 SaaS^{Software As A Service}, PaaS^{Platform As A Service}, IaaS^{Infrastructure As A Service}로 구분할 수 있다. 그 밖에도 다양한 유사 개념이 등장하고 있지만, 대표적인 세 가지 개념만 알아도 충분하다.

그림 1-3 SaaS, PaaS, IaaS의 가상화 단계



SaaS

‘On-demand Software’로도 불리며 말 그대로 소프트웨어, 즉 프로그램을 서비스로 제공하는 것이다. 프로그램 및 관련 데이터는 중앙 서버에서 호스팅하고 사용자는 웹 브라우저 등의 클라이언트를 통해 접속하는 형태로, 문서 프로그램이나 DBMS, CAD, CRM, ERP와 같은 다양한 비즈니스 프로그램을 서비스로 제공한다. Microsoft Word, PowerPoint와 같은 프로그램을 특정 컴퓨터에 설치하여 사용하는 것이 아니라 언제 어디서나 사용할 수 있도록 가상화하여 제공하는 것을 말한다. 대표적인 예로는 Google Docs가 있는데, 온라인상에서 개인 또는 그룹 계정으로 문서를 작성하고 공유할 수 있다.

PaaS

SaaS의 개념을 개발 도구의 단계까지 확장한 방식으로, 주로 프로그램 개발을 위해 플랫폼을 구축할 필요가 없도록 모든 개발 도구와 라이브러리를 웹으로 서비스하는 것이다. SaaS는 사용자가 응용 프로그램을 가상화하여 사용하는 것이라면, PaaS는 사용자가 가상화된 개발 도구를 사용하는 것이어서 개발자가 PaaS의 주 사용자가 된다. 프로그램의 제작과 테스트, 적용 및 확장에 필요한 모든 프로그램을 제공하고 환경 구축을 더 쉽고 빠르게 할 수 있다. 예로는 Google App Engine과 Windows Azure가 있다.

IaaS

이 책에서 주로 논의할 IaaS는 CPU, 메모리, 디스크, 네트워크와 같은 IT 인프라를 가상화하여 서비스로 제공하는 방식이다. IaaS를 잘못 이해하면 서버 기반의 호스팅과 동일한 것으로 생각할 수 있다. 그러나 서버 기반의 호스팅은 어떠한 가상화 기술 없이 서비스와 물리적인 컴퓨팅 자원이 1:1로 매핑되는 구조고, IaaS는 공유할 수 있는 유허 컴퓨팅 자원을 다른 가상 서버와 공유하는 구조라는 차이가 있다. 예를 들어, 게임 서비스는 저녁 및 심야 시간대가 Peak Time이라면 기

무어의 법칙(Moore's Law)에 따르면 반도체 메모리 칩의 성능 즉, CPU의 속도나 메모리의 용량은 18개월에서 24개월마다 2배씩 증가하고, 그 가격은 18개월마다 반으로 떨어진다. 기술의 발전에 따라 물리적인 컴퓨팅 자원, 즉 하드웨어는 급격한 발전을 하며 그 가격은 지속해서 하락하고 있다.

01 출처: http://en.wikipedia.org/wiki/Moore's_law



Part 2

비즈니스별 클라우드 아키텍처

Cloud Architecture for Each Business Type

모든 컴퓨팅 서비스를 클라우드에서 운영하는 것이 옳을까? 그렇지 않다. 클라우드는 1장 1절에서 언급한 바와 같이 저비용 고효율에 대한 구성이 필요하거나 유지보수에 대한 인력 자원 투입을 아끼기 위한 서비스에 적합하다. 예를 들어, 대규모 서버를 유연하고 신속하게 구축하는 실험이나 사업이 중요 요소가 아니라면 클라우드 환경으로 전환하는 것은 불필요하다. 또한, 성능이 낮은 개인 PC에서 고성능의 작업을 할 때 버벅이는 정도를 참을 수 있는 경우, 개인 홈페이지를 운영하거나 방문자가 적어 직접 서버를 운용할 수 있는 경우도 마찬가지다.

클라우드는 다음과 같은 비즈니스 모델에 매우 유용하게 활용될 수 있다.

- 1 다양한 컴퓨팅 환경에서 수행해야 하는 서비스
- 2 어떠한 장애나 재난에도 중단되지 않고 운영해야 하는 서비스
- 3 계산이 복잡하거나 데이터 양이 방대한 서비스

- 4 야구 포털 사이트나 동영상 서비스와 같이 특정 기간에 많은 사용자나 트래픽을 지원할 수 있는 인프라가 필요한 서비스
- 5 게임 서비스와 같이 서비스의 흥망에 따라 운영해야 하는 컴퓨팅 자원의 개수가 급격하게 변화하는 서비스
- 6 보안 장비나 특정 전용 장비를 서버에 연동하는 서비스
- 7 콘텐츠 배포 서비스
- 8 수많은 서버 운용과 장애 대응 및 서버 구성에 대해 특화된 인력 지원이 필요한 서비스

