

# 유비쿼터스 환경을 위한 코바 기반 미들웨어 비교 연구

이웅주<sup>a</sup>, 윤희용<sup>a</sup>, 조위덕<sup>b</sup>

<sup>a</sup>성균관대학교 대학원 컴퓨터공학과  
경기도 수원시 장안구 천천동 300번지  
Fax: +82-031-299-6689, E-mail: oistrach@skku.edu, youn@icc.skku.ac.kr

<sup>b</sup>유비쿼터스 컴퓨팅 프론티어 사업단  
경기도 수원시 팔달구 원천동 산5번지 아주대학교  
Fax: +82-031-219-1718, E-mail: chowd@aju.ac.kr

## 요약

최근 유비쿼터스 컴퓨팅 분야가 이슈로 떠오르면서 유비쿼터스 환경하에 각종 디바이스들이 이기종 하드웨어 및 소프트웨어 자원에 종속되지 않고 손쉽게, 자유롭게 외부환경과 이음새 없는 서비스를 지원하기 위한 미들웨어 기술의 중요성이 강조되고 있다. 유비쿼터스 지향 미들웨어는 향후 개발될 새로운 기술이나 디바이스들을 용이하게 유비쿼터스 컴퓨터에 수용할 수 있고, 사용자나 시장의 요구에 따라 손쉽게 확장과 축소할 수 있는 적응성, 융통성, 재사용성을 만족시킬 수 있어야 한다. 본 논문에서는 분산 미들웨어인 COM/DCOM과 CORBA를 비교하여 유비쿼터스 환경을 위한 미들웨어로서 CORBA가 좀더 적합함을 설명하고, 유비쿼터스 환경에서 원활한 서비스를 지원해 주기 위해서는 우수한 성능의 미들웨어가 요구되기 때문에 DCOM보다 범용성이 뛰어난 코바 기반 미들웨어를 성능평가 한다.

## 키워드

유비쿼터스, 미들웨어, CORBA, ORB

## 1. 서론

컴퓨터는 지난 수십 년 간 사람들의 일터와 주거 공간, 공공장소에 깊숙이 침투해 우리의 생활패턴과 문화를 완전히 바꾸어 놓았다. 최근에는 초소형 SoC(System on Chip), MEMS(Micro Electro Mechanical System) 및 센서, 이동 컴퓨팅 기기, 무선 네트워크, 인터넷, 분산 컴퓨팅, 내장형 시스템, 멀티미디어 및 HCI(Human Computer Interaction) 기술의 비약적인 발전으로 유비쿼터스 컴퓨팅 환경을 탄생 시켰다[1][2][3].

유비쿼터스 환경은 컴퓨터가 도처에 편재하여 센싱과 트래킹을 통해 장소나 시간에 따라 변경된 정보를 서비스 받을 수 있는 환경을 말한다. 분산화된 많은 양의 컴퓨팅 노드 존재와 동적인 Ad-hoc 네트워크를 이용한 유비쿼터스 컴퓨팅은 장소, 이동여부, 휴대용 컴퓨터 기기, 통신기기 및 통신 대역폭의 다양성과 같은 물리적 요소로부터 독립적인 모델이 되어야 하며 분산화된 환경에서 변화에 따른 이기종간에 객체들이 원활하게 상호작용을 하기 위해서는 환경 및 대역폭에 독립적이고 시스템을 동적으로 재구성할 수 있는 유비쿼터스 환경에 적합한 객체들간에 통신이 가능하도록 소프트웨어 버스를 제공하는 분산객체용 미들웨어 기술이 필요하다[3][4][5].

본 논문에서는 유비쿼터스 컴퓨팅과 미들웨어에 대한 개념을 소개하고 유비쿼터스 환경을 위한 분산 미들웨어인 Microsoft의 COM/DCOM과 OMG의 CORBA를 서로 비교, 분석한다. 또한 유비쿼터스 환경에서 원활한 서비스를 지원해 주기 위해서는 우수한 성능의 미들웨어가 요구되기 때문에 범용성 측면에서 좀더 뛰어난 CORBA를 성능평가 함으로써 유비쿼터스 환경 지향적인 미들웨어로서의 적합성에 대해 분석해 본다.

