3

Cloud Computing을 통한 Bioinformatics 산업의 한계 극복

삼성SDS ▌김성권 · 박용민

1. 서 론

Bioinformatics는 IT 자원을 이용해 DNA, RNA, 단백질 등 분자생물학과 관련된 정보를 수집하고 분석하는 기술이다. 이들 데이터는 그 양이 매우 방대하여 저장하고 분석하기 위한 하드웨어 자원 소요가 크다. 또한 데이터가 연구 기관마다 각각 다른 형식으로 저장되어 있고, 데이터를 분석하는 알고리즘도 기관마다 상이하여 이들을 얼마나 효율적으로 표준화하여 통합하는지가 기술개발의 중요한 목표가 된다. 하지만 지금까지의 Bioinformatics 시스템은 이러한 사항을 반영하지 못하여 시장을 크게 키우지 못하였고, 고객사(제약회사, 바이오 기업)의 in-house 부서로 통합되는 경우가 빈번하였다.

이러한 문제를 해결하기 위해 2009년을 기점으로 일부 Bioinformatics 업체는 Cloud Computing 기반의 시스템을 파일럿 형태로 출시하고 있다. 2009년 3월에는 DNA시퀀서 제조업체인 Applied Biosystems의 software development community 멤버인 GeoSpiza가 그림 1과 같이 자사의 웹 기반 유전자 분석 시스템을 Amazon 웹 서비스의 컴퓨팅 자원을 바탕으로 서비스한다고 밝혔다[5].

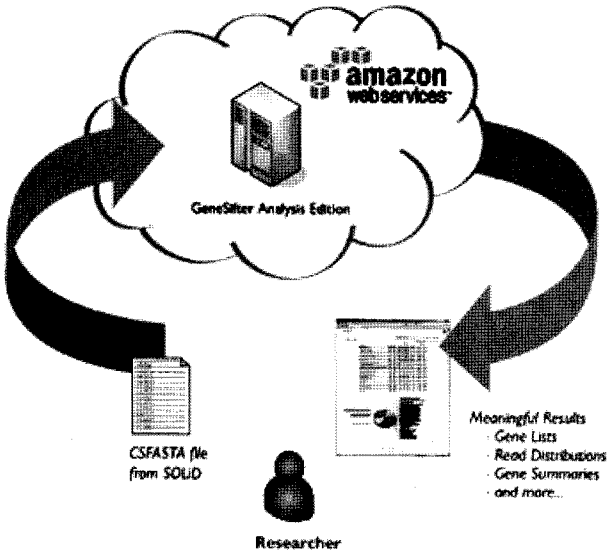


그림 1 GeoSpiza사의 Cloud Computing 기반 유전자 분석 시스템

GeoApiza 사와 같이 Bioinformatics 기업은 Cloud Computing을 도입함으로서, 표준 API를 통해 다양하고 방대한 데이터와 알고리즘을 효율적으로 통합하고 분산 컴퓨팅과 가상화를 통한 하드웨어 운용 비용 절감을 시도하고 있다.

이에 본 고에서는 우선 Bioinformatics 시스템의 기능을 살펴보고, 이를 통해 통합되지 않은 데이터와 알고리즘, 하드웨어 자원의 비용 증가라는 Bioinformatics 시스템의 두 가지 한계의 원인을 추적한다.

그리고 Cloud Computing 이 이러한 한계를 극복하기 위한 기술요소를 갖춘 최적의 방안이 될 수 있는 이유를 설명하고자 한다.

2. Bioinformatics에 적용되는 IT 기술 요소

Bioinformatics란 생물자원 정보를 IT기술을 통해 분석하여 신약개발과 예방의학 등에 활용하기 위한 의미 있는 정보를 추출하는 학문이다.

생물자원의 정보는 DNA, RNA, 단백질, 신진 대사 등과 관련된 정보인데 이 중 1인의 DNA 염기 서열만 해도 약 30억 개의 문자 쌍으로 이루어진 대용량의 데이터이다.

이러한 대용량 데이터를 효율적으로 관리하고 다양한 알고리즘의 적용을 통한 분석을 위해 IT 기술이 필요하다.

Bioinformatics에서 IT의 역할을 명확히 하기 위해 Bio에 해당하는, 즉 분자 생물학의 프로세스를 간단하게 요약하면 그림 2와 같다.