이 장에서는 다양한 비즈니스 모델에 따라 어떠한 클라우드 아키텍처가 적합한지에 대해 알아보겠다. 여기서 소개하는 아키텍처는 복합적으로도 구성할 수 있으며 이러한 구성은 클라우드 사업자가 어떠한 인프라와 서비스를 제공하는지에 따라 다르다. 여기에서는 KT ucloud biz를 기준으로 어떠한 구축 방법이 효율적인지 알아보겠다. 또한, 이 내용은 필자가 다년 간 기술 컨설팅과 연구 개발을 하며 사용자의 서비스에 따라 어떠한 구축 방법이 적합한지, 어떠한 구축이 효율적인지에 대해 검증한 구성이다. 물론 서비스에 가장 적합하게 구성하려면 자체 테스트를 거쳐 구축해야 함은 당연한 부분이다. 따라서 각각의 구축 방법에 대해 각종 사례 및 테스트 방법에 대해서도 알아본다.

고성능 연산이나 데이터 저장에 집중된 서비스 구축

Build Data Store or High Performance Oriented Service

이 장에서는 연산^{CPU Oriented}이나 데이터 저장^{I/O Oriented}에 집종화된 서비스를 구축하는 방법을 알아본다. 저장 방법으로는 하나의 ucloud server에 여러 개의 HDD나 SSD의 추가 디스크를 구성하는 방법과 여러 개의 ucloud server가 하나의 NAS로 저장 장치를 구성하는 방법이 있다. 또한, 오브젝트 스토리지 형태의 ucloud storage를 ucloud server와 연동하여 사용할 수도 있다.

4.1 고성능 연산을 위한 가상 서버 구성

DB와 같이 고성능 연산이 필요한 가상 서버의 경우에는 연산 집중화된 구성이 필요하다. DB에 따라 권고하는 기본 가상 서버 스펙이 있기도 하고, 특정 기능을 사용하기 위한 권장 스펙이 있기도 하다. 예를 들어, 리눅스에 구성된 Oracle 7⁰¹의 경우 최소 메모리 사양은 1GB고, MySQL의 DBA를 위한 통합형 비주얼 도구인 Workbench⁰² 기능을 사용하기 위해서는 MySQL 서버 5.1 버전 이상, CPU Single Core 3GHz나 Dual Core 2GHz 이상, 6GB 이상의 메모리가 권장 사

01 오라클 버전별 권장 스펙: <http://www.oracle.com/us/technologies/linux/product/specifications/index.html>

02 MySQL Workbench 기능 권장 스펙: <https://www.mysql.com/support/supportedplatforms/workbench.html>

양이다. 이외에도 고성능 연산을 요구하는 많은 서비스에 대해서는 고성능 가상 서버를 구성하여 사용하면 된다.

다음은 가상 서버의 사이징에 대해 알아두어야 할 부분이다.

KT ucloud server는 32GB 메모리까지 지원하는 표준 형태의 가상 서버와 124GB 메모리까지 지원하는 High Memory 형태의 가상 서버를 제공하는데, 일반적으로 물리 서버와 동일하거나 한 단계 크게 사이징하면 된다. 해당 스펙은 클라우드 콘솔 내, “상품 변경” 버튼을 통해 Scale Up/Down이 가능하므로 너무 정밀하게 사이징하지 않아도 된다. 다만 Scale Up/Down를 변경하려면 해당 서버를 정지한 후 변경하여야 하므로 서비스 순단이 일어나게 된다.

Scale In/Out으로 가상 서버의 개수를 감소/증가시켜 부하 분산할 수 있다. Scale In으로 가상 서버 개수를 줄이면 해당 서버에 수용된 세션이 끊어지게 되므로 이를 위해 세션 정보를 모든 서버가 공유하도록 설정하는 것이 바람직하다. Scale Out으로 가상 서버 개수를 증가시킬 경우 계정별로 수용할 수 있는 세션 수가 정해져 있으므로 계정 개수를 늘리는 것도 고려하여야 한다. 이러한 많은 네트워크 연결을 지원하는 서비스 구축을 위해서는 5장을 참고하면 된다. 많은 네트워크를 연결하는 것이 아니라 고성능 서버를 여러 대 구축하기 위해서는 하나의 계정 내에서 구성할 수 있다.

가상 서버를 Scale In/Out할 경우에는 대상 서버를 스냅샷이나 이미지를 생성하여 새 가상 서버를 생성하면 동일 기능을 하는 가상 서버를 생성할 수 있다. 단, 스냅샷을 생성할 경우에는 가상 서버를 정지하지 않고 바로 생성할 수 있지만, 동작 중인 가상 서버에서 바로 이미지를 생성하려고 할 경우에는 가상 서버를 정지해야 한다. 따라서 스냅샷을 생성하여 해당 스냅샷으로 이미지를 생성하는 것을 권장한다.

영상 트랜스코더^{Transcoder}, 고성능 컴퓨팅^{HPC: High performance computing} 등 물리 서버

환경에서도 CPU 부하율이 매우 높은 서버의 경우에는 물리 서버 대비 2배의 가상 코어수를 선택해야 한다.