## 2. 유비쿼터스 컴퓨팅

유비쿼터스 컴퓨팅(Ubiquitous Computing)은 다양한 컴퓨터가 사용자를 중심으로 동작하는 것이라고 말할 수 있는데, 유비쿼터스는 라틴어로 “언제 어디서나”, “동시에 존재한다” 라는 의미를 갖는다. 이를 위하여 상황인지(context awareness)와 통신 기능을 가진 칩, 센서 및 컴퓨터가 주위의 모든 사물 또는 공간에 보이지 않게 내장되어 사람, 사물 및 기계 등 무엇이든 서로 접속하여 실시간

으로 어떠한 정보든지 주고받을 수 있으며, 이러한 방법을 통하여 컴퓨팅(Computing), 커뮤니케이션 (Communication), 접속(Connection), 콘텐츠 (Contents), 조용함(Calm) 등 5C의 5Any화(Anytime, Anywhere, Anynetwork, Anydevice, Anyservice)를 제공 함으로써 인간에게 최적의 서비스를 제공할 수 있는 차세대 기술이며 다음의 서비스를 제공한다.

- 상황과 환경에 적응 가능한 네트워크 환경 (어디서나 네트워크에 연결)
- 편의성, 다양성이 높은 단말 환경(무엇이든 단말로 사용)
- 서비스 및 애플리케이션의 자유로운 이용 환경(자유로운 콘텐츠 사용)
- 다수 사용자가 동시 이용 가능한 초고속망 환경(편안한 네트워크)
- 안전한 정보이용 환경(안심하고 사용할 수 있는 네트워크)

어디에 있어도 네트워크, 단말 및 콘텐츠를 자유롭게 의식하지 않고 스트레스 없이 안심하고 이용할 수 있다.

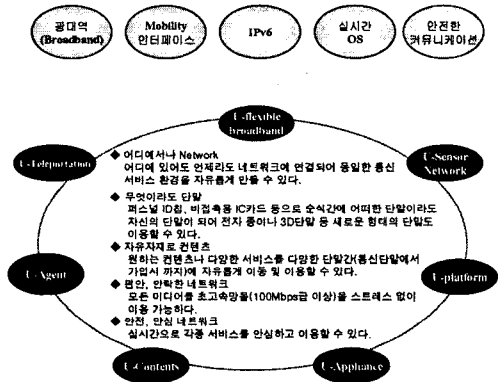


그림 1-유비쿼터스 컴퓨팅 개념

유비쿼터스 컴퓨팅을 구현하기 위해서는 내장형 프로세서 기술, 다양한 유무선 통신프로토콜을 지원하는 네트워크 기술, 능동적 센서와 이를 연결해주는 센서 네트워크 기술, 센서 미들웨어 기술, IPv6 및 All-IP 등과 같은 객체 인지기술, 전역 및 지역 위치감지 기술 등이 필요하다. 그리고 내장형 실시간 운영체제 및 미들웨어, 에이전트, 보안, 인간과 컴퓨터간의 대화를 위한 인터페이스 기술 등이 요구된다.

특히, 유비쿼터스 컴퓨팅은 인간, 사물, 환경을 유기적으로 통합을 목적으로 하는 방대한 컴퓨팅 패러다임이므로 매우 동적이고, 지극히 분산되어 있으며, 방대한 시스템 규모를 가지게 된다. 유비쿼터스 컴퓨팅의 구성 요소인 각종 디바이스들은 일련의 개인 지원기기(예, PDA, 스마트 폰, 웨어러블 컴퓨터 등)뿐만 아니라 반도체 칩이 들어간 냉장고, 세탁기, 가스레인지, 스테레오, 전기발삼 등의 홈

네트워크에서 다루는 정보가전기기도 포함된다. 유비쿼터스 지향의 세계에서는 이같이 다양한 장치들이 서로 네트워크로 연결 되게 된다. 또한 그 장치들이 서로 정보를 교환하고 또한 공유해야 한다. 그리고 이러한 네트워크 환경은 단순한 유무선 네트워크뿐만 아닌 광역 모바일 네트워크, 센서 네트워크, Ad-hoc 네트워크를 지원해야만 한다. 즉, 다양한 네트워크 환경에서 다양한 장치들 간의 투명성을 제공해야만 하는 것이다.