아래 그림과 같이 분자 생물학이 다루는 DATA는 크게 DNA, RNA, 단백질과 관련된 데이터이다. 또한 순서대로 연관관계를 가지고 있는데, 이를 자세히 살펴보면 DNA는 A,T,G,C 의 문자로 이루어지는데, 여기서 유전되는 서열만 분리한 후 T를 G로 바꾸어 RNA

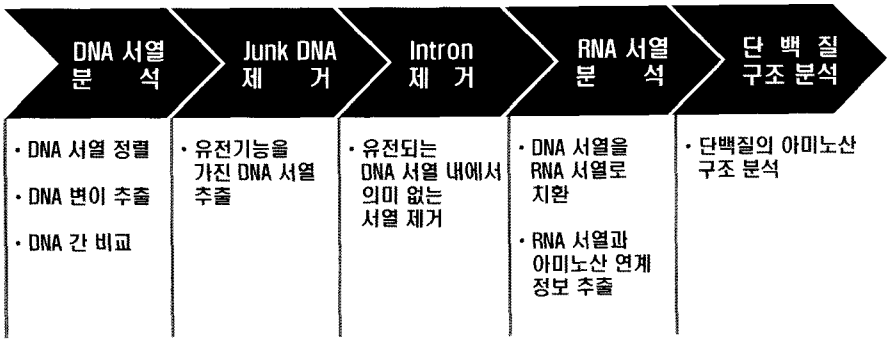


그림 2 분자 생물학의 프로세스

서열을 생성하며, 다시 RNA 서열을 3개의 문자 단위로 해석하여 아미노산과 연계시킨다. 아미노산의 집합은 단백질을 구성하게 된다.

이러한 과정들에 IT가 필요한 이유는 DNA, RNA, 단백질이 알파벳 문자로서 표현된다는 점이다. 따라서, 문자간의 조합과 비교, 검색을 통해 그림 1과 같은 분자 생물학의 프로세스를 IT 기술을 통해 수행하고 지원할 수 있다.

Bioinformatics에 활용되는 소프트웨어는 표 1과 같이 Search Tools, Visualization Tools, Data Mining and Clustering로 구분되며 프로세스 전반에 걸쳐 Data Management Tools이 사용된다. Data의 형태는 DNA, RNA, 단백질에 따라 표 2 형태를 가진다.

살펴본 바와 같이 IT 기술은 다양한 종류의 데이터의 스키마를 정의하고 보관하며 각 데이터 간의 관계를 규명할 수 있는 분석 알고리즘을 제공한다. 또

표 1 Bioinformatics 관련 소프트웨어 역할

구분	소프트웨어 역할
Search	· DNA, RNA염기서열 검색 · 단백질의 아미노산 서열 검색
Visualization	· 단백질 3차원 구조 표현 · 도표 등을 통한DNA/RNA의 표현
Data Mining	· 단백질간의 아미노산 서열 정보 비교 · 단백질의 아미노산 서열로부터 3D 구조 예측 · 통계적으로 유사한 아미노산 서열을 가지는 단백질 추출 · 유사한 단백질간의 분류 · DNA간, RNA간 염기서열간 비교 · Junk DNA 및 Intron을 제외한 유전자 추출 · DNA/RNA 서열 조합 및 Mapping · DNA/RNA와 단백질간의 관계 추적 · 특정 질환군 환자의DNA/RNA 정보와 질병정보간의 관계 추적
Data Management	· 대용량 데이터에 대한 분류 · 데이터 Life cycle 정책 수립 및 적용 · 데이터 접근 권한 제어 · 데이터 표준Interface 정의

표 2 Bioinformatics 관련 데이터 Type

구분	데이터 형태
DNA	· 정렬되지 않은 DNA 서열 데이터 · 표준 DNA 서열 데이터 · 정렬이 완료된 DNA 서열 데이터 · DNA 변이정보 데이터 · 공통서열(유전자발현이 높은 서열정보) 데이터 · Junk DNA 가 제거된 DNA 서열 데이터
RNA	· mRNA 서열 데이터 · mRNA 서열과 DNA 서열 관계 데이터 · Intron 을 제거하고 Exon 만 남은 mRNA 서열 데이터 · Open Reading Frams 데이터
단백질	· 아미노산 서열 데이터 · mRNA서열과 아미노산의 관계 데이터 · 아미노산과 단백질관계 데이터 · 단백질과 신진대사 관계 데이터 · DNA 서열과 mRNA 서열 관계 데이터 · DNA 서열과 단백질의 관계 데이터

한 신약 개발기간이 평균 5~8년이 소모되므로 해당 기간 동안의 데이터 백업 및 복구를 위한 기능도 제공하며, 분석하는 데이터의 양이 방대하여(1인의 DNA 서열 데이터: 최소 3Gbyte) 이를 장기간 동안 저비용으로 관리할 수 있는 기능도 구현한다.

프로그램 개발 언어의 경우 과거에는 Perl을 이용한 경우가 다수였으나 특정 분야에 종속된 소프트웨어만 제작만 가능하여 SW의 활용성에 많은 문제가 있었으며 분석된 데이터를 GUI 기반으로 변환하기가 어려웠다. 이를 해결하기 위해 최근엔 JAVA 기반으로 개발이 이루어지고 있으며 관련된 API도 배포되고 있다.