가상 서버의 CPU, 메모리 스펙과 네트워크 성능은 정비례하지 않으므로 적은 수의 고사양 가상 서버보다는 많은 수의 저사양 가상 서버로 네트워크 부하를 분산하도록 설계하는 것이 좋다. 물론 가상 서버 내부에서 각자의 서비스에 알맞게 TCP/IP를 튜닝하는 것도 필요하다.

4.2 다양한 종류의 데이터 저장 구성

데이터 저장은 다양한 방법으로 구성할 수 있다. 예를 들어, 각각의 가상 서버에 대용량의 데이터를 저장하거나 여러 서버에서 동일 데이터에 접근해야 할 수도 있고, 사용 빈도에 따라 IOPS와 비용이 다른 저장 방식을 선택할 수도 있다. 각각의 가상 서버에 대용량의 데이터를 저장할 경우에는 HDD나 SDD의 추가 디스크를 여러 개 붙여 사용할 수 있고, 여러 서버에서 동일 데이터를 접근해야 할 경우에는 NAS를 사용할 수도 있다. 또한, 사용 빈도가 낮은 데이터의 경우에는 상대적으로 낮은 IOPS와 저비용의 스토리지를 연동할 수 있다. 물론 각각의 서비스는 저장량과 전송량, IOPS에 따라 각기 다른 과금 체계가 적용되고, 각기 다른 성능을 제공한다.

다음의 [표 4-1]에서는 서비스별 사용 가능 용량과 특이 사항을 정리하였다. 각각의 데이터 저장 서비스에 읽기/쓰기^{Read/Write}할 때 원하는 성능이 나오는지에 따라 서비스를 선택하면 된다. 일반적으로 SSD, Server HDD, NAS, Storage 순으로 고성능을 보이지만 물리 자원을 공유하는 타 사용자와의 자원 경합 및 서비스의 종류, 사용 방식에 따라 다른 성능을 보이게 된다.

표 4-1 서비스별 용량 및 특징 비교

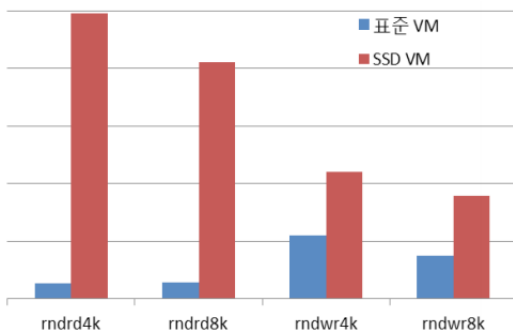
서비스 종류	VM 디스크	NAS	ucloud storage
최대 용량	VM당 12개의 데이터 디스크 (300GB × 12 = 3.6TB)	계정당 10TB 이하(예외를 적용해도 수십 TB 미만)	무제한
성능 수준	일반 : 2,000 IOPS 미만 SSD(천안) : 5,000 IOPS 미만 SSD(서울) : 최대 20,000 IOPS 이웃 효과(Neighbor Effect) 불가피	2,000 IOPS 미만 이웃 효과(Neighbor Effect) 불가피	업로드 300Mbps (×N) 다운로드 500Mbps
파일 시스템	OS 파일 시스템	NFS, CIFS, i-SCSI	Object storage, RESTful API
특징	데이터 디스크가 크면 VM 이 동성 저하	공유형 스토리지	기본 3 copy 저장 모든 파일에 고정 URL 부여 CDN 연계 openStack component AWS S3 compatible
용도	SW 패키지, 자주 변경되는 데이터, DB	여러 서버가 공유하는 파일	배포용 대용량 콘텐츠 아카이빙/백업

4.2.1 데이터 저장을 적절하게 구성하는 방법

다음은 데이터 저장을 적절하게 구성하기 위해 추가 디스크나 NAS를 사용하는 방법이다.

- 일반 가상 서버는 2,000 IOPS 미만의 디스크 I/O 성능을 제공한다.
- 가상 서버에 다수의 디스크를 추가하여 소프트웨어 RAID 구성을 해도 디스크 I/O 성능이 향상되지 않는다.
- SSD 가상 서버는 일반 가상 서버에 비해 높은 디스크 성능을 제공한다. 그렇지만 SSD 특성 상 랜덤 읽기 성능이 탁월한 반면에 쓰기/순차 읽기/순차 쓰기 성능은 일반 가상 서버와 차이가 크지 않다([그림 4-1] 참고).