이처럼, 광범위한 기술 스펙트럼과 각종 다른 하드웨어 디바이스 네트워크, 운영체제, 입출력 장치 등이 혼재된 유비쿼터스 환경이 제대로 동작하기 위하여 이기종 하드웨어 및 소프트웨어 자원에 종속되지 않고 손쉽게, 자유롭게 외부환경과 이음새 없는(seamless) 서비스를 지원할 수 있는 에이전트 플랫폼 기술과 이기종간의 상호 호환성, 사용자 투명성을 제공하기 위한 미들웨어 및 응용 수준에 이르는 모든 영역에서, 재구성, 재사용, 적응성을 효과적으로 지원할 수 있는 에이전트 통합 소프트웨어 구조 개발의 중요성이 부각되고 있다.

### 3. 미들웨어

미들웨어는 클라이언트가 서버로 서비스를 요청하는 일종의 가교 역할을 한다. 유비쿼터스 환경에서 다양한 장치들이 서로를 인지하고 통신하며 적합한 서비스를 제공하기 위해서는 물리적 공간, 시간, 네트워크 환경, 각종 장치들을 인지하여 서비스를 제공하는 상황인식 기술이 반드시 필요하다. 그리고 매우 다양한 장치들이 통신하며 운영되어 지기 때문에 효율적인 협업 시스템 구현이 필수적이다. 유비쿼터스 지향 미들웨어는 유비쿼터스 환경이 다양한 방법으로 통신하며 의미와 정보를 유출하기 때문에 각 컴퓨팅 환경에 가장 적절한 프레임워크 방식의 보안 방식이 지원 되어야 한다. 더욱이, 향후 개발될 새로운 기술이나 디바이스들을 용이하게 유비쿼터스 컴퓨터에 수용할 수 있고, 사용자나 시장의 요구에 따라 손쉽게 확장과 축소할 수 있는 적응성, 융통성, 재사용성을 가진 소프트웨어 구조 개발이 중요하다.

이러한 에이전트 및 미들웨어 기술들은 유선 네트워크상의 정적인 분산 시스템 환경에서는 성공적으로 사용되어 왔지만 유비쿼터스 환경으로 이를 바로 적용하기에는 다음과 같은 이유로 인하여 많은 난제들을 가지고 있다.

- 분산 트랜잭션, 오브젝트 요청이나, RPC (Remote Procedure Call) 등은 컴포넌트들 간에 항상 가용한 높은 대역폭의 네트워크 연결을 가정하지만, 유비쿼터스 시스템들은 단절(unreachability)과 낮은 네트워크 대역폭을 표준으로 고려한다.
- CORBA와 같은 객체 지향 미들웨어 시스템

은 대부분 동기화된 P2P (Peer-to-Peer) 통신을 지원하지만, 유비쿼터스 시스템은 저전력 제약으로 인하여 서비스 지역 이탈과 같은 부득이한 단절 등으로 인하여 클라이언트와 서버간의 연결이 동일한 시간에 이루어지지 않는다.

전통적인 분산 시스템들은 안정적이며, 높은 대역폭, 고정된 위치, 잘 알려진 서비스와 같은 정적인 실행환경에서 작동되는 것을 가정하지만, 유비쿼터스 시스템들은 계속 되는 디바이스 위치 변경, 이동 중 새로운 서비스 발견과 같은 동적인 환경에서 실행된다.

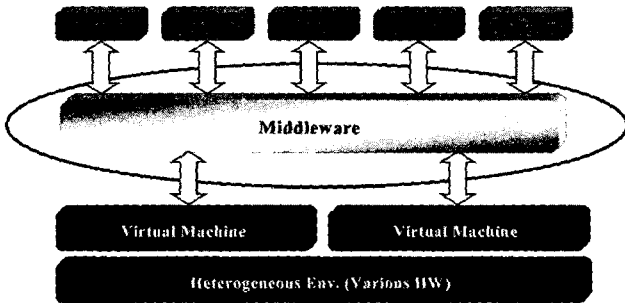


그림 2-미들웨어 개념도

따라서 전통적인 상용 미들웨어인 CORBA나 DCOM등을 이동 내장형 시스템에 적용하기에는 위에서 언급한 바와 같은 문제점들과 함께 너무 무겁다(heavy)는 제약을 가지고 있다. 이처럼 현재 유비쿼터스 디바이스들의 동작에 필요한 에이전트 플랫폼 및 소프트웨어는 다양한 환경에 유연하게 적응하지 못하고, 산재한 유비쿼터스 컴퓨팅 인프라를 충분히 지원하지 못하고 있다. 이러한 점 때문에 각종 유비쿼터스 디바이스의 다양한 하부 플랫폼과 상관없이 동작할 수 있는 유비쿼터스 지원 에이전트 개발 및 실시간 분산처리 기술의 개발이 중요하며 이에 대한 장기적이고 체계적인 연구 개발이 요구된다.

### 3.1 ORB(Object Request Broker)

ORB는 분산 객체 또는 컴포넌트 모델에서 가장 중요한 공통된 객체간 통신 서비스로서 분산된 객체들 간의 연결성을 만들어 주는 핵심적인 객체 미들웨어 기술이다. 아래 그림 3과 같이 응용 또는 특정 서비스 객체가 구현객체(Object Implementation)의 오퍼레이션을 호출할 때 Client의 호출을 받아 자신이 가지고 있는 구현객체들의 정보를 참조하여 Client가 원하는 구현객체를 찾고 Client

에서 보내준 서비스 요청 정보를 해당 구현객체에게 전달하여 해당 객체로 하여금 그 작업을 수행하며 작업이 완료되면 ORB는 역으로 그 결과 값을 받아 Client에게 전달하는 역할을 수행하게 하는 역할을 한다.

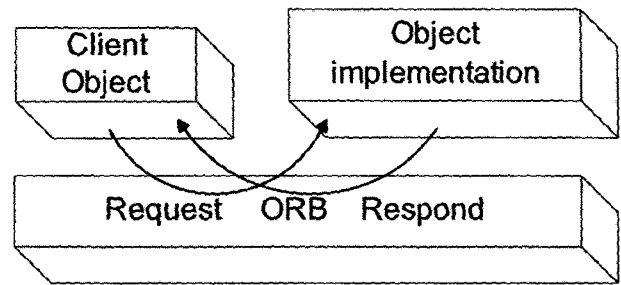


그림 3-ORB 기능

ORB는 객체간의 통신을 전달하는 소프트웨어 버스(BUS)로 개념을 정의 할 수 있다. 또한 ORB는 컴퓨터 하드웨어 안에서 다양한 하드웨어 컴포넌트(메모리, 프로세스 등)가 서로 통신 할 수 있도록 해주는 하드웨어 버스와 같이 분산 객체 또는 컴포넌트에서 서비스의 위치와 관계없이 제공할 서비스에 대해 클라이언트가 요구하는 시점부터 그 요구가 완료될 때까지 연결시켜주는 브로커 역할을 하는 프로그램이다. 이기종의 분산된 환경하에서 응용 프로그램을 서로 통합할 수 있는 표준 기술로는 Microsoft의 COM/DCOM과 OMG의 CORBA 가 있다.

### 3.2 COM/DCOM 정의

Microsoft의 OLE (Object Linking and Embedding) 기술은 1990년 발표됐으며 이후로 OLE2가 발표되면서 윈도우 애플리케이션 간의 일반적인 상호 운영이 가능하여 상이한 유형의 문서를 공유할 수 있게 됐다. 이를 위해 OLE2에는 OLE2 인식 애플리케이션에서 화면 사용을 조정할 수 있는 단순한 단일 시스템 통신 모델이 포함되어 있다. OLE가 OLE2로 발전하면서 COM(Common /Component Object Model)이라는 새로운 모델이 탄생했다. OLE 자동화를 통해 애플리케이션을 운영하는 C++, Visual Basic 등과 같은 언어의 컴포넌트를 OLE 컨트롤이라고 한다. COM은 컴포넌트 기반 소프트웨어 애플리케이션 구축을 위한 인프라스트럭처로 구현됐다. DCOM(Distributed COM)은 COM과 동일한 인터페이스를 지원하는 라이브러리지만 네트워크 시스템 이나 단일 플랫폼의 컴포넌트 사이에서의 운영도 가능하다. DCOM은 COM 객체가 근거리 통신망, 원거리 통신망 또는 인터넷에서 동작할 수 있도록 COM을 확장한 것이다. 네트워크 상에 있는 COM 객체를 활성화시키고 그 메소드를 호출할 수 있는 기능을 제공

한다. DCOM은 COM과 함께 또는 COM 위에 만들어진 규격이고 서비스이다.

### 3.3 CORBA 정의

OMA(Object Management Architecture)의 객체 지향 기술을 기반으로 하여 이종의 분산된 환경하에서 응용 프로그램들을 서로 통합할 수 있는 표준기술으로써 이것은 응용 프로그램간의 결합뿐만 아니라 객체의 생성, 소멸에서 부터 저장, 트랜잭션 기능에 이르기까지 분산 객체 환경에서 필요한 모든 서비스를 총칭하는 것이다. OMA가 제공하는 모든 서비스는 객체 개념을 기반으로 있으며 OMA는 이종의 분산 환경에서 통신을 담당하는 CORBA와 객체를 조작하는 데 필요한 각종 기본 기능들을 정의하고 있는 COSS(Common Object Service Specification), 그리고 추가적으로 제공되는 지원 서비스가 존재하고 있다. 이에 더하여 위와 같은 이들 기능들을 이용하여 사용자들이 작성한 각종 응용 객체들이 존재한다. OMA 구조는 그림 4 와 같다.

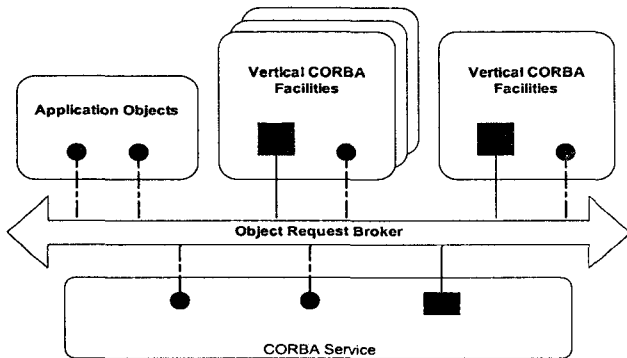


그림 4 - Object Management Architecture

### 3.4 ORB 기술 비교

DCOM은 COM의 구현을 하나의 컴퓨터 영역을 넘어서 네트워크 상에서 사용할 수 있도록 확장했다. 현재 COM의 차세대 버전인 COM+를 개발 중 이고, 개발의 주 대상이 데스크탑용 컴포넌트라면 무료이다. 또한 DCOM은 사용자들에게 친근한 환경을 제공해 주고, Visual Basic, Delphi 등의 visual 개발도구에 익숙할 경우, 전체 분산 시스템이 마이크로 소프트 플랫폼일 경우 유리하다.

CORBA는 이기종 환경에서 분산 객체 시스템을 위한 표준이고, 프로그래밍 언어와 플랫폼에 독립적인 표준, OMG의 표준이다. 개발의 주 대상이 Enterprise 레벨로, 그 시스템이 mission-critical하고, 고가용성을 필요할 때 유리하다(전체 분산 시스템이 이기종에서 수행될 때 유리하다). 또한 인터넷을 기반으로 하는 시스템을 만들 때 유리하다.

아래 표 1, 2에서 DCOM과 CORBA의 장단점을 정리해 보고 표 3에서 CORBA와 DCOM을 비교해 보았다.

표 1 - DCOM의 장점과 단점

장점	단점
통합 개발환경 제공	제한된 운영체제 지원
프록시와 스텝코드 생성 및 COM객체 추가용이	인터페이스만 상속
다양한 컴포넌트	정적 및 동적 호출에 상이한 인터페이스 필요
정적 또는 동적 인터페이스 지원	멀티스레드 수행시 COM라이브러리 초기화
위치 투명성	결합허용 및 부하조정 기능 미약
멀티스레드 서버 지원	학습의 어려움

표 2 - CORBA의 장점과 단점

장점	단점
다양한 플랫폼 지원	독립적인 CORBA 확장 지원
강력한 프로그램 언어 지원	과다한 교육 시간 필요
위치 투명성	비표준 명령 지원
정적 또는 동적 메소드 호출 지원	개발 툴 지원 부족
구현 상속성	X
강력한 보안 기능	X
멀티스레드 서버 지원	X
멀티스레딩을 위한 초기화 없음	X
강력한 결합-허용 및 부하-조정 스펙 지원	X

표 3 - CORBA와 DCOM의 비교

항목	CORBA	DCOM
주요 관련 벤더	OMG 컨소시엄 Sun, HP, IBM, DEC	마이크로소프트 Software AG, DEC
플랫폼	UNIX, MVS, NT, Win95, Mac	NT 4.0, Win95, Solaris 2.5, Mac (예정)
상호 운용 가능	ORB에 따라 다르다	마이크로소프트

C 결합(Binding)	지원(ORB 종속)	지원
C++ 결합(Binding)	지원(ORB 종속)	지원
Smalltalk 결합(Binding)	지원(ORB 종속)	지원하지 않는다
Java 결합(Binding)	지원 예정	Java용 윈도우 가상 머신 (VM) 지원
COBOL 결합(Binding)	지원 예정	지원하지 않음
비용(NT 전용)	높다	낮다
비용(이기종 환경)	높다	측정 불가
복잡성	높다	매우 높다
언어 독립성	높다	낮다
정적/동적	프로그램 호출이 용이	프로그래밍이 어렵다
툴지원	미약하다	미약하지만 지원이 강화되고 있다
분산 기능	Name, Directory	개발자가 구현
데이터베이스 통합	OMG 질의 서비스	OLE 자동화
메시지 통합	INOA, IBM, DEC	Microsoft Message Queue Server (MSMQ)
인터넷 통합	IIOP, VisiBroker for C++	Jakarta (COM 과 Java), Active/X to DCOM from broker
트랜잭션 지원	OMG OTS with IONA, Encina ++	MTS
기업환경 지원	실무 경험 보유	성장하고 있다

소스 미들웨어이다.

- TAO □ Schmidt 교수와 그가 속한 그룹에서 개발한 real-time을 지원하는 오픈 소스 미들웨어이다. 또한 CORBA의 모든 서비스를 지원한다.
- ORBacus - OOC에서 처음 개발했고 지금은 IONA에서 개발되고 있는 오픈 소스 미들웨어이다.
- VisiBroker □ Borland 사에서 개발한 상업적 목적으로 쓰이고 있는 미들웨어이다.

성능평가 요소는 10000개의 메시지를 띄웠을 때 method invocation time의 분포를 측정하는 Invocation(그림 5), 클라이언트에서 서버로, 또는 서버에서 클라이언트로 들어오는 데이터 길이에 따른 method invocation time을 측정하는 In Sequence, Out Sequence(그림 6,7), 클라이언트와 서버에 등록된 오브젝트 수를 증가시키면서 오브젝트 invocation time을 측정하는 Obeject(그림 8), 현재 수행되는 스레드를 증가시키면서 오브젝트 invocation time을 측정하는 Multithreading(그림 9)으로 나뉘볼 수 있다. 대부분의 항목에서 omniORB가 가장 좋은 성능을 나타냈고, VisiBroker가 비교적 낮은 성능을 보임을 알 수 있다.

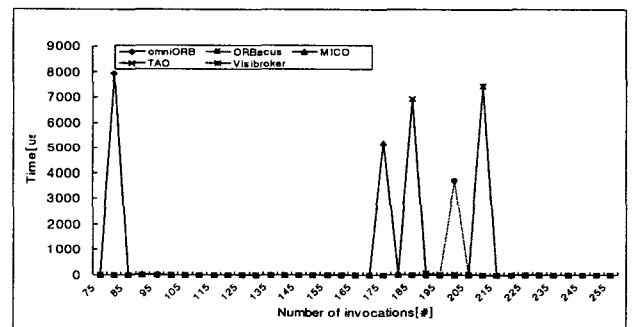


그림 5 - Invocation : method invocation times 분포

#### 4. CORBA 미들웨어 성능평가

본 논문에서는 현재 여러 분야에서 폭넓게 쓰이고 있는 코바 기반 미들웨어인 omniORB 4.0, Obacus 4.0, Mico 2.3, TAO 1.4, VisiBroker 4.5의 성능을 평가, 비교해 보았다. 각각의 미들웨어들의 특징을 간단히 살펴보면 다음과 같다.

- omniORB □ Olivetti & Oracle Research Lab에서 처음 개발되었고 후에 AT & T Laboratories Cambridge에서 개발하고 있는 가볍고, 높은 성능을 나타내는 오픈 소스 미들웨어이다.
- MICO □ 멀티쓰레드를 지원하지 않는 오픈

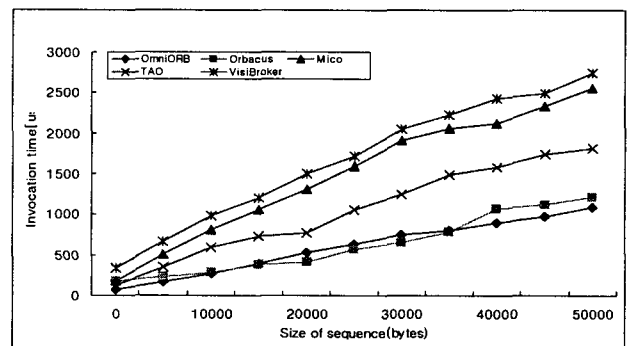


그림 6 - In Sequence: 클라이언트에서 서버로 들어오는 데이터 길이에 따른 method invocation time

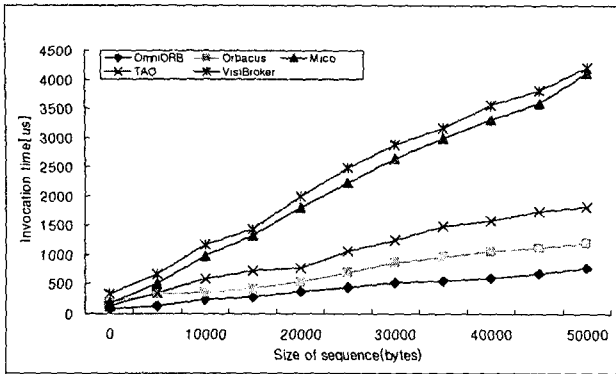


그림 7 - Out Sequence: 서버에서 클라이언트로 들어오는 데이터 길이에 따른 method invocation time

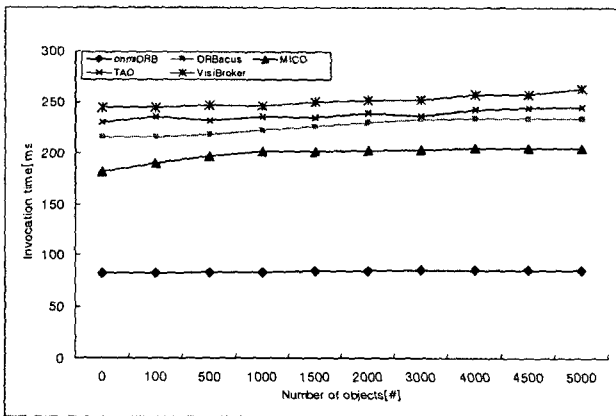


그림 8 - Object: 클라이언트와 서버에 등록된 오브젝트 수 증가에 따른 오브젝트 invocation time

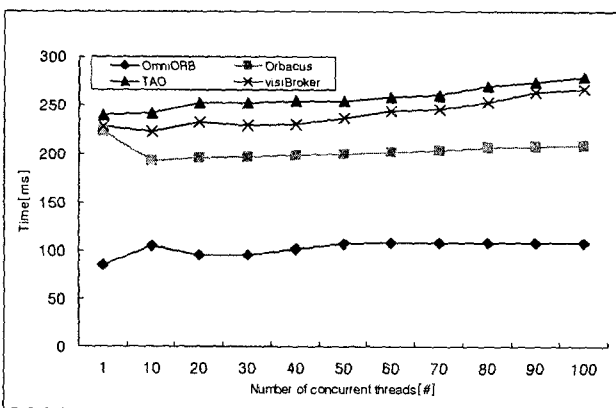


그림 9 - Multithreading: 현재 수행되는 스레드수의 증가에 따른 오브젝트 invocation time

## 5. 결론 및 향후 연구 방향

본 논문에서는 유비쿼터스 컴퓨팅과 미들웨어에 대한 개념을 정립하고 유비쿼터스 환경에 적합한

미들웨어 연구를 위해 이기종의 분산된 환경하에서 응용 프로그램을 서로 통합할 수 있는 Microsoft의 COM/DCOM과 OMG의 CORBA를 서로 비교, 분석해 보았다. 유비쿼터스 환경에서 요구하는 ORB 기술은 이기종간의 미들웨어가 모두 상호 작용을 원활하게 할 수 있도록 하기 위해 서비스를 수행할 서버를 유일하게 식별 할 수 있는 naming service를 가지고 있으며, 극도의 분산된 환경에서도 상호 작용이 원활하면서 무선 이동 환경 및 자원 제약에도 효율적으로 동작 될 수 있는 기술이 필요하다. 따라서 유비쿼터스 환경을 위한 미들웨어에 적합한 기술로는 DCOM보다 안정적이며 이기종 환경과 다양한 언어를 좀 더 효율적으로 제공하는 CORBA가 적합할 것이다. 또한 CORBA기반 미들웨어인 ORB들을 선정하여 성능 평가 해 본 결과 omniORB가 여러 항목에서 대체적으로 우세한 성능을 나타냄을 알 수 있다. 향후 유비쿼터스 환경에서 이러한 미들웨어를 사용하기 위해서는 사용될 환경에 따라 성능을 더 개선 시키거나, 필요한 기능을 추가 또는 불필요한 기능을 축소, 제거 하는 연구가 필요 하겠다.

## 참고 문헌

- [1] M. Satyanarayanan, "Pervasive Computing Vision and Challenges", IEEE Personal Comm, vol. 6, no. 8, Aug. 2001, pp.10-17.
- [2] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "EasyLiving: Technologies for intelligent environments," presented at the Conf. Handheld and Ubiquitous Computing, Bristol, UK, Sept. 25-27, 2000.
- [3] S. S. Intille, Designing a home of the future, in IEEE Pervasive Computing, vol. 1, pp. 76-82, Apr./June 2002.
- [4] N. Cook, S. Shrivastava, and S. Wheeler. Distributed Object Middleware to Support Dependable Information Sharing between Organisations. Technical Report 753, Dept. of Computing Science, Univ. Newcastle, Apr. 2002.
- [5] Object Management Group, The Common Object Request Broker: Core Specification, version 3.0.2, OMG formal/02-12-02, Dec 2002
- [6] Distributed Systems Research Group, CORBA Comparison Project, Project Extension Final Report, Charles University, Prague, 1999
- [7] Callison, H., R., Butler, D., G., Real-time CORBA Trade Study, Boeing, 2000
- [8] Plasil, F., Tuma, P., Buble, A., Charles University Response to the Benchmark RFI, OMG bench/98-10-04, Oct 1998
- [9] University of Helsinki, Response to ORBCS PTF Benchmark RFI, OMG bench/98-10-03, 1998

- [10] Object Management Group, White Paper on Benchmarking, OMG bench/99-12-01, 1999
- [11] Tuma, P., Buble, A., Open CORBA Benchmarking, Proceedings of the 2001 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '01), published by SCS, Orlando, FL, USA, Jul 2001
- [12] Gokhale, A., S.; Schmidt, D., C., Measuring and Optimizing CORBA Latency and Scalability Over High-speed Networks, IEEE Transactions on Computers, Vol. 47, No. 4, Apr 1998
- [13] W. Zhao, L. E. Moser and P. M. Melliar-Smith. Design and implementation of a pluggable fault tolerant CORBA infrastructure. In Proceedings of the International Parallel and Distributed Processing Symposium, Fort Lauderdale, FL, April 2002.
- [14] Juric, M., B.; Rozman, I., Java 2 RMI and IDL Comparison, Java Report, No. 2, SIGS Publications, Feb 2000
- [15] Juric, M., B.; Rozman, I., RMI, RