

기업의 정보처리 특성과 클라이언트-서버 아키텍처 구현 전략에 관한 연구

김 영 길[†] · 박 영 면^{††}

요 약

오늘날 많은 이들이 클라이언트-서버 컴퓨팅은 차세대 전산 환경의 주역이 될 것으로 인식하고 이미 여러 기업에서 정보시스템을 클라이언트-서버 환경으로 구축해 왔다. 하지만, 이러한 많은 관심과 노력에도 불구하고 만족할 만한 결과를 얻지 못하고 있는 실정이다. 이러한 회의적인 반응에 대한 원인으로서는 많은 요인들이 있겠지만 연구에서는 무엇보다도 기업의 다양한 정보처리 특성에 대한 면밀한 검토 없이 시대적 조류에 밀려 도입에 급급했다는 것을 지적하고 있다.

그래서, 본 연구에서는 다섯 가지 클라이언트-서버 아키텍처들을, 일곱 가지의 정보 처리에 관련된 기준들에 따라 각각 평가 분석해 보고 기업의 정보처리 특성과의 적합성을 분석하여 가장 이상적인 클라이언트-서버 아키텍처를 제안한다. 그리고 제안된 아키텍처를 기반으로 성공적인 정보 시스템이 구현될 수 있는 전략을 제시한다. 마지막으로 본 논문에서 제시된 상황모델의 검증을 위해 클라이언트-서버기술을 구현한 세가지 사례를 탐색적 수준에서 분석해 보았다.

Enterprise Information Processing Characteristics and Client-Server Architecture Implementation

Young-Gul Kim[†] · Young-Myun Park^{††}

ABSTRACT

Client-Server Computing is now recognized as the most powerful and flexible computing mode of the future. Many companies have already committed significant amount of their corporate IS resources to building the client-server computing environment. Despite this active interest and commitment, however, results from such client-server projects are far from being convincing. Among the many factors, we suspect that the less than satisfactory client-server experiences are perhaps due to the company's lack of understanding of its essential information processing characteristics.

We analyze the five client-server architectures in terms of seven IS-related criteria. We then develop a contingency model to match company's information processing characteristics to the ideal client-server architecture for the firm. Based on the client-server architecture analysis from the previous step, we recommend specific implementation strategies for each client-server architecture selected. Finally, on an exploratory level, we look into the three client-server implementation cases to check the validity of our contingency model.

† 중신회원: 한국과학기술원 테크노경영대학원
†† 정 회 원: 일성종합건설 기획조정실 전산팀장
논문접수: 1996년 9월 20일, 심사완료: 1997년 4월 25일

1. 서 론

경쟁환경의 변화와 정보기술의 발전이 오늘날 기업의 경쟁력 확보에 막대한 영향을 주고 있으며 기업내 정보시스템 환경도 규모의 경제에 입각한 종래의 중앙집중식에서 다양성의 경제에 기초한 분산 개방형으로 바뀌고 있다. 최근 추세에 의하면 1990년대 말에는 사실상 모든 신규 응용시스템이 분산컴퓨팅의 핵심 유형인 클라이언트-서버(Client-Server: C/S) 모델을 사용하여 개발될 것이라고 예측된다[19, 20]. 그동안 많은 기업들이 막대한 기업정보자원을 투자하여 자신들의 중앙집중식 정보시스템 환경을 클라이언트-서버 환경으로 전환하려는 시도를 했으나 결과는 당초 기대보다 훨씬 못 미치는 경우가 많았다[23, 24]. 이는 기업들이 다양한 C/S 아키텍처들의 특성과 자사의 업무 환경 및 정보처리특성에 대한 면밀한 분석없이 특정 도구나 특정방법론을 맹목적으로 적용했을 때 발생하기 쉽다. 이러한 기업들은 C/S 도구나 교육, 기존 데이터의 전환 등에 엄청난 투자를 마친 뒤에야 자신들의 새로운 C/S 시스템이 자사의 정보처리요구사항들을 만족시키지 못하거나, 만족시키더라도 기존의 시스템에 비해 엄청난 추가비용을 야기함을 깨닫곤 한다[23].

위와 같은 어려움과 많은 문제점에도 불구하고 국내의 정보시스템 분야의 문헌에서는 C/S 관련 하드웨어, 데이터베이스나 네트워크기술에 대한 연구는 활발한 반면 C/S 아키텍처의 선정이나 구현전략에 관한 이론적 체계나 실증적 연구는 극히 미미한 실정이다. 본 연구의 목표는 C/S 환경구축을 위한 실용적 방법론을 제시함으로써 이 분야의 향후 보다 정교한 실증연구에 필요한 가설이나 변수들을 발견함에 있다. 이를 위하여 먼저, 다양한 C/S 아키텍처들의 특성을 분석하고, 기업의 업무별 정보처리 특성에 따른 이상적인 C/S 아키텍처의 선정을 도와줄 C/S 상황모형을 개발한다. 다음으로 각 아키텍처 별로 바람직한 실제 구현전략을 제시한다. 마지막으로 C/S 환경을 구축한 선행기업들의 사례를 탐색적 차원에서 분석하여 제안 방법론의 타당성을 검증하고자 한다.

2. 클라이언트-서버 아키텍처의 개념적 체계

2.1 클라이언트-서버 아키텍처 구성요소

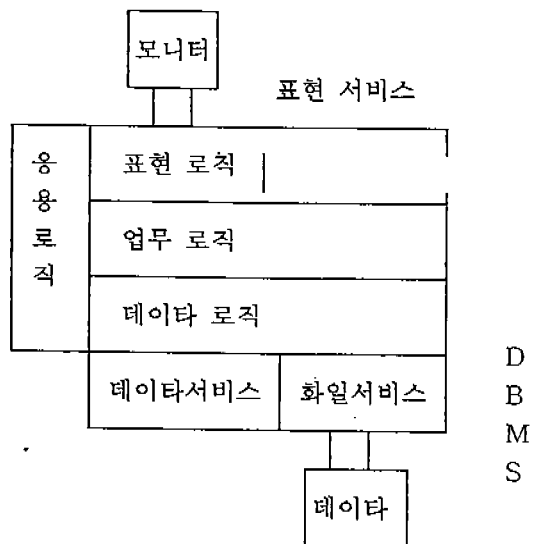
클라이언트-서버 컴퓨팅에서 가장 핵심기능이 협동처리(Cooperative Processing: CP)라고 판단할 수 있으며 Edelstein[18]은 이를 위해 (그림 1)에서와 같이 응용로직(Application Logic)을 분할하고, 각 로직들이 자신의 목적을 충족하기 위하여 시스템 내부적으로 실행하는 과정을 서비스로 정의했다.

표현 서비스: 사용자의 입력내용을 접수하는 입력장치들이나, 표현 로직이 의미하는 바를 사용자에게 전달해 주는 물리적인 출력장치들에 의한 행위를 뜻한다.

표현 로직: 사용자와 컴퓨터 사이의 상호작용을 통제하는 로직으로서, 사용자가 컴퓨터에게 작업을 요구하기 위하여 메뉴 옵션으로부터 메뉴를 고르고, 버튼을 눌러 주거나, 열거된 작업 리스트로부터 원하는 것을 골랐을 때 일어나는 현상을 조정하게 된다.