3. Bioinformatics 산업의 한계

2007년 전 세계 바이오 기업의 매출은 약 85조원인데 이 중 Bioinformatics 기업의 매출은 2% 정도(약 2조원)이다. Bioinformatics가 바이오 기업에서 수행하는 사업에 있어 기반이 되는 분야임에도 불구하고 매출 비중이 낮은 것은, 제약업체나 바이오 기업이 in-

house에서 Bioinformatics 기술을 사용한다는 사실에 기인한다. 그 이유는 Bioinformatics에서 생성하는 데이터와 알고리즘이 업체의 시스템마다 상이하여 통합된 지식으로 승화되지 못하고 있고, 시장규모의 확장을 위해 필요한 소프트웨어 개발 비용과 하드웨어의 비용을 감당할 수 없었기 때문이다.

아래의 가상 시나리오를 통해 현재 Bioinformatics 산업의 한계를 인지할 수 있다.

'제약회사에 근무하는 A연구원은 블록버스터 의약품의 특허만료로 인해 회사의 수익급감이 예상됨에 따라, DNA와 RNA 그리고 단백질 등을 이용한 바이오 신약 개발에 투입되었다. 경쟁사 역시 동일한 움직임을 보이고 있으므로 단시간에 신약물질 후보 군을 찾아야 한다. 신약개발을 위해 A연구원은 암을 유발하는 단백질 후보 군을 찾고 이 단백질과 연계된 RNA와 DNA의 서열과 구조를 밝혀 암 유발 단백질을 선정한 후 바이오 신약을 개발해야 한다. 하지만 각 단계마다 활용하는 소프트웨어가 제각각 이라 사용하기가 불편하였고, 임상검증과 단백질, DNA, RNA의 구조와 역할 등에 해석에 대한 데이터 수가 부족하여 통계적으로 적합하지 않아 보였다. 이를 개선하고자 바이오인포메틱스 SW사에 연락하였으나 다른 기관의 데이터를 가져와서 마이그레이션 하기 위해선 많은 시간이 소요되고 소프트웨어기능 역시 변경이 되어야 하므로 단시간 내에 개선이 불가능하다고 한다. A연구원은 이러한 사실을 경영층에 보고하였고, 경영층은 유용성이 커 보이지 않는 소프트웨어를 유지할 필요가 없다고 판단하여 바이오인포메틱스 SW사와 계약을 해지하고 회사 내에 자체적으로 개발팀을 만들어 오픈 소스를 활용하여 회사에 적합한 소프트웨어를 만들기로 하였다.'

3.1 통합되지 않은 데이터와 알고리즘

2003년 인간의 DNA에 대한 염기서열이 완성된 후 정부, 연구기관, 바이오 기업, 제약 업체 등에서 유전자 분석을 통한 다양한 연구를 진행하고 있다.

각 기관들은 자체 내에 DNA, RNA, 단백질 분석 등과 관련된 DataBase를 구축하였으며, 현재 공개된 것만 약 500개에 달하며 분석알고리즘에 관해서도 수많은 논문이 발표되고 있다. 정확한 유전자 분석을 위해서는 이들 DataBase를 연결하여 이질적인 데이터의 통합을 통해 지식으로서 승화시키고 다양한 알고리즘을 적용할 수 있는 소프트웨어 유연성이 확보되어야 한다.

이러한 사실은 제조업이나 소비재 산업 등 여타의 산업에 적용되는 IT 시스템이 Scale-up , 즉 특정기업 내부에서 Customizing 되는 것을 강조하는 반면, 제약산업에서는 산업 내 기업과 연구기관, 정부 등 산업 내 전체 구성원들간의 Scale-out, 다시 말해 확장성이 더 가치 있음을 의미한다.

하지만 기존의 Bioinformatics 시스템들은 다양한 기관들간의 Scale-out은 물론 시스템 내부에서도 Scale-out을 지원하는 방식으로 설계되지 않았다.

이러한 문제점을 자세히 살펴보면 아래와 같다.

1) In-house와 Internet에 있는 DataBase 간의 연결이 어려움

Internet 상에 공개된 유전자 관련 DataBase는 데이터 연동을 위한 다양한 방법을 제공하고 있으나. DataBase 마다 연결방식이 상이하여 통합하기가 쉽지 않다.

2) 다양한 알고리즘에 대한 적용을 용이하게 하는 소프트웨어의 유연성 미흡

Bioinformatics의 데이터는 각 연구기관과 업체에 따라 상이할 수 있다. 그 이유는 각 데이터를 생성하기 위한 소프트웨어의 알고리즘이 상이하기 때문이다. 즉, DNA의 서열을 정렬하기 위한 통계적인 알고리즘이 틀리며, mRNA의 서열을 정리하는 방법 또한 다양할 수 있다. 하지만 이러한 다양한 알고리즘의 적용을 용이하게 하는 Interface가 제공되는 시스템이 많지 않다.

3) Standard protocol의 부재

다양한 데이터와 알고리즘을 효율적으로 교환하기 위해서는 표준화된 프로토콜이 필요하지만 국제적으로 지정된 표준이 없다.

4) 통합되지 않은 용어

각 기관마다 DNA, RNA, 단백질의 기능에 대한 다양한 의견을 DataBase에 저장하고 있다. 하지만 이러한 의견들의 용어가 각 기관마다 상이하여 시스템이 동일한 데이터로 인식하지 않는다. 현재 미국, 일본, 유럽의 공공 DataBase간에는 용어에 대한 정의를 해놓은 상태이다.

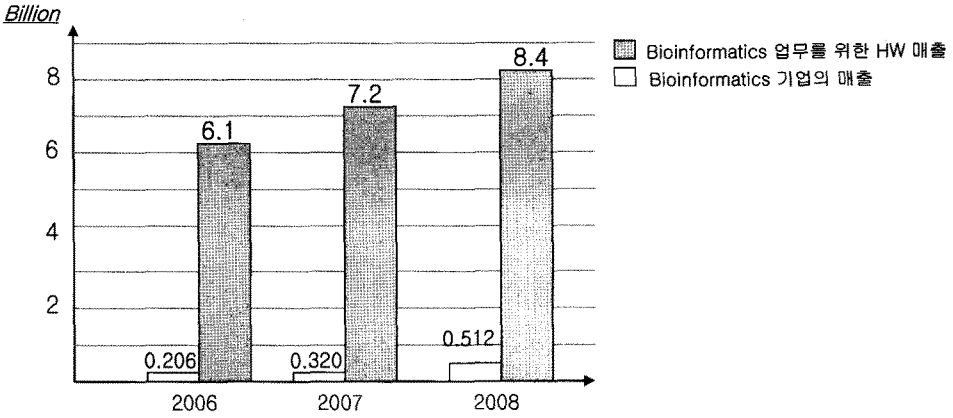
5) 지속적이지 않은 데이터 Update

새로운 이론과 데이터에 대한 갱신, 그리고 새로운 Public DB에 대한 링크정보 갱신이 지속적으로 이루어지지 않고 있다.

3.2 하드웨어 자원 비용의 증가

DNA 서열 분석 장비의 기술과 가격이 발전함에 따라 DNA의 전체 서열에 대한 데이터가 증가하고 있으

표 3 Bioinformatics 기업의 매출과 사업을 위해 소요되는 하드웨어 매출[1]



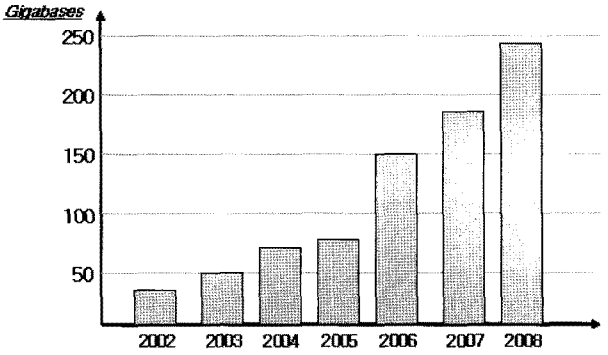


그림 3 EMBL(European Molecular Laboratory)의 데이터 증가추이

며, 더불어 RNA와 단백질에 대한 데이터도 증가하고 있다. 따라서 이를 보관하기 위한 Storage 수요도 급격히 발생하고 있다.

그 예로서, EMBL(European Molecular Laboratory)에서 구축하여 운영중인 DNA,RNA,단백질에 관련된 데이터 베이스는 그림 3에서와 같이 2002년 이후 매년 크게 증가하는 추세이다[1].

하지만 현재의 Bioinformatics 기업은 이러한 수요를 충족시키지 못하고 있다.

데이터의 증가에 따른 Storage의 증가는 CPU 등의 컴퓨팅파워를 요구하게 되고 컴퓨팅 파워는 다시 하드웨어 장비에 대한 운영비용을 증가시킨다. 표 3과 같이 Bioinformatics 관련 업무에 필요한 하드웨어 장비의 매출은 Bioinformatics 기업의 매출을 초과한다[1].

4. Cloud Computing 도입을 통한 한계 극복

Bioinformatics의 성장한계를 극복하기 위한 방안은 데이터/알고리즘의 Integration과 IT 비용절감이라는 두 가지의 키워드로 요약이 가능하며 이를 위한 실천 방안을 찾기 위해 현재의 Bioinformatics시스템의 구조를 살펴볼 필요가 있다.

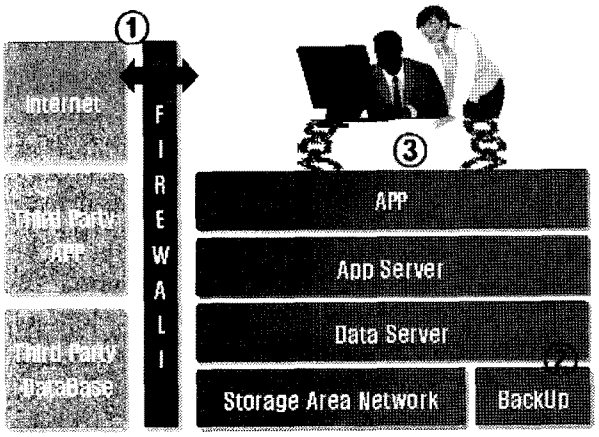


그림 4 현재의 Bioinformatics 시스템의 구조

그림 4는 현재의 Bioinformatics 시스템의 구조이다. 이 그림을 통해 앞서 설명한 Bioinformatics의 두 가지 한계, 즉 통합되지 않은 데이터/알고리즘과 하드웨어 비용 증가의 원인을 확인할 수 있다.

먼저 1번으로 표기된 지점을 살펴보면 Bioinformatics 애플리케이션과 타 기관의 DataBase간에 직접적인 연결고리가 존재하지 않는다. 따라서 Client가 직접 인터넷을 통해 공개된 데이터를 가져온 뒤, 기존에 설치한 애플리케이션에서 생성한 데이터와 통합해야만 한다.

2번으로 표기된 지점을 살펴보면 Storage의 구성이 Storage Area Network, 즉 다수의 Storage를 LAN이나 광케이블로 엮은 NAS, SAN의 형태로 이루어져 있다.

따라서 가상화를 통한 효율적인 Storage 사용이 제공되지 않으면 데이터의 증가에 따른 추가적인 Storage의 구매가 필수적이게 된다. 뿐만 아니라 데이터가 분산되어 저장되지 않으므로 Backup을 위한 별도의 소프트웨어와 장비 구매가 필요하다.

한편 3번으로 표기된 지점을 보면 Client가 사용하

는 애플리케이션에 타 기관의 Client가 접속할 수 없어 협업기능을 제공할 수가 없다. 또한 하나의 Client에 종속된 애플리케이션의 특성상 Client의 불필요한 요구사항까지 반영되어 소프트웨어의 유연성이 떨어지게 된다.

이러한 시스템 아키텍처의 문제점을 해결하기 위해서는 Broker 아키텍처 패턴과 Reflection 아키텍처 패턴, Rest 방식의 데이터 교환 인터페이스 그리고 하드웨어 인프라의 가상화 등이 필요한데 이러한 기능들은 Cloud Computing을 통해 구현이 가능하다.

4.1 데이터/알고리즘의 통합을 위한 Broker 아키텍처 패턴

Broker 아키텍처 패턴은 그림 5와 같이 Agent들이 자신들의 서비스를 Broker에 등록하고, 인터페이스를 통해 다른 Agent들이 Broker에 등록된 서비스를 사용할 수 있도록 한다.

Broker는 Agent간 요청을 전송할 뿐만 아니라 그에 대한 응답과 예외를 호출한 Agent에게 전송한다. 이를 위해 서버를 등록하는 오퍼레이션과 서버의 메서드를 호출하기 위한 API를 제공하게 된다[3].

Broker 아키텍처 패턴은 데이터의 통합을 위해 필요한 부분과 필요하지 않은 부분을 명확하게 구별하여 데이터의 교환에 관여된 Agent간의 효율적인 협업을 지원하므로 각 기관의 Bioinformatics 소프트웨어가 각 DataBase에 저장된 DNA, RNA, 단백질 데이터를 통합하는데 유용하다.

즉, Broker가 데이터 교환을 위한 표준을 인퍼테이스를 통해 정의한 후, API를 통해 각 기관의 Bioinformatics 소프트웨어가 호출할 수 있도록 함으로서, 다양한 기관의 DataBase가 하나의 양식으로 표준화되어 활용할 수 있다.

4.2 다양한 알고리즘의 적용을 위한 Reflection 아키텍처 패턴

Refection 아키텍처 패턴은 그림 6과 같이 시스템의 기능을 Basic Level과 Meta Level로 구분한다. Basic Level의 Component들은 애플리케이션 비지니스 로직을 포함하며 Meta Level은 시스템의 특성과 기본기능을 캡슐화 한다[3].

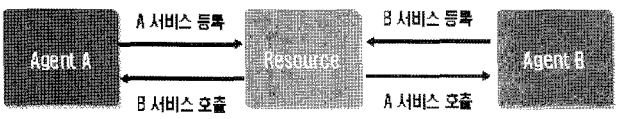


그림 5 Direct Communication 방식의 Broker 아키텍처 패턴

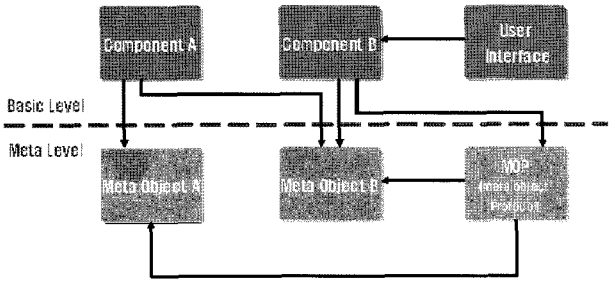


그림 6 Reflection 아키텍처 패턴

MOP(Meta Object Protocal)는 Basic Level에 있는 Component가 Meta Level에 있는 객체들은 변경할 수 있도록 인터페이스를 제공한다[3].

Reflection 아키텍처 패턴은 Bioinformatics의 다양한 알고리즘을 Meta Object로 각각 캡슐화하여 비지니스 로직을 수행하는 Component가 적합한 Meta Object를 선택하여 호출할 수 있도록 한다. 그리고 MOP에 정의한 방식대로 사용자가 알고리즘을 추가하고 변경하게 함으로서 소프트웨어의 변경을 용이하게 할 수 있다.

또한 Storage를 비롯한 컴퓨팅 파워의 제어기능을 Meta Object로 캡슐화 하여 비즈니스 로직을 담당하는 Component가 각각의 고유한 기능에만 전담할 수 있게 한다.

4.3 인터넷 기반의 데이터 교환을 위한 REST 전송방식

REST(Representation State Transfer)란 네트워크 시스템의 구조적인 형식을 표현하기 위한 용어로서, 그림 7과 같이 XML, HTML 등의 리소스(컨텐츠)를 URL을 식별자로 하여, 웹을 통해 배포하는 형식을 말한다. 이때 URL는 서비스 사용자가 URL 자체만으로도 서비스의 유형을 쉽게 파악할 수 있도록 체계적으로 쉽게 표현 되어야 한다.

Rest 방식은 앞서 설명한 Reflection 아키텍처 패턴과 Broker 아키텍처 패턴에 적용되어 데이터를 교환하고 소프트웨어간에 통신을 하는 것을 URL을 통해 간편한 웹 서비스의 형태로 가능하게 한다.

4.4. 하드웨어 자원 비용절감을 위한 가상화

기존의 하드웨어 자원들은 특정 IT 시스템에만 종속되어 운영되므로 가동률이 낮음에도 불구하고 여유자원을 다른 시스템에서 활용할 수가 없었다. 또한 추

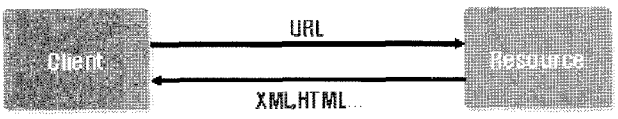


그림 7 Rest 방식의 데이터 교환

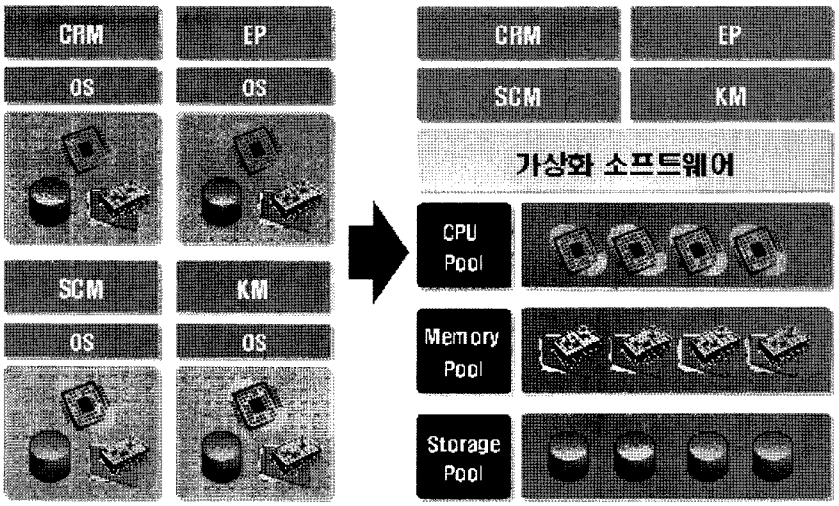


그림 8 가상화된 하드웨어 아키텍처

가적인 하드웨어 자원의 수요가 발생할 시에도 기존의 자원 여유분을 활용하지 못하고 많은 비용을 투입하여 자원을 구입하고 또 이를 관리하기 위해 비용을 추가해야만 했다.

이러한 문제점을 해결하기 위해 최근 도입되고 있는 것이 가상화이다. 가상화란 그림 8에서 보는 바와 같이 물리적으로 독립된 하드웨어 자원을 사용자가 필요한 만큼 논리적인 단위로 재배치하여 컴퓨팅 자원을 낭비 없이 효율적으로 활용하기 위한 기술이다.

가상화 기술을 통해 하드웨어 자원과 소프트웨어 자원 간의 종속성을 제거하고 동일한 속성의 하드웨어 자원들을 Pool로 엮어 낭비 없이 최대한 활용할 수 있다.

Mckinsey의 보고에 따르면 가상화와 서버분해를 통해 서버의 평균 가동률이 5.6%에서 9.1%로 증가하였다고 한다[4].

이는 추가적인 컴퓨팅 자원의 구매 없이 기존의 자원을 최대한 활용함으로서 비용절감의 효과를 가져 온다.

4.5 Broker 패턴 + Reflection 패턴 + Rest +가상화 = Cloud Computing

위에서 설명한 방식들은 Cloud Computing의 도입을 통해 동시에 적용할 수 있다.

그림 9는 대표적인 Cloud Computing 시스템인 Google의 App Engine의 구조이다[2].

그림의 맨 하단 Layer부터 확인해 보면 GFS와 Big-Table, 그리고 Memcashe는 하드웨어 자원을 분산 처리하여 애플리케이션의 신뢰성있고 빠른 실행을 보장한다.

또한 분산된 하드웨어 자원을 가상화를 통해 통합하여 효율적인 자원 활용이 가능하다.

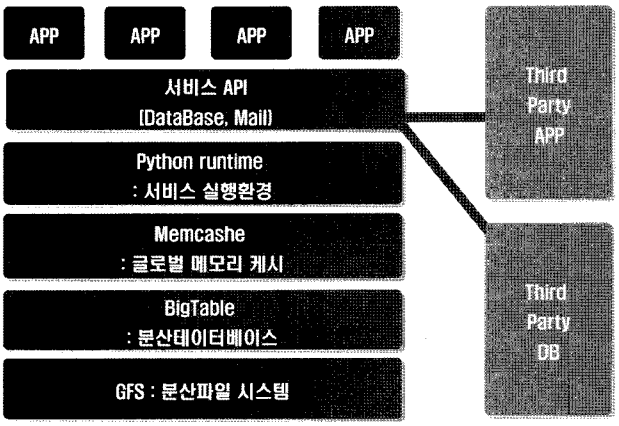


그림 9 구글의 App Engine

Python runtime 환경과 서비스 API는 Reflection 아키텍처 패턴과 같이 독립적이고 공유가 가능한 가능을 별도의 객체로 두어 애플리케이션의 유연성을 향상시키고 Broker 아키텍처 패턴처럼 서비스 API를 통해 외부 시스템들과 데이터를 연동 할 수 있게 한다. 이때 서비스 API는 REST방식으로 URL을 통해 제공되어 외부 시스템이 쉽게 데이터를 연동할 수 있다.

4.6 Cloud Computing 기반의 Bioinformatics 시스템으로의 전환

기존의 Bioinformatics 시스템의 구조를 Google의 App Engine의 구조에 대응해 보면 아래의 그림 10과 같다.

A 부분을 먼저 살펴보면 애플리케이션만 남고 서버는 사라졌다. 애플리케이션을 구동하기 위한 미들웨어와 DataBase는 Cloud Computing 내에 구축되어 있으며 사용자는 단지 이들에게 접속할 수 있는 가상의 이미지나 API 만 받아와서 애플리케이션에 연동하면 된다.

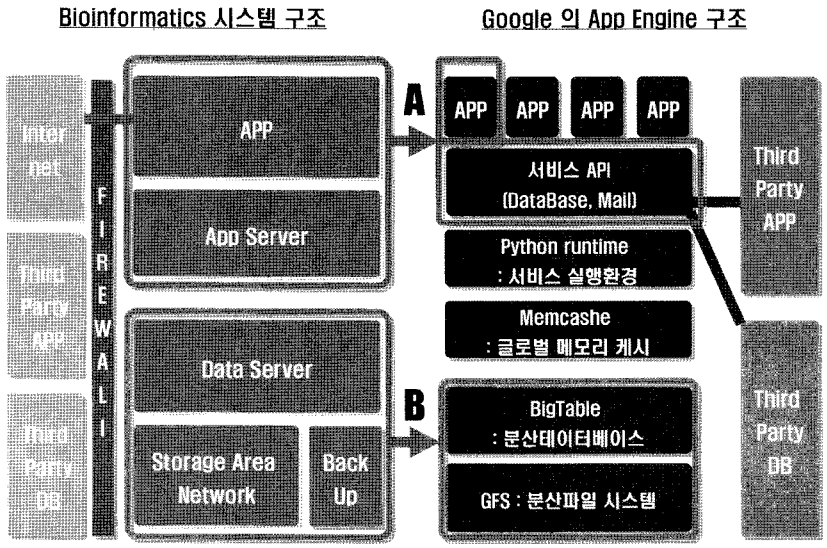


그림 10 Bioinformatics 시스템 구조와 Google의 App Engine 구조의 연관관계

타 기관의 애플리케이션 및 DataBase와 교류하는 방식도 인터페이스를 정의한 후 API의 형태로 사용할 수 있다. 즉 자신의 데이터를 가져갈 수 있는 API와 타 기관의 데이터를 가져올 수 있는 API를 통해 이기종 데이터 간의 교환양식을 단일화 할 수 있다.

또한 신약개발 R&D 프로세스의 변화에 따른 신속한 대처도 가능하다. 왜냐하면 DataBase 연결, 서버 설정, 언어 설정 등 기존의 애플리케이션에서 고려해야 할 시스템 공통적인 부분을 Cloud Computing 내의 공통 Component에서 설정 및 관리해 주어, 애플리케이션은 업무 프로세스 자체에 좀더 많은 관심을 둘 수 있기 때문이다.

B부분을 살펴보면 분산파일 시스템을 통해 파일이 저장되어 별도의 백업 시스템의 구축이 필요 없으며 가상화를 통해 전체 Storage를 재구성하므로 스토리지 증설에 대한 비용이 절감 된다. 또한 분산 DataBase는 기존의 관계형 DataBase의 확장성 문제를 해결한다.

이와 같이 Cloud Computing은 기존의 Bioinformatics 시스템의 두 가지 한계, 통합되지 않는 데이터와 알고리즘, 하드웨어 비용의 증가 문제를 해결할 수 있는 적절한 대안이 된다.

5. 결 론

Bioinformatics의 주요 고객 산업인 제약업의 Product는 의약품으로서 투여 받는 환자의 생명과 직접적인 영향이 있다. 따라서 신약의 개발을 위해서는 다양한 기관과 협업을 통해 신약 Target이 되는 단백질이나 DNA 등의 후보군을 함께 찾아내고, 이들에 대한 분석과 연구를 동시에 진행해야 한다.

본고에서는 이러한 기능을 제공하기 위한 방안으로 Cloud Computing 은 최선이 될 수 있음을 제시하였으며 그 이유를 요약하면 다음과 같다.

첫째, Cloud Computing은 데이터와 알고리즘에 대한 통합을 지원한다. Cloud Computing은 다양한 기관사이의 데이터 연계 표준을 제시하고 이를 바탕으로 인터넷을 기반으로 한 쉬운 데이터 연계를 지원한다. 또한 캡슐화를 통해 여러 기관들의 상이한 유전자분석 알고리즘을 적용할 수 있는 환경을 제공한다.

둘째, 가상화를 통해 데이터의 통합과 분자생물학의 발전에 따른 Storage의 급속한 증가에 효율적으로 대처할 수 있는 플랫폼의 역할을 한다.

Cloud Computing이 Bioinformatics 산업의 한계 극복을 위한 열쇠라는 것은 소프트웨어가 차세대 성장 동력 중에 하나인 바이오 신약개발의 성장을 위한 열쇠라는 것과 동일하다. 즉, 지금까지는 분자생물학 지식을 중심으로 소프트웨어 기술이 가미된 Bioinformatics 가 중요하였지만, 바이오 신약 개발의 비중이 확대되는 제약산업에서는 소프트웨어가 중심이 된 Bioinformatics 기술이 주도적인 역할을 할 것이다.

이를 위한 소프트웨어 업체의 전략은 Cloud Computing 을 바탕으로, DNA, RNA, 단백질의 분석 데이터 확보와 통합을 시도함으로써, 데이터와 알고리즘에 대한 접근 및 활용 우월성을 경쟁 우위의 원천으로서 확보하는 것이 될 수 있다.

이러한 경쟁 우위의 원천을 바탕으로 Bioinformatics 산업의 구매자인 바이오 제약업체에 대해 공급자 우위의 전략을 수행할 수 있을 것이다

참고문헌

[1] Bioinformatics Markets, Trimark Publications, p. 66, p. 86-87, 2009

[2] 한재선, "클라우드 컴퓨팅 플랫폼 기술", 월간 마이크로소프트웨어, p. 153, 2009.01

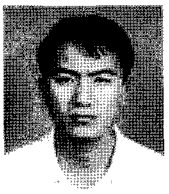
[3] Frank Buschmann, Pattern-Oriented Software Architecture, Wiley, 1996

[4] How to cut carbon emissions and costs, Mckinsey, p.10, 2008

[5] Geospiza, http://www.geospiza.com/cloud

김 성 권

1998 University of Maryland at College Park, B.S. Computer Science
1999~2000 IBM Global Service, IT Specialist
2000~2005 BearingPoint, Consultant
2005~2007 HP, Consulting Project Manager
2004~2006 MBA, Robert H. Smith School of Business - University of Maryland at College Park
2007~현재 삼성SDS 정보기술연구소 R&D 센터 미래기술전략그룹 책임연구원
E-mail : sungkwonomc.kim@samsung.com



박 용 민

2003 성균관대학교 정보공학과(학사)
2003~현재 삼성SDS 정보기술연구소 R&D 센터 미래기술전략그룹 선임연구원
E-mail : ym100.park@samsung.com