그림 4-1 ucloud 일반 가상 서버와 SSD 가상 서버의 디스크 I/O 차이



- 가상 서버의 경우에는 이미지를 생성하여 다른 존으로 복사해서 사용할 수 있지만 데이터 디스크의 경우에는 존 간의 복사가 안 된다. 따라서 새로운 디스크 공간을 만들어서 수작업으로 복사해야 한다.
- SSD 가상 서버에 일반 HDD를 연결할 수도 있고, 표준 가상 서버에 SSD를 연결할 수도 있다. High Memory 서버에도 일반 HDD나 SSD를 연결할 수 있으므로 필요에 따라 적합한 저장 장치를 사용하면 된다. SSD로 추가할 경우에는 해당 가상 서버가 4vCore 이상일 경우에만 가능하다.
- 가상 서버의 디스크는 운영체제가 설치되어 있는 디스크(ROOT로 시작되는 이름의 디스크)와 데이터를 저장하기 위한 디스크(DATA로 시작되는 이름의 디스크)가 있다. 데이터 디스크는 하나의 가상 서버에 최대 12개까지 부착할 수 있다. 또한, 데이터 디스크는 가상 서버에 의존적이지 않기 때문에 용도에 따라 다른 가상 서버에 붙일 수도 있다.
- 가상 서버와 추가 디스크, NAS 등 각각의 서비스는 존에 따라 지원하는 내용이 다르므로, 서비스하려는 존을 선택할 때에 원하는 서비스를 사용할 수 있는지도 확인하여 구성하여야 한다. (표 4-2 참고)

표 4-2 존별 사용 가능한 데이터 저장 방식

구분	Central-A 존(천안)	Central-B 존(천안)	Seoul-M 존(서울)	HA존 (김해)	JPN존 (일본)
표준 가상 서버	O	O	O	O	O
High Memory 가상 서버	O	O	O	O	O
SSD 가상 서버	O	O	X	X	X

구분	Central-A 존(천안)	Central-B 존(천안)	Seoul-M 존(서울)	HA존 (김해)	JPN존 (일본)
HDD 추가 디스크	O	O	O	O	O
SSD 추가 디스크	X	X	O	X	X
NAS	O	O	O	X	X
ucloud storage	O	O	X	X	O

(2015년 4월 기준)

4.2.2 추가 디스크 사용 방법 및 유의 사항

윈도우의 경우에는 포털에서 추가 디스크를 생성할 수 있으며 가상 서버에 연결하면 바로 사용할 수 있다. 리눅스의 경우에는 다음과 같이 추가 디스크 생성 및 연결을 한다.

1. 포털에서 추가 디스크 생성 및 가상 서버에 연결
2. 가상 서버에 접속하여 `fdisk -l`로 확인(연결 완료 시 `/dev/svdb`와 같이 표시됨)
3. `fdisk /dev/xvdb → 파티션 추가(n) → primary 파티션(p) 명령으로 추가`
4. `mkfs.ext3 /dev/xvdb1` 명령으로 원하는 형태의 파일 시스템 생성
5. `mount /dev/xvdb1 /data`와 같이 해당 디스크를 마운트(재부팅 시 자동으로 마운트하려면 `/etc/fstab`에 마운트 내용을 추가)해야 한다

이렇게 생성 및 추가된 디스크를 삭제할 때에는 반드시 가상 서버 내에서 해당 디스크의 `umount`를 수행하고 `/etc/fstab` 내역을 삭제한 후 진행해야 한다. 추가 디스크를 사용 중인 서버의 ROOT 디스크를 스냅샷이나 이미지로 만들 경우에는 반드시 `/etc/fstab`의 추가 디스크 내역을 주석 처리한 후 이미지를 생성해야 추후 생성된 가상 서버에서 정상 부팅이 가능하다.

한편, 여러 디스크를 묶어서 사용할 때에는 특정 디스크에 문제가 발생하여 RAID 구성이 깨지는 문제가 발생할 수 있고, 심각한 경우 전체 데이터가 손실되는 큰 문

제로 이어질 수 있다. 이를 방지하려면 리눅스 운영체제에서는 여러 디스크를 묶어서 사용하는데 그 방법은 다음과 같다.

LVM 방식

논리적 볼륨 매니저 방식 LVM, Logical Volume Manager 은 기본적으로 각 디스크를 물리적 볼륨 PV, Physical Volume 으로 만들고 여러 PV를 묶어서 볼륨 그룹 VG, Volume Group 으로 만든 뒤 최종적으로 논리적 볼륨 LV, Logical Volume 으로 묶어 사용하는 방법이다. LV를 구성할 때 단순 묶음 또는 Striping 구성 등을 할 수 있지만, LV의 메타 데이터가 깨질 경우 전체 볼륨을 사용할 수 없게 되는 문제가 있기 때문에 별도의 백업이 필요하다. 이를 위해 볼륨 그룹 디스크립터 영역 VGDA, Volume Group Descriptor Area 백업을 사용한다. 이는 LVM을 제어하는 메타 데이터에 대한 백업으로, 해당 명령어는 실제 데이터에 대한 백업이 아닌 LVM 구성에 필요한 메타 데이터에 대한 백업 기능이다. 데이터의 변동이 많은 /var과 같은 디렉터리는 백업 도중에 데이터가 영킬 수 있기에 이러한 상황을 방지하기 위해서 스냅샷 LV를 만들어 관리하면 좋고, 스냅샷 LV의 경우 원본 LV와 같은 크기로 하는 것이 좋다. 스냅샷 LV가 꼭 차면 자동으로 스냅샷을 drop시켜 사용할 수 없게 되며 원본 크기보다 같거나 커야 오버플로우가 발생하지 않는다. 또한, vgcfgrestore 명령어로 백업한 내용을 복구할 수 있고, lvreduce 명령어로 LV의 용량을 줄일 수 있다.

mdadm 방식

리눅스에서 소프트웨어 RAID 구성을 지원하는 프로그램으로, RAID-0, 0+1, 1, 4, 5, 6, 10 등 다양한 RAID 구성을 지원한다. 운영체제에서 별도로 mdadm 구성을 저장하지 않기 때문에 구성 정보에 대한 설정 파일(/etc/mdadm/mdadm.conf)을 백업해 두는 것이 좋다. mdadm -- detail --scan 명령으로 현재 구성 정보를 확인할 수 있다. --grow 옵션으로 디스크를 추가·변경할 때에는 백업 파일의 위치를 지정해야 한다. 이 경우 별도의 디바이스에 백업 파일을 저장하는

것이 좋으며, 백업된 파일은 --assemble 명령어로 복구할 수 있다. 모니터 모드에서 장치 변경, 실패 등에 대한 모니터링을 수행할 수 있으며, 멀티패스 기능을 사용할 때 mdmpd 유틸을 사용하여 여러 경로의 디바이스를 모니터링할 수도 있다.

다음은 ucloud biz 환경에서 여러 디스크 구성 방식에 대한 성능 벤치마크를 진행한 것이다. 결과적으로 mdadm으로 S/W RAID 구성을 하는 방법이 LVM Striping 모드보다 다소 나은 성능을 보이는 것을 확인하였다. [그림 4-2]는 EXT3와 EXT4 파일 시스템이 단일 디스크로 구성된 것과 LVM Striping 방식과 mdadm RAID-0(Striping) 방식으로 구성된 볼륨에 대한 vdbench 벤치마크 테스트 결과다. 연속/랜덤/혼합sequential/random/mixed 방식의 읽기/쓰기 속도 측정 결과이며, 단일 EXT3 디스크 기준 상대 수치로 보면 된다.

- 테스트 환경: CentOS 5.8 64bit (2vCore, 2GB 사양)

- 디스크 구성

EXT3 파일 시스템의 단일 디스크: 50GB 디스크

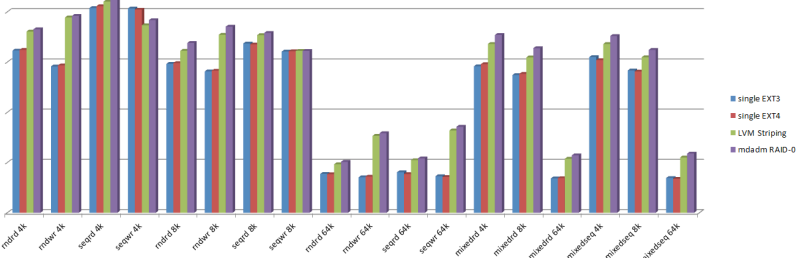
EXT4 파일 시스템의 단일 디스크: 50GB 디스크

LVM: 50GB Disk(EXT3)×6, Striped 구성

mdadm: 50GB Disk(EXT3)×6, RAID-0 구성

- 테스트 결과

그림 4-2 디스크 구성 방식에 따른 상대 성능 평가



(rnd = 랜덤random, seq = 연속sequential, mixed = 혼합rnd+seq, rd = 읽기read, wr = 쓰기write)

마지막으로, 디스크를 사용할 때 많은 파일을 사용하고 있는 가상 서버의 경우 유의해야 한다. 디스크 용량이 충분한데도 inode라는 리눅스 운영체제에서 파일 시스템 오브젝트를 대표하는 자료구조인 인덱스 노드의 개수 부족 현상이 일어나는 경우가 있다. 이러한 현상은 파일 사이즈가 작은 파일을 많이 저장하거나 디렉터리나 심볼릭 링크를 많이 생성하였을 경우, 컨테이너화 서비스(Docker 등)를 사용할 경우에 주로 발생하는 현상이다. `df -h` 명령으로 가용 용량을 확인했을 때 디스크에 용량이 충분한데도 'No space left on device' (디바이스에 가용 공간이 없다)라는 메시지와 함께 파일이 저장되지 않는다. 이 경우 `df -I` 명령으로 확인하면 inode가 꽉 찬 경우인 것을 알 수 있다. 일반적으로 ucloud biz에서 가상 서버의 디스크 사용을 위해서는 다음의 과정을 거치는데, 포맷할 때 inode 관련 설정값을 바꿀 수 있다.

- 디스크 파티셔닝: `fdisk /dev/xvdb`
- 해당 디스크 포맷: `mkfs.ext3 /dev/xvdb1`

기본 설정값은 다음 명령으로 확인할 수 있다.

```
# cat /etc/mke2fs.conf | head -5
[defaults]
base_features = sparse_super,filetype,resize_inode,dir_index,ext_attr
blocksize = 4096
inode_size = 256
inode_ratio = 16384
```

inode_ratio 값이 기본으로 16384로 설정되어 있는데, mkfs를 수행할 때 `-i` 옵션으로 해당 값을 변경할 수 있다. 여기서 inode_ratio란 inode당 바이트 수로, 1024 이상의 값이면서 블록 크기 이상의 값이어야 한다. 즉, inode_ratio를 낮춰서 inode 수를 늘릴 수 있다. 다음은 inode_ratio를 4096으로 설정하여 포맷한 경우다.

```
# mkfs.ext3 /dev/xvdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
5242880 inodes, 20970841 blocks
1048542 blocks (5.00%) reserved for the super user
```

```
# mkfs.ext3 -i 4096 /dev/xvdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
20971520 inodes, 20970841 blocks
1048542 blocks (5.00%) reserved for the super user
```

이와 같이 inode_ratio를 16384에서 4096으로 ¼배 조정하니, inode의 수는 4배로 늘어난 것을 확인할 수 있다. 그런데 기존에 사용 중인 디스크의 경우 다이나믹하게 inode를 증가시키는 것은 일반적으로 불가능하다. 이와 같은 경우에는 새로 포맷해야 하므로 다음의 과정으로 진행한다.

1. 대상 디스크 파일 시스템 별도 백업
2. 디스크 언마운트
3. 앞의 포맷 과정 (inode_ratio 적용) 재수행
4. 파일 원복

앞에서 살펴본 LVM의 경우에는 LV 용량을 늘리는 것으로 inode를 증가하는 것이 가능하고, 증가하는 용량에 비례하여 inode가 증가한다. 이는 다음과 같이

lvextend [옵션] [LV명] 명령으로 수행할 수 있다.

```
# lvextend -L+1G /dev/vg/var
umount /var
resize2fs /dev/vg/var
mount /var
```

NOTE 디스크 성능 측정 방법

A. 리눅스(FIO 사용)

FIO^{Flexible I/O Tester}로 ucloud server에 부착된 일반 디스크와 SSD의 성능을 측정할 수 있다. 이와 같은 성능 측정은 동일한 물리 서버에 타 사용자가 어떻게 사용하고 있는지에 따라 다른 결과값이 나올 수 있다.

1. 설치법

```
yum install libaio* gcc wget make
wget http://brick.kernel.dk/snaps/fio-2.2.6.tar.gz
gunzip fio-2.2.6.tar.gz
tar -xf fio-2.2.6.tar
cd fio-2.2.6
./configure
make
make install
```

2. 사용 방법

[랜덤 읽기(Random Read) 테스트]

```
fio --directory=[마운트한 폴더명] --name [파일 이름] --direct=1 --rw=randread
--bs=4k --size=1G -- numjobs=16 --time_based --runtime=180 --group_reporting
-norandommap
```

[랜덤 쓰기(Random Write) 테스트]

```
fio --directory=[마운트한 폴더명] --name [파일 이름] --direct=1 --rw=randwrite
--bs=4k --size=1G -- numjobs=16 --time_based --runtime=180 --group_reporting
-norandommap
```

3. 실행 화면

```
[root@testssd ssd]# fio --directory=/ssd/ --name fio_test_file --direct=1 --rw=randread --bs=4K --size=1G --numjobs=16 --time_based --runtime=180 --group_reporting --norandommap
fio_test_file: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=sync, iodepth=1
...
fio_test_file: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=sync, iodepth=1
fio-2.0.14
Starting 16 processes
Jobs: 16 (f=16): [rrrrrrrrrrrrrrrr] [17.7% done] [80664K/0K/0K /s] [20.2K / 0 iops] [eta 02m:29s]
```

B. 윈도우(SQLIO 사용)

윈도우의 경우 마이크로소프트에서 제공하는 I/O 측정 툴로 테스트할 수 있다.

1. 설치법

다음 링크에서 설치 파일을 다운로드하고 SQLIO.msi 파일을 실행하여 설치한다.

<http://www.microsoft.com/en-us/download/details.aspx?id=20163>

2. 사용 방법

설치 폴더에서 param.txt 파일을 열어 파일 경로를 설정하고 크기를 MB단위로 수정한다. 커맨드 창을 열어 다음과 같이 테스트를 수행하면 된다.

[랜덤 읽기(Random Read) 테스트]

```
sqlio.exe -kR -t8 -o8 -s120 -frandom -b4 -LS -Fparam.txt
```

[랜덤 쓰기(Random Write) 테스트]

```
sqlio.exe -kW -t8 -o8 -s120 -frandom -b4 -LS -Fparam.txt
```

3. 실행 화면

```
C:\Program Files <x86>\SQLIO>sqlio.exe -kR -t8 -o8 -s120 -frandom -b4 -LS -Fparam.txt
sqlio v1.5.SG
using system counter for latency timings, 6250000 counts per second
parameter file used: param.txt
file e:\testfile.dat with 2 threads (0-1) using mask 0x0 (0)
2 threads reading for 120 secs from file e:\testfile.dat
using 4KB random I/Os
enabling multiple I/Os per thread with 8 outstanding
size of file e:\testfile.dat needs to be: 2147483648 bytes
current file size: 1073741824 bytes
need to expand by: 1073741824 bytes
expanding e:\testfile.dat ... done.
using specified size: 2048 MB for file: e:\testfile.dat
initialization done
CUMULATIVE DATA:
throughput metrics:
I/Os/sec: 20088.64
MBs/sec: 78.47
latency metrics:
Min_Latency(ms): 0
Avg_Latency(ms): 0
Max_Latency(ms): 24
histogram:
ms: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24+
%: 91 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

4.2.3 NAS 사용 방법 및 유의 사항

NAS 서비스는 윈도우의 경우에는 서버 내에서 '디스크 관리'로 부착할 수 있지만 리눅스의 경우에는 다음과 같은 방법으로 각 서버에서 설정해야 연결할 수 있다.

1. 마운트할 수 있는 볼륨이 있는지 확인(showmount -e [대상주소])

```
[root@test ~]# showmount -e 10.17.238.200
Export list for 10.17.238.200:
/root (everyone)
```

2. 마운트 디렉터리 생성(mkdir [디렉터리명])

```
[root@test ~]# mkdir nfs_mnt
```

3. 마운트(mount -t nfs [대상주소:/경로] [mount할 디렉터리])

```
[root@test ~]# mount -t nfs 10.17.238.200:/root /root/nfs_mnt/
```

4. 마운트 상태 확인

```
[root@test ~]# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
18G 1.2G 16G 7% /
tmpfs 497M 0 497M 0% /dev/shm /dev/xvda1 485M 73M 387M 16% /boot
10.17.238.200:/root 1000G 256K 1000G 1% /root/nfs_mnt
```

5. 볼륨 공유 상태 확인(첫 번째 가상 서버에서 공유할 파일 생성)

```
[root@test ~]# cd /root/nfs_mnt
[root@test nfs_mnt]# touch test.txt
[root@test nfs_mnt]# ll
total 0
-rw-r--r-- 1 root root 0 Nov 3 11:07 test.txt
```

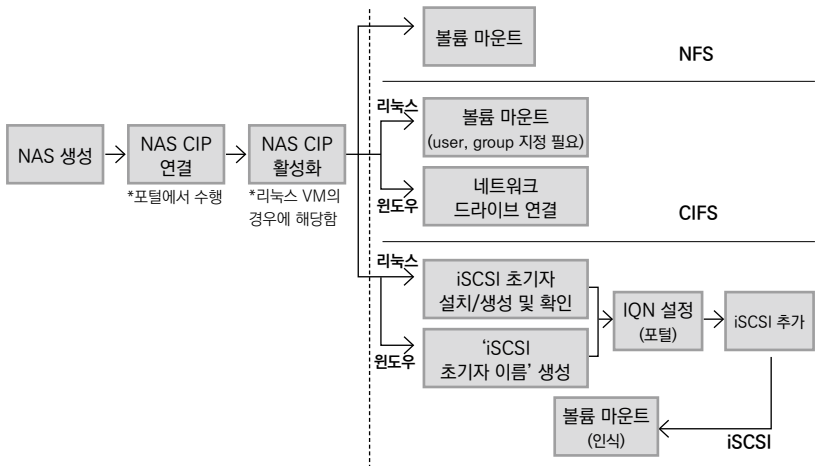
6. 볼륨 공유 상태 확인(두 번째 가상 서버에서 공유된 파일 확인)

```
[root@test ~]# cd /root/nfs_mnt
[root@test nfs_mnt]# ll
total 0
-rw-r--r-- 1 root root 0 Nov 3 11:07 test.txt
```

NAS는 ACL(접근제어) 기능을 사용하여 볼륨별로 접근할 수 있는 IP 대역을 지정하여 가상 서버를 설정할 수 있다. 또한, 리눅스 운영체제를 위한 NFS와 윈도우 운영체제를 위한 CIFS와 iSCSI 프로토콜을 제공한다. 하나의 NAS 볼륨으로 리눅스와 윈도우 양쪽 운영체제에서 모두 사용하려면 CIFS와 iSCSI 방식으로 구성할 수 있다. 리눅스 운영체제에서 CIFS 볼륨 마운트 시 다음과 같이 해당 볼륨을 사용할 사용자와 그룹을 지정해야 한다(그림 4-3 참고).

```
#mount -t cifs //192.168.0.13/linux /home/Myhome/Files -o uid=name, gid=users, username=admin
```

그림 4-3 NAS 생성 절차 및 연결 방식



이외에도 NAS를 윈도우 IIS에서 매핑⁰³하거나 윈도우 미디어 서비스로 구성⁰⁴ 및 Filezilla FTP 서버로도 구성⁰⁵할 수 있다.

4.3 구성 사례: 사용량이 적은 대용량 영상 파일을 별도로 관리하는 ucloud server와 storage의 복합 구성

L그룹사는 스마트 TV에서 사용하는 대용량의 영상 파일을 ucloud server의 디스크와 ucloud storage의 복합 구성으로 관리하였다. 해당 서비스에서 각각의 파일에 접근하는 속도를 측정한 뒤 자체 서비스의 알고리즘에 따라 덜 사용하는 파일은 스토리지에 적재하고, 최근에 사용하는 파일이나 자주 사용하는 파일은 디스크에 적재하는 구조다. 이와 같은 구성은 스토리지에 저장된 데이터가 CDN과 같이 배포될 소지가 없고, 서버에서 관리만 적절하게 되면 고효율을 누릴 수 있는 구조이므로 적절한 속도와 비용 절감을 동시에 얻을 수 있다.

K그룹사의 경우 CCTV 서비스를 ucloud server와 ucloud storage에 적재하였다. CCTV 영상을 FTP로 업로드 및 로드밸런싱하여 ucloud storage에 저장하고, 사용자는 별도의 웹 서버로 접근하여 저장된 영상 파일을 스트리밍받는 구조다. CCTV는 영상을 저장할 때 채널당 트래픽 수준이 높으므로(256kbps ~ 1,024kbps 수준으로 CCTV 기종별로 다양) CCTV와 연동하는 영상 저장 서버를 여러 계정에 분산하는 구조로 구성하였다. 이와 같은 계정 분산 구조는 4장을 참고하면 된다.

[그림 4-4]를 보면 서비스별 서버는 ucloud server에 구성하고, 실제 영상 데이터는 ucloud storage에 저장하는 구조다. 영상 저장 서버는 CCTV로부터 수신한 동영상 파일을 가공하여 ucloud storage에 저장한다. 이때 Swift 툴을 사용

03 NAS를 IIS에서 매핑하는 방법: <http://cafe.naver.com/ucloudbiz/90>

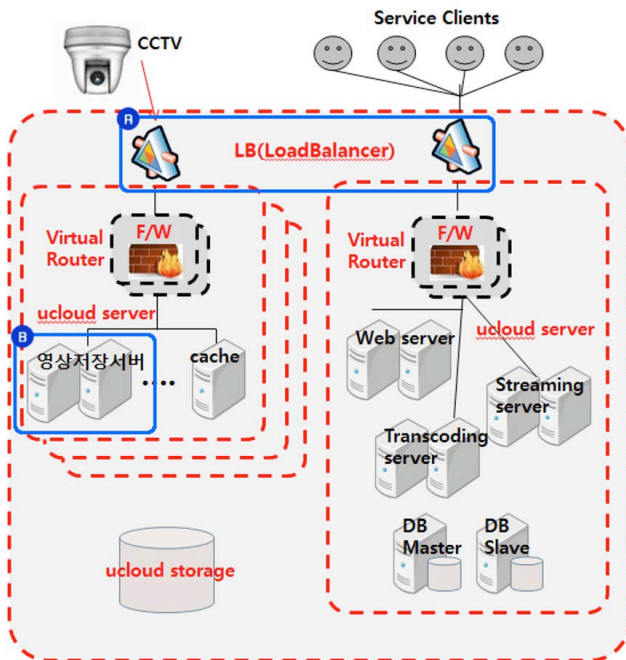
04 NAS를 이용한 윈도우 미디어 서비스 구성: <http://cafe.naver.com/ucloudbiz/77>

05 NAS를 이용한 Filezilla FTP 서버 구성: <http://cafe.naver.com/ucloudbiz/93>

하여 파일을 chunking하면 업로드 속도를 높일 수 있다. 또한, 영상 저장 서버가 파일을 저장하기 전에 서버에서 파일을 이중화 관리하기 위해 메모리 기반의 캐시 서버를 운영하였고, 사용자가 별도의 웹 서버로 접근하여 저장된 영상 파일을 스트리밍받을 수 있게 설계하였다.

해당 사례에서 사용한 서버의 사양은 CCTV 저장을 위해 VR당 영상 저장 서버(4vCore/4GB) 7대, 캐시 서버(4vCore/8GB) 1대, DB 서버(Master/Slave, 8vCore/16GB) 2대, 웹 서버(4vCore/8GB) 1대, 스트리밍과 트랜스코딩^{Transcoding} 서버를 각 1대씩(8vCore/8GB)으로 구성하였다. 여기에 사용한 동시 접속 세션은 VR당(계정 당) CCTV 1,000개 세션을 접속하는 구조이며, 3개의 계정으로 총 2,700개의 CCTV를 수용하였다.

그림 4-4 K그룹사의 CCTV 영상을 ucloud storage에 구성한 구조



NOTE

사용한 로드밸런서는 HTTP가 아닌 FTP에 대해 부하 분산하며 영상 저장 서버를 지정하는 역할만 하기 때문에 대용량의 로드밸런서가 필요한 것은 아니다. FTP Passive Port를 지정할 때에는 잘 알려진 포트^{well-known port}가 아닌 10,000번 이상의 포트를 사용하되, ISP 사업자의 보안 정책(차단하는 포트들)을 사전에 확인할 필요가 있다. FTP 이외에 VR의 포트 포워딩에도 22, 80과 같이 잘 알려진 포트를 공공 포트^{public port}로 사용하는 것은 바람직하지 않지만, 사설 포트^{private port}인 ucloud 내부적으로만 사용되는 포트에는 잘 알려진 포트를 사용해도 무방하다.