업무 로직: 응용 시스템이 구현하고자 하는 목적을 달성하기 위해 수행되는 계산이나 의사결정, 조정작업 등을 표현하는 로직으로서, 종업원 급여계산이나 주문을 접수한다던가, 대부 여부를 평가하는 등의 업무특성을 표현할 수 있는 로직을 의미한다.

데이터 로직: 업무 로직이 목적을 달성하기 위하여 데이터를 조정 통제하는 로직이다. 예를 들면, 관계형



(그림 1) 응용시스템 실행을 구성하는 요소 (Fig. 1) Components in executing the application

데이터베이스에서 'SELECT', 'UPDATE', 'INSERT' 등과 같은 SQL 문장을 말한다.

데이터 서비스: 데이터 로직이 뜻하는 즉, 데이터 조작이나 데이터정의, 트랜잭션 처리/복구 등과 같은 작업을 직접 수행하는 데이터베이스 관리시스템 (DBMS)의 행위를 말한다.

화일 서비스: DBMS의 역할에 따라 데이터 저장 장소를 찾고, 필요 데이터를 작업영역으로 가져오는 운영시스템 차원의 행위를 말한다.

2.2 클라이언트-서버 아키텍처의 분류

앞 절에서 정의한 구성요소를 기능별 특성에 따라 효율적인 배치로 상호처리의 형태를 갖춘 클라이언트-서버 아키텍처를 정립할 수 있으며, 정보시스템 설계자는 적절한 상호처리를 고려하여 클라이언트-서버 구조의 이점을 최대한 얻을 수 있는 시스템 구조를 갖추는데 목적을 두어야 한다[13]. (그림 2)에 앞에서 분류했던 로직이나 서비스 각각의 역할과 의미에 따라 분할 배치함으로써 '화일서버', '2단계 아키텍처', '3단계 아키텍처'로 분류하고, '2단계 아키텍처'는 다시 '원격 데이터'와 '원격 프리젠테이션', '로직 분할' 아키텍처로 분리했다[18].

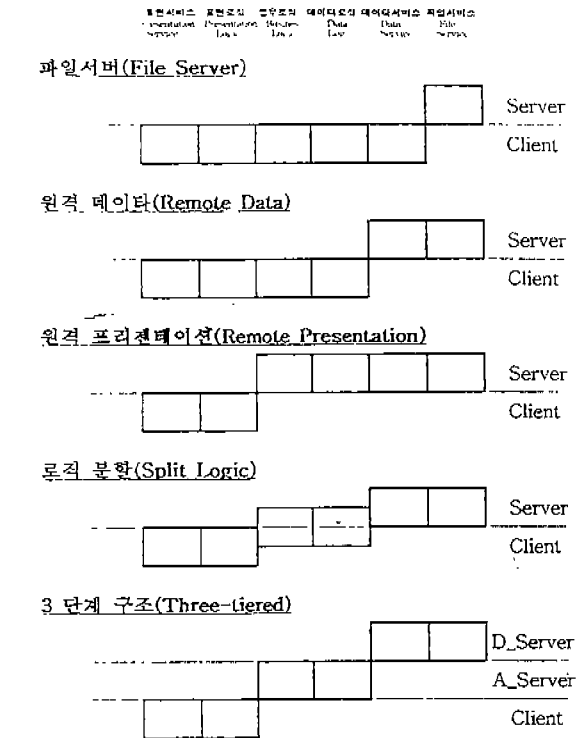
(1) 화일서버: 화일 서비스를 제외한 모든 기능이 클라이언트 측에서 진행되는 아키텍처로써 서버는 단순히 화일이나 데이터 저장장소로 이용되는 가상 장치일 뿐이다.

(2) 원격 데이터 C/S: 서버는 데이터 서비스와 화일 서비스 역할을 하고 표현 로직과 서비스, 응용 로직들은 클라이언트에서 작업하는 구조로서 현재 가장 보편화되어 있는 모델이다.

(3) 원격 프리젠테이션 C/S: 업무 로직과 데이터 로직, 데이터 및 화일 서비스가 서버에서 실행하며, 표현 로직과 표현 서비스만 클라이언트에서 수행되는 구조이다.

(4) 로직 분할 C/S: 표현 서비스를 전담하는 클라이언트와 데이터 서비스를 전담하는 서버에 응용로직 즉 업무 로직과 데이터 로직을 적절히 분배하는 구조로써 업무처리 특성이나 하드웨어, 데이터베이스 관리시스템 등과 같은 자원들의 지원정도에 따라 융통성을 발휘할 수 있는 아키텍처이다.

(5) 3단계 C/S: 서버를 데이터 서버와 응용 서버로 분



(그림 2) 클라이언트-서버 아키텍처(Adapted from edelstein, 1994)

(Fig. 2) Client/server architecture

리하여 데이터 서비스와 화일 서비스가 데이터가 상주하는 데이터 서버에서 작업하며 업무 로직과 데이터 로직은 응용 서버에서 실행된다. 그리고 표현 로직과 표현 서비스는 원격 프리젠테이션과 같이 클라이언트에서 실행하는 3단계 아키텍처를 갖는다[18, 20].

2.3 클라이언트-서버 아키텍처들의 분석

응용시스템 개발과 관리 그리고 반응시간, 투입자원, 데이터 안정성 등과 같이 시스템 성능면에서 다음과 같이 일곱 가지로 평가 기준을 분류했다.

- (1) 서버 기기의 확장성
- (2) 네트워크 부하
- (3) 클라이언트 기기의 부하
- (4) 응용시스템 관리
- (5) 개발 및 테스트의 용이성
- (6) 데이터 관리

(7) 자원 투입 요건

(1) 서버 기기의 확장성

클라이언트의 양적 증가에 따라 서버가 받는 영향을 평가하는 것으로 클라이언트의 증가로 늘어나는 서비스 요구에 서버가 받는 부하는 업무처리나 트랜잭션이 서버의 후처리계(Back end engine)의존도에 따라 평가가 가능하다. 따라서 클라이언트의 의존도가 상대적으로 클수록 확장성이 좋다고 판정할 수 있다.

(2) 네트워크 부하

네트워크 트래픽, 데이터 충돌, 병목현상 등과 같이 네트워크의 지원 한계로 인한 성능저하를 말하며 일반적으로 네트워크 전송데이터 량이 클수록 부하가 크게 된다.

(3) 클라이언트 기기의 부하

업무처리가 클라이언트에 많이 의존할수록 고성능 데스크 탑을 요구하며 (1)항의 확장성과 의미적으로 대칭 되는 평가 요소로 볼 수 있다.

(4) 응용시스템 관리

응용시스템 유지/보수차원에서 쉽게 처리할 수 있는 가능성을 파악하는 것으로써 응용시스템이 클라이언트에 많이 분포되어 있거나, 2단계 C/S 아키텍처와 같이 클라이언트측과 서버측으로 양분되어 있으면 상대적으로 응용시스템에서 발생한 문제해결이나 관리에 어려움이 있다.

(5) 개발과 테스트의 용이성

응용시스템개발과 테스트에 대하여 용이성을 평가하는 것으로, 응용 로직 설계시 클라이언트와 서버로 분산되는 양이 많고 복잡할 수록 개발이나 테스트에 어려움이 따른다.

(6) 데이터 관리

데이터 운용과 유지관리상의 어려움을 측정하는 기준으로 이상적인 데이터 관리는 복잡한 처리과정에서도 데이터는 일관성, 무결성이 유지되어야 한다.

(7) 자원 투입 여건

시스템을 성공적으로 구현하기 위해서 필요한 시간이나 비용 또는 인력, 공간 등과 같은 관리적 지원 요소들이 요구되는 수준에 따라 평가하는 기준이다. 그러므로 설계와 분석이 까다로운 '2단계'나 '3단계' 아키텍처는 많은 시간과 인력을 요하게 된다.

평가기준에 대한 아키텍처의 분석 결과를 <표 1>에

요약했다. 이러한 분석결과는 개별 아키텍처의 특성을 개념적 차원에서 상대평가하였으므로 보다 정교한 실증연구를 통하여 추후 추가검증 및 보완되어야 할 필요가 있다.

<표 1> 클라이언트-서버 아키텍처 분석

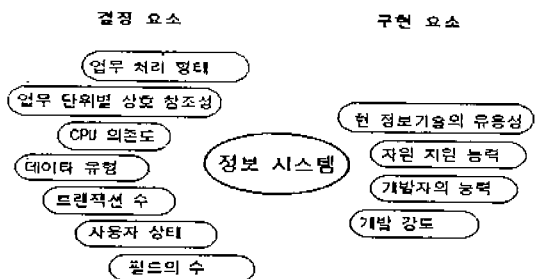
<Table 1> The analysis of client/server architecture

아키텍처 기준	파일 서버	원격 데이터	원격 프리젠테이션	로직 분할	3단계 구조
서버의 확장성	높음	높음	낮음	중간	중간
네트워크 부하	높음	중간	낮음	중간	중간
클라이언트의 부하	높음	높음	낮음	중간	낮음
응용시스템 관리	어려움	어려움	쉬움	중간	쉬움
개발, 테스트의 용이성	쉬움	쉬움	중간	어려움	어려움
데이터 관리	어려움	중간	쉬움	쉬움	쉬움
자원 투입 요건	낮음	중간	중간	높음	높음

3. 업무 환경과 정보처리적 특성

3.1 기업 정보 처리에 영향을 주는 요소 분석

기업의 정보처리적 특성에 영향을 줄 수 있는 환경 요소를 C/S 구현에 관한 문헌들을 중심으로 간추려 (그림 3)과 같이 열 한 가지로 구분했으며, 각 요소를 정보처리에 기여하는 용도에 따라 클라이언트-서버 아키텍처 선정에 영향을 주는 결정요소와 실제 구현시 고려해야 할 구현 요소로 구분하였다[2, 11, 12]. 이 요소들은 기업의 정보처리에 영향을 주는 모든 요소



(그림 3) 기업 정보처리 환경요소

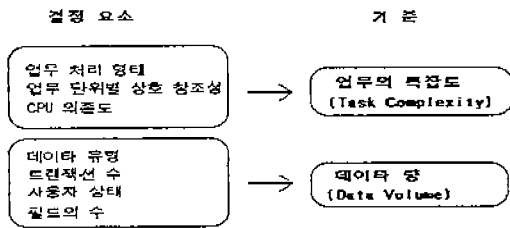
(Fig. 3) The environmental components in corporate information processing

를 포함하는 것은 아니며 요인분석과 같은 실증적 연구를 통하여 추출된 것이 아니므로 추후 추가 검증 및 보완이 필요하다고 보겠다.

3.1.1 결정 요소

업무처리 방법이나 단위 시간당 트랜잭션(Trans-action) 수, 데이터 양 등과 같이 업무에 대한 기본적인 고유한 속성을 가짐으로써 변동이 어려워 정보 시스템이 증속될 수밖에 없는 요소들이다.

(그림 4)에서 보듯이 결정요소를 ‘업무 복잡도’와 ‘데이터 량’으로 함축시켜 업무에 대한 정보처리적 특성을 분류하고자 한다.



(그림 4) 결정 요소와 기준
(Fig. 4) Determinants and criteria

(1) 업무 복잡도

업무 기능과 응용 시스템간의 기본적인 관계로 정의할 수 있으며 다음과 같은 요소를 내포한다.

업무처리 형태: 업무처리 과정을 말하며 일시에 많은 작업이 수행되는 배치성이 많거나 처리과정이 고정되어 있지 않은 비정형성을 띄면 업무 복잡도는 높아진다.

중앙처리장치 의존도: 업무 처리가 중앙처리장치(CPU)에 분담하는 부하를 의미하며, CPU 사용량이 많거나 작업이 시작되어 결과를 받기까지의 시간이 클수록 업무 복잡도는 높다.

상호참조: 업무 단위작업 그룹간 상호 참조적 작용이나 공유 데이터의 수준을 나타내는 요소로서 일반적으로 단위 작업간에 발생하는 업무 협조 사항들이 많거나 타 업무의 데이터를 조작해야 할 경우가 자주 발생하면 높게 된다.

(2) 데이터 량

보통 서버에 분포되어 있는 데이터를 클라이언트가 요구하는 방법, 처리 형태, 입/출력 빈도 그리고 네트워크에 실려 있는 데이터 량 등에 따라 다음과 같은 요소를 갖는다.

데이터 유형: 데이터 처리 방법이나 형태에 따른 기준으로 텍스트 데이터 보다는 이미지 데이터가 높고 데이터의 크기나 수정요구가 빈번히 발생하면 데이터 량이 높다.

트랜잭션의 수: 단위 시간당 데이터를 액세스하여 조작하는 빈도 즉, 데이터 트랜잭션의 빈도에 따라 많으면 높고 작으면 낮다.

사용자의 상태: 사용자의 위치나 환경에 대한 요소로써 사용자 수가 많거나 원격 작업일 경우 네트워크에 흐르는 데이터 량에 따라 시스템에 영향을 주게 된다.

필드의 수: 논리적으로 객체(Entity)를 표현하는 속성(Attribute)의 수로써 많을수록 데이터 량이 높다.

3.1.2 구현요소

업무처리 특성을 반영한 선택적 요소와는 달리, 주변 환경을 주로 내포하고 있는 요소들로서 중앙집중식 환경에서 정보시스템 관계자 및 최고 경영자에게 조성되어 있는 마인드나 H/W, S/W 환경이 클라이언트-서버 컴퓨팅에서는 큰 변화를 겪게 되므로 잘 관리되지 않을 경우에는 특정 아키텍처의 선정에 제약 사항으로 작용할 수도 있다.

현존정보 기술의 유용성: 현존하는 기업내 정보기술의 용량이나 활용 가능성을 뜻하며 특히 현재 보유 중인 서버나 클라이언트 자원의 활용가능도가 선정된 아키텍처의 구현에 영향을 미칠 수 있다. 예를 들어 원격데이터 C/S 아키텍처가 선정되었을 때 현재 보유중인 수백대의 클라이언트 PC를 상위기종으로 교체해야 한다면 구현과정상의 저항이 클 수도 있다.

개발자들의 능력: 개발인력들의 능력으로써 시스템 구축 성패를 좌우할 수 있는 중요한 요소이다. 기존의 시스템 개발이 단순한 개발언어에 대한 지식 위주였다면 C/S환경의 구현은 분산데이터베이스, 네트워크, 트랜잭션 모니터링, 그래픽 개발툴 등에 관한 새로운 지식과 경험의 보유가 필수적이다.

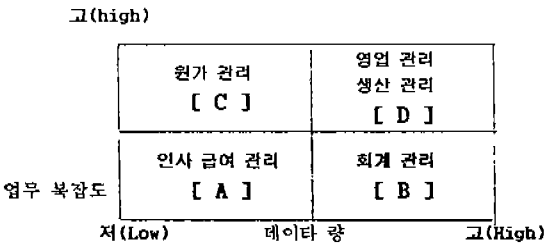
개발 강도: 시스템 구현 목적에 따라 프로젝트별로 개발하고자 하는 범위, 개발수준, 개발 주체에 대한

내/외부 적용비율 등과 같이 개발대상의 한계나 처리 과정을 결정하는 것이다. 일반적으로 개발강도가 높을수록 개발프로젝트의 위험과 구현단계에서의 저항이 높아진다고 볼 수 있다.

자원지원여력: 개발기간이나 비용, 하드웨어, 통신 기반, 내부 마인드 등과 같이 관리적인 자원들의 지원 능력을 의미한다. 로직분할구조나 3단계 C/S 구조로 전사적 거래처리시스템을 구현할 때에는 충분한 관리적 자원지원이 필수적이다.

3.2 업무의 정보처리적 환경 분류

전 절의 기업환경에 영향을 주는 요소중 업무처리 특성을 나타내는 결정 요소를 중심으로 (그림 5)와 같이 업무에 대한 정보처리 특성을 네 가지로 분류했다. 여기서 업무별 업무복잡도와 데이터량의 높고 낮음의 판단은 기업마다 또 각 기업이 속한 산업마다 절대적 수치가 다르기 때문에 정보기술 전문가들과의 패널토의를 통하여 상대적 분류기준을 따라 정했다.



(그림 5) 업무처리 특성
(Fig. 5) The characteristics of information processing

(1) Type A

응용시스템이 CPU에 많은 부하를 주지 않는 비교적 단순한 기능을 가지며 사용자 수도 작고 데이터 조작이 드문 환경으로 정형성이며 소규모 업무를 뜻한다.

(2) Type B

응용시스템의 정보처리 과정이 간단하여 중앙처리 장치에 주는 부하가 낮고 많은 처리시간을 요하지 않지만 데이터 볼륨은 크고 사용자는 많아 데이터와 관련된 작업이 매우 빈번한 특징을 갖는다

(3) Type C

응용시스템의 처리가 매우 복잡하여 중앙처리장치에 많은 부하를 주며 시스템 로직간 상호 참조비율은 높지만 데이터 량이나 데이터 참조율 즉, 데이터 조작행위가 작은 업무로써 복잡하고 비정형적이며 사용자 수가 작은 특징을 갖는다.

(4) Type D

다양한 사용자 환경과 복잡한 업무로직으로 CPU에 큰 부하를 주며 데이터에 대한 입/출력, 수정, 삭제, 삭제 등의 요구도 매우 빈번하고 데이터 량도 큰 방대한 처리를 해야하는 업무를 뜻한다.

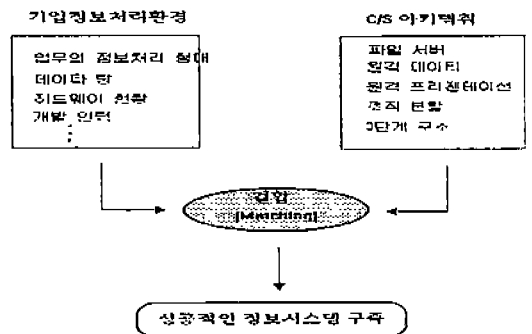
클라이언트-서버 아키텍처가 효율적인 시스템으로 평가 받기 위해서 작업처리 유형에 따라 CPU중심형 작업처리와 데이터 입/출력중심형 작업처리로 구분하여 하나의 조직체 또는 동일 업종의 기업체에서 공통으로 이루어지는 전산화 대상 업무별로 상대적으로 평가하는 것이 보다 현실적일 것이다.

4. 기업 환경과 클라이언트-서버 아키텍처

4.1 기업 정보 시스템 구축 전략

기업 환경분석을 위해서는 각 기업이 갖추고 있는 정보시스템의 상태 및 사업상 필요사항들을 파악하고 그 기업의 경영전략 요소 및 정보기술의 비전을 세워 가장 알맞고 유연성 있는 시스템을 유도해 내는 체계적인 접근 방법이 필요하다.

(그림 6)과 같이 제 2 장에서 연구한 클라이언트-서버 아키텍처 평가 결과와 제 3 장에서 연구한 기업 정



(그림 6) 시스템 구축 방향
(Fig. 6) The guideline for implementing the information system

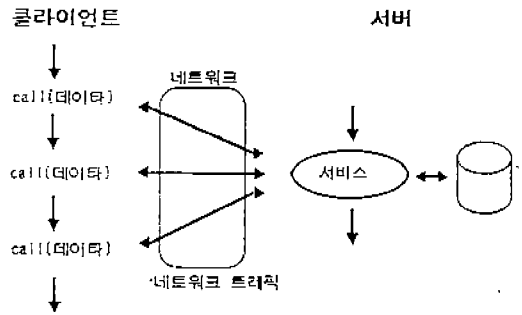
보처리환경에 따른 업무적 특성을 분석하여 최적의 클라이언트-서버 아키텍처를 도출함으로써 성공적인 정보시스템을 구축하는 방향을 찾고자 한다.

4.2 정보처리 특성과 아키텍처 설계의 기본 개념

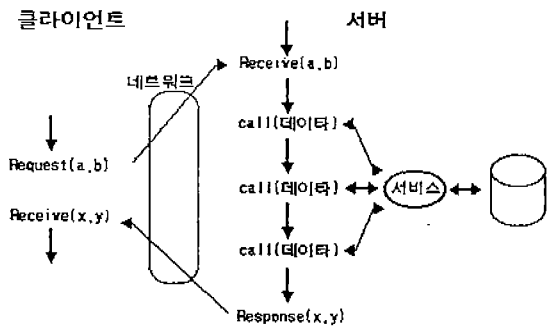
업무의 정보처리적 특성을 표현해 주는 두 가지 독립변수(데이터 량, 업무의 복잡도)를 고려해 볼 때, (그림 7)에서 보듯이 데이터 량으로 명명된 X축이 상승하게 되면 데이터와 관련된 작업이 매우 많아지는데, 만약 (그림 8)의 a 경우와 같이 데이터가 상주하고 있지 않는 클라이언트에서 어플리케이션 로직(특히 데이터 로직)이 실행된다면 빈번히 일어나는 데이터 액세스 명령들이 일일이 네트워크를 통하여 서비스 요구와 응답이 서로 전송되므로 다수의 클라이언트로부터 요구가 빈번할 때, 네트워크 트래픽으로 인하여 시스템 성능이 떨어진다. 게다가 일부 데이터를 선택하여 참조하는 경우는 선택된 데이터가 의사결정로직이 있는 클라이언트 측으로 화일 단위로 전송되어 병목현상이 유발되기도 한다. 그러므로 이 경우(X축으로 상승)의 아키텍처 설계는 (그림 8)의 b와 같이 데이터와 관련된 로직들은 서버에서 실행할 수 있게 해야 한다.

한편, 업무 복잡도를 기준으로 한 Y축은 높은 수준으로 갈수록 CPU에 많은 부하를 주며, 처리시간이 길고, 응용시스템이 매우 복잡해지는 특성을 가진다. 따라서 데이터와 데이터 서비스가 함께 있는 서버에서 복잡하고 많은 시간을 요하는 응용시스템(업무로직, 데이터 로직)이 실행되면 (그림 8)의 a와 같이 서

버의 부하로 다수의 클라이언트로부터 온 서비스 요구가 적체되어 시스템능력이 저하되며, 서버의 확장성 문제가 발생하게 된다. 그러므로 Y축으로 커 가는 업무일수록 (그림 8)의 b와 같이 복잡한 응용시스템을 클라이언트 각자가 처리하여 시스템 처리효율성을 높일 수 있게 설계해야 한다.

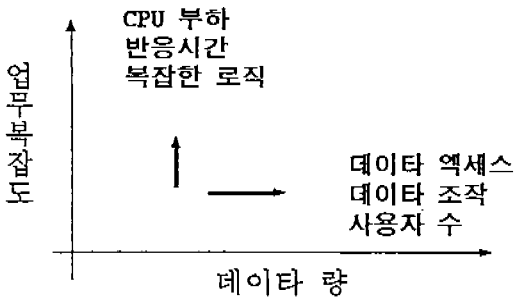


a. 어플리케이션 로직이 클라이언트에서 실행



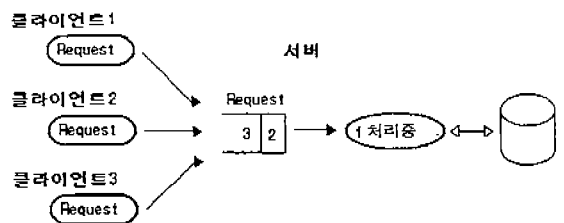
b. 어플리케이션 로직이 서버에서 실행

<그림 4-3> 데이터 량에 따른 위치

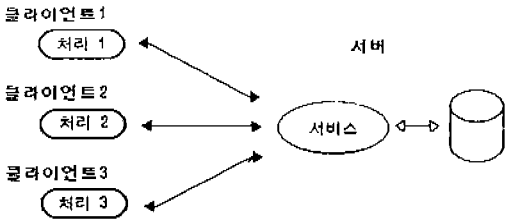


(그림 7) 업무의 정보처리적 특성

(Fig. 7) The characteristics of information processing of tasks



a. 서버에서 응용시스템 실행



b. 클라이언트에서 응용시스템 실행

(그림 8) 업무 복잡도에 따른 위치

(Fig. 8) The location of application processing according to the complexity of task

4.3 클라이언트-서버 설계 전략

3.2절에서 분류한 네 가지 기업의 업무처리 환경과 2.3절의 클라이언트-서버 아키텍처들의 분석 결과, 그리고 전 절의 정보처리 특성과 아키텍처 설계의 기본개념에 따라 가장 적합한 모델을 유도했다. 먼저, 소규모의 업무적 특성을 갖는 'Type A'에는 적은 비용으로도 쉽게 구축할 수 있으며 어플리케이션 관리 면에서 좋은 '화일 서버' 아키텍처가 적합하다. 물론 '2단계'나 '3단계' 아키텍처로도 시스템 성능을 훨씬 뛰어나게 구축할 수 있지만 효율성(투자대비 효과)에서 떨어지게 된다.

'Type B'는 업무처리가 데이터와 매우 밀접하여 빈번한 입/출력을 요하므로 네트워크 최적화와 데이터 안정성이 요구된다. 따라서 물리적으로 데이터가 존재하고 있는 서버측에 데이터와 관계 있는 로직과 서비스가 함께 공존하는 것이 적합할 것이며 CPU 의존도가 낮은 특성이 있으므로 서버에서 업무로직이 실행되어도 큰 부담을 주지 않게 된다. 그러므로 네트워크 부하와 데이터 안정성이 양호하고 서버 의존율이 높은 '원격 프리젠테이션 C/S' 아키텍처가 가장 적합한 모델로 평가된다. 그리고 업무로직이나 데이터 로직을 합리적으로 분할할 수 있는 설계자와 강력한 데스크 탑이 지원되면 '로직 분할 C/S' 아키텍처로 더욱 효율적인 시스템을 구축할 수도 있다.

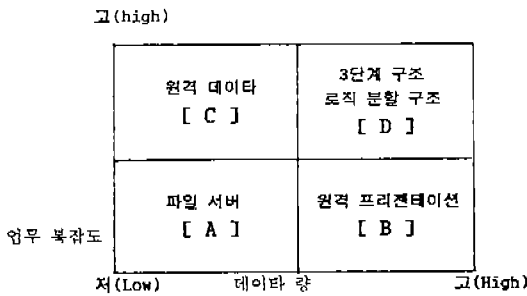
'Type C'는 'Type B'와 논리적으로 대칭 되는 개념으로 비교적 응용시스템이 CPU에 주는 부담이 매우 크고, 데이터 입/출력 또는 수정 등과 같이 데이터 제어요구는 비교적 작으며 업무처리상 사용자 수가 작은 특징을 갖고 있으므로 네트워크 부하나 데이터 안

정성과 같은 조건은 상대적으로 의미가 없고 CPU의 부하나 반응시간을 검토해야 한다. 따라서 서버측에 데이터베이스 관리시스템과 응용시스템이 함께 공존하여 실행될 경우, 서버의 부하로 시스템 성능이 하락할 수 있고 네트워크를 통한 데이터 전송량이 많지 않으므로 응용시스템은 클라이언트측에서 개별로 처리하는 것이 바람직하다. 그러므로 '원격 데이터 C/S' 아키텍처가 적합하며 이런 업무에도 네트워크 환경이나 인력과 서버의 성능에 따라 '로직 분할 C/S' 아키텍처로 시스템 효율성을 더욱 높일 수 있다.

'Type D'는 생산관리, 영업관리 등 다양한 사용자 환경과 복잡한 업무로직으로 CPU에 큰 부하를 주며 동시에 데이터에 대한 대화식 입/출력, 수정, 발취, 삭제 등의 요구도 빈번하여 처리해야 할 데이터량도 방대한 업무특성을 가진다. 이 경우 파일서버나 원격프리젠테이션 아키텍처는 서버의 부하가중으로 서버성능저하를 가져올 수 있고 원격 데이터 아키텍처의 경우는 강력한 클라이언트 기기를 필요로 하나 소수 사용자를 가지는 원가분석 (Type C)업무와는 달리 수백 내지 수천명의 사용자를 가지므로 경제성을 가지기 어렵다. 따라서 '로직분할 C/S' 아키텍처를 활용하여 복잡한 업무 로직을 표현 서비스 및 DBMS-독립적인 처리를 담당하는 전 처리 부분과 DBMS를 담당하는 후 처리부분으로 분리시켜 응용시스템을 분할처리케 하는 것이 바람직하다. 그러나 지원가능한 자원(H/W, DBMS등..)상황에 제약이 있어 이기종 분산환경에서 분할로직을 설계할 만한 인력이 부족하거나 분산된 클라이언트 관리비용을 감당할 수 없을 경우에는 중앙집중식 응용시스템관리가 훨씬 효과적이면서도 응용서버와 데이터서버가 분리되어 서버부하가 통제가능한 '3단계 C/S' 아키텍처가 바람직하다. (그림 9)에 적합한 아키텍처를 표시했다.

이상과 같이 업무특성과 최적 아키텍처의 선택이 이루어지게 되면 성공을 위한 구현요소들의 여건이 다음 과제로 남는다.

먼저, 'Type A'에 적합한 모델로 평가된 '화일서버' 아키텍처는 파일서비스를 제외한 모든 처리가 클라이언트에서 이루어지므로 강력한 클라이언트측 기기를 필요로 하며 네트워크 부하가 많이 걸리므로 네트워크 최적화와 클라이언트 상에서의 응용시스템 및 데이터 관리가 보장되는 것이 바람직하다. 'Type B'



(그림 9) 업무처리 특성과 C/S 아키텍처
(Fig. 9) The characteristics of tasks and C/S architecture

의 ‘원격 프리젠테이션’ 아키텍처는 서버에서 많은 양의 작업을 하게 되며, 매개변수단위로 데이터를 전송하므로 강력한 서버 기기와 우수한 인력을 보강하는 것이 관건이며, ‘Type C’는 클라이언트에서 많은 작업을 하는 ‘원격 데이터’ 아키텍처로서 클라이언트 기기의 성능이 충분한지 검토해야 하며 또한 각 클라이언트에서 응용시스템이 저장 및 사용되므로 응용시스템 관리의 최적화에도 유의해야 한다. ‘Type D’의 ‘로직 분할’과 ‘3 단계’ 아키텍처는 클라이언트와 서버로 응용시스템을 분할하거나 응용서버와 데이터서버 간의 실시간 트랜잭션 처리를 위해서는 아직 일반화되지 않은 트랜잭션처리관리기(TP monitor) 등의 미들웨어에 익숙한 개발 및 네트워크 관리 전문가의 확보가 성패를 좌우할 수 있으며, 타 아키텍처들에 비해 자원투입요건이 높은 편이므로 전사적 차원의 충분한 관리적 자원지원이 필요하겠다. 또한 ‘분할 로직’ 아키텍처에서는 응용시스템의 일부가 클라이언트에서 처리되므로 응용시스템 관리에 대한 대책과 비교적 강력한 클라이언트 기기가 필요하며, ‘3단계’ 아키텍처에서는 응용서버와 데이터서버 간의 대용량 데이터 전송을 무리없이 처리할 수 있는 강력한 네트워크 기반 조성이 필요하다.

결론적으로, <표 2>에 업무처리 특성에 따라 전략적 차원에서 최적 클라이언트-서버 시스템을 정의했으며, 각 아키텍처의 특징을 고려하여 단점을 보완할 수 있는 적절한 환경조성과 기업목표에 따라 수립된 경영전략에 일치하는 치밀한 정보시스템 구축계획을 수립함으로써 경쟁력 제고의 기반을 다져 놓는 것이 더욱 중요하다.

<표 2> 클라이언트-서버 시스템 구현 전략
(Table 2) The strategy of implementing C/S system

구분 전략	업무처리상 특징 (선택적 요소)		클라이언트-서버 아키텍처	구현 요소
	데이터 량	업무 복잡도		
1	저	저	파일 서버 (File-Server C/S)	강력한 클라이언트 기기 네트워크 최적화 데이터 관리 강화 응용시스템 관리 강화
2	고	저	원격 프리젠테이션 (Remote-Pres- entation C/S)	고성능 서버기기 개발자 능력 향상
3	저	고	원격 데이터 (Remote-Data C/S)	강력한 클라이언트 기기 응용시스템 관리 강화
4	고	고	로직 분할 구조 (Split-Logic C/S)	개발자 능력 강화 외부 전문가 자문 강력한 클라이언트 기기 충분한 관리적 자원 지원
5	고	고	3 단계 구조 (Three-Tiered C/S)	개발자 능력 강화 외부 전문가 자문 네트워크 최적화 충분한 관리적 자원 지원

5. 사례분석

이 장에서는 C/S 아키텍처를 구현한 세 기업의 사례를 통하여 앞에서 제시했던 상황모델의 타당성을 탐색적 수준에서 검증해 보고자 한다. 잘 정리된 가설의 통계적 검증도 아니고 발생가능한 외생변수들에 대한 통계도 전혀 이루어지지 않은 채 사후적으로 사례를 분석하는 것이므로 본 사례들의 결과나 상황모델을 적용한 해석은 일반화하기 어려움을 밝혀둔다.

5.1 A은행

A은행 업무는 계정계와 정보계로 구분하여 정보시스템이 구축되었으며 <표 3>에 각 업무별 특성과 본 연구 결과에 따른 제안 모델, 그리고 현재 구현되어 있는 현황을 나타냈다.

A은행 계정계 정보처리는 하루 평균 45만건의 거래가 발생하는 많은 트랜잭션이 주로 데이터 입, 출력 및 수정 중심의 정형화된 단순반복형으로 처리되

〈표 3〉 A은행 업무처리 특성과 현황
 (Table 3) The characteristics of A bank's information processing environment

업무	업무 특성	정보처리모델	구현 아키텍처
계정계	<ul style="list-style-type: none"> · 트랜잭션: 45만건/일 · 데이터 입/출력 중심의 단순 반복형 · 온라인 처리가 많음 · 데이터 안정성과 빠른 응답 속도 요구 	<ul style="list-style-type: none"> · Type B · 원격 프리젠테이션 (Remote-Presentation C/S) 	<ul style="list-style-type: none"> · 로직 분할 (Split-Logic C/S)
정보계	<ul style="list-style-type: none"> · 비 정형성 · 조건검색이 많음 	<ul style="list-style-type: none"> · Type C · 원격 데이터 (Remote-Data C/S) 	<ul style="list-style-type: none"> · 로직 분할 (Split-Logic C/S)

며 클라이언트의 7할이상이 이 작업에 종사하고 온라인처리가 60%를 차지하는 특성을 갖는다. 또한 주로 고객대상 업무이므로 완벽한 데이터 안정성과 빠른 응답이 요구된다. 그러므로 업무처리 특성모델에서 'Type B'형에 속하는 업무로 볼 수 있다. 한편 정보계 업무는 데이터 검색이 빈번하지는 않지만 조건에 따라 광범위하여 CPU에 많은 부하를 줄 가능성이 크다. 업무도 비정형적이며 조건 검색량이 상대적으로 많은 특성을 가진다. 따라서 'Type C'와 같은 업무로 분류된다. 따라서 계정계와 정보계 업무는 각각 '원격 프리젠테이션'과 '원격 데이터' 아키텍처가 적합한 모델로 판정된다.

그러나 현재 A은행 시스템들은 제안된 아키텍처보다 복잡하지만 기능이 뛰어난 '로직 분할' 아키텍처로 구축되어 있어 성능상 별 무리없이 두 가지 업무를 적절히 운영하고는 있지만, 그러한 응용시스템 및 네트워크 구축에 엄청난 투자와 노력이 투입되었다는 점에서 투자 대비 효율성측면에서는 재 분석이 필요하다. 예를 들어, 서버의 CPU 사용율을 보면 보통 30% 선을 유지하고 있어 고성능의 서버기기(HP, Sun)를 충분히 활용 하지 못하고 있음을 알 수 있다.

5.2 B사

B사는 1945년 설립된 연간 매출액 2000억 규모의 중견제조업체로서 1000여종의 제품을 생산-출하-유통하는 다품종 소량생산체제로 인해 매출액에 비해 처리하는 정보량은 매우 많은 회사이다. 정보시스템환

경은 1984년부터 IBM 4331을 도입하여 종합정보시스템 구축 후 IBM 4361로 전이시켜 중앙집중형 텍스트모드로 운영해 왔었다. 91년 4월부터 다운사이징 검토로 시작된 신 시스템(클라이언트-서버) 구축 프로젝트는 도구(tool)와 네트워크 운영시스템(NOS)을 몇 번씩 바꾸는 시행착오를 거친 끝에 개발이 완료되어 현재 모든 시스템이 구동중에 있다. 시스템 구성상의 특징은 데이터 조작 속도 증진을 위해 DBMS가 지원하는 저장 프로시저를 많이 사용하였으므로, 의견상으로는 '로직분할' 아키텍처를 채택한 듯하나, 실질적으로는 거의 대부분의 데이터로직과 업무로직이 서버에서 실행되고 프리젠테이션 로직과 서비스는 클라이언트에서 실행되는 '원격 프리젠테이션' 아키텍처를 갖추고 있다.

B사의 정보시스템은 인사/급여, 회계, 생산/자재, 원가, 영업관리 등으로 구분하여 구축되었으며 각 업

〈표 4〉 B사 업무처리 특성과 현황
 (Table 4) The characteristics of B company's information processing environment

업무 [사용자 만족도]	업무 특성	정보처리 모델 추천 아키텍처	구현 아키텍처
인사/급여 [95%]	<ul style="list-style-type: none"> · 단순 반복형 · 데이터 량: 적다 · 정기성 	<ul style="list-style-type: none"> · Type A · 파일 서버 (File-Server) 	<ul style="list-style-type: none"> · 원격 프리젠테이션 (Remote-Presentation C/S)
회계 [97%]	<ul style="list-style-type: none"> · 정형화된 단순 반복 · 데이터 조작이 빈번 	<ul style="list-style-type: none"> · Type B · 원격 프리젠테이션 (Remote-Presentation C/S) 	<ul style="list-style-type: none"> · 원격 프리젠테이션 (Remote-Presentation C/S)
생산/자재 [67%]	<ul style="list-style-type: none"> · 데이터 량: 크다 · 사용자 수: 많다 · 작업량이 많다(온라인, 배치작업) · CPU 부하가 크다 	<ul style="list-style-type: none"> · Type D · 로직 분할 (Split-Logic C/S) 	<ul style="list-style-type: none"> · 원격 프리젠테이션 (Remote-Presentation C/S)
원가 [70%]	<ul style="list-style-type: none"> · 타 시스템 참조성이 높다 · 배치성 작업이 많다 · CPU 부하가 크다 	<ul style="list-style-type: none"> · Type C · 원격 데이터 (Remote-Data C/S) 	<ul style="list-style-type: none"> · 원격 프리젠테이션 (Remote-Presentation C/S)
영업 [60%]	<ul style="list-style-type: none"> · 온라인성 데이터 조작이 많다 · 타 시스템 참조성이 높다 · 데이터 량: 크다 · 사용자 수: 많다 	<ul style="list-style-type: none"> · Type D · 로직 분할 (Split-Logic C/S) 	<ul style="list-style-type: none"> · 원격 프리젠테이션 (Remote-Presentation C/S)

무별 특성과 제안모델 그리고 구현현황을 <표 4>에 나타냈다. B사의 신 시스템에 대한 자체 평가 결과, 구현된 아키텍처가 본 연구에서 제안한 모델보다 한 단계 높거나(인사/급여) 제안모델과 같았던 경우(회계관리), 높은 사용자 만족도를 보인 반면, '영업', '생산/자재', '원가'등 구현 아키텍처가 제시 모델과 현격히 달랐을 경우는 사용자 만족도가 매우 낮았다. 특히, 영업관리와 생산/자재관리는 가장 많은 사용자를 가지며 상호 참조성이 매우 빈번한 복잡한 응용시스템 입에도 불구하고 '원격 프리젠테이션'으로 구현되어 평시 서버의 CPU 가동율이 50-70%로써 전체시스템 장애의 원인이 되고 있다. 사업 확장에 따른 서버의 확장성 문제도 발생하게 된다. 원가관리는 조회작업이 많고 가끔 발생하는 배치성 마감작업시 CPU 부하가 큰 데, 역시 같은 아키텍처로 구축되어 서버가 큰 부하(실제로 CPU 가동율이 90%까지 올라감)를 받게 되어 반응시간에 대한 불만을 초래한다. 물론 이와 같은 결과는 사용된 고객만족 측정도구가 검증되지 않았으며 다른 외부요인들도 존재가능했으므로 C/S 아키텍처의 선정이 고객만족의 차를 가져온 유일한 결정요인이었다고 볼 수는 없었다.

5.3 C 병원

C 병원은 기존 병원들과의 경쟁력 제고를 위해 업무전산화에 큰 비중을 두고 91년 5월부터 병원업무전산화 수립 후, 국내외의 병원정보시스템 사례와 첨단 정보기술들을 조사, 분석, 평가한 후 개발에 착수하여 94년 9월 진료시스템 시험 가동, 94년 11월 정식으로 개원하였다. C병원은 전산화 대상 업무범위를 진료, 원무, 진료지원, 일반관리업무로 정의하고 이중 그래픽 환경이 바람직하고, 각 단말기에서 환자에 관한 다양한 정보를 한 번에 조회, 처리할 필요가 있는 진료시스템만 클라이언트-서버로 구현했다. 구현 아키텍처는 클라이언트에서 모든 작업을 수행하고 데이터관리만 원격 DBMS가 맡는 '원격 데이터 C/S' 모델이었다.

<표 5>에서 보듯이, C병원의 진료시스템은 전체 트랜잭션의 40%(약 30만건/일)를 차지할 만큼 상대적으로 업무량 및 데이터량이 많다. 또한 전체 사용자의 60%가 이 시스템을 사용하며 그들의 대부분이 의사와 간호사로서 전통적으로 전산화가 어려운 대상

자이므로 사용자 입지를 충분히 고려하여 사용하기 쉬운 환경을 제공할 수 있어야 한다. 응용시스템은 그다지 복잡하지는 않고, 데이터 액세스가 매우 빈번하며, 데이터 량도 비교적 크므로 업무의 정보처리적 특징은 'Type B'에 가깝다고 판단되어 '원격 프리젠테이션' 아키텍처가 적합하다고 판정된다. 그러나 빈번한 데이터 전송 시 네트워크 부하가 커지는 '원격 데이터' 아키텍처로 이루어진 C병원의 진료시스템은 개원 초기에는 데이터 량이 많지 않으므로 문제가 없었으나 시간이 흐르면서 사용자가 증가하고 데이터가 방대해 짐에 따라 시스템 성능(응답시간)이 서서히 떨어지는 것을 감지할 수 있었다. 또한 막강한 서버 기기(DEC7630)를 보유하고도 아키텍처의 영향으로 단말기 모드에 30%도 못 미치는 CPU 사용율을 보여 비효율적인 운용을 하고 있다.

<표 5> C병원 진료업무처리 특성과 현황
<Table 5> The characteristics of C hospital's information processing environment

업무	업무특성	정보처리모델	구현 아키텍처
진료	<ul style="list-style-type: none"> 트랜잭션: 30만건/일 사용자 많다 데이터 조작이 많고 량도 크다 	<ul style="list-style-type: none"> Type B 원격 프리젠테이션 (Remote-Presentation C/S) 	<ul style="list-style-type: none"> 원격 데이터 (Remote-Data C/S)

6. 결 론

본 연구에서는 클라이언트-서버 환경의 성공적인 구현을 위하여 C/S 아키텍처의 특성과 기업의 정보처리특성을 반영한 상황적 C/S 구현모델을 탐색적 차원에서 제시 및 검증하였다. 제 5장의 사례연구를 통하여 본 논문의 상황모델이 제시한 대로 C/S 환경이 구현된 경우 대체적으로 시스템 성과나 사용자 만족도가 높았던 반면, 그렇지 못했던 경우는 시스템 성능의 저하, 사용자 불만족 및 과대투자에 따른 시스템 사용을 부진 등을 확인할 수 있었다. 이러한 결과는 물론 작고 임의적인 표본에서 나온 것이므로 통계적으로 유의한 수준의 검증이 이루어졌다고 볼 수는 없으나, 탐색적 연구수준의 검증은 제시해 주었다고 평가할 수 있겠다.

본 연구의 한계와 추후과제로는 막대한 투자와 노

력을 들인 기존 시스템을 클라이언트-서버 시스템으로 연계할 수 있는 방안에 대한 검토가 없었다는 점, C/S 아키텍처의 평가기준과 기업정보처리 특성의 평가기준간의 논리적 연결이 약하다는 점과 업무특성의 분류가 비교적 단순하여(2차원) 향후 보다 구체적인 세밀한 분석과 분류가 필요하다는 점 등을 들 수 있다. 또한, 시스템개발이 대부분 4GL이나 특정 클라이언트-서버 개발 도구에 의존하고 있어 지원하는 도구에 따라 아키텍처가 결정되는 경우가 많으므로 향후에는 업무특성에 따라 다양한 아키텍처를 자유로이 구현할 수 있는 융통성 있는 도구의 개발이 필요하다고 본다. 마지막으로 보다 많은 기업들의 C/S 구현 환경에 대한 보다 정밀한 이론체계 및 실증연구의 수행을 통하여 제시된 방법론의 타당성을 지속적으로 검증/보완하는 작업이 남아 있다.

참 고 문 헌

- [1] 강영무, "클라이언트/서버:90년대를 주도하는 정보 기술 환경", 한국경영정보학회 논문집, 1994. 11. 10, pp499-512.
- [2] 광주은행, "다운사이징 기법을 통한 은행의 계정 업무무 구축사례", ORACLE, 1994.
- [3] 김종권, "Client/Server 시스템 구축 방법", 1993. 8.
- [4] 박태훈, "클라이언트/서버 아키텍처 환경하의 다운사이징 시스템 구축을 위한 성공요인", 학위논문, 1994. 6.
- [5] 서길수, 김영주, "Downsizing", 한국경영정보학회 논문집, 1993. 6. 18, pp451-498.
- [6] 심정보, "C/S환경에서 ORACLE RDBMS의 역할", ORACLE, 1994.
- [7] 이진영, "전사적 정보구조(Enterprise Information Architecture)", 오픈 컴퓨팅, 1994. 5, pp182-185.
- [8] 이길형, "은행정보시스템의 분산처리 방안에 관한 연구", 학위논문, 1993.
- [9] 전기홍, "병원정보시스템과 클라이언트-서버 환경", 1993. 6.
- [10] 최도준, "삼성의료원 종합정보시스템 구축사례", 삼성그룹 오라클사용자회, SDS, 1993. 12.
- [11] IBM, "클라이언트/서버 컴퓨팅 디자인 가이드", 1993. 3.
- [12] SDS, "Unix 환경의 정보시스템구축", 정보기술의동향, 1993. 12.
- [13] Atre, S., "Twelve Steps to Successful Client/Server", DBMS, May 1994, pp70-76.
- [14] Berson, A., Client/Server Architecture, McGRAW_HILL, 1994.
- [15] Dewire, D. T, CLIENT/SERVER COMPUTING, McGRAW HILL, 1993.
- [16] DBMS, "Downsizing, Upsizing, Rightsizing", 8 Case studies from DBMS, December 1992.
- [17] DBMS, "Out of the Lab, Into production", 5 Case studies from DBMS, December 1993.
- [18] Edelstein, H., "Unraveling Client/Server Architectures", DBMS, May 1994, pp34-42.
- [19] Gagliardi, G., CLIENT/SERVER COMPUTING :Killing the Mainframe Dinosaur and Slashing Runaway MIS Costs, PTR Prentice Hall, 1994.
- [20] Harmon, P.(editor), "Client Server Systems: Overview", CASE Strategies, April 1993.
- [21] Joe, S., Guide to Client/Server Databases, ZD Press, 1993.
- [22] Renaud, P. E, INTRODUCTION TO CLIENT/SERVER SYSTEMS, WILEY, 1993.
- [23] Semich, J. W, "Where Do C/S Apps Go Wrong?", DATAMATION, January 1994, pp30-36.
- [24] Smith, P., Client/Server Computing, SAM's Publishing, 1992.
- [25] Vaskevitch, D., CLIENT/SERVER STRATEGIES :A Survival Guide For Corporate Reengineers, IDG BOOKS, 1993.
- [26] Watt, P., "Flexible Computing at Sprint", DBMS, January 1994, pp79-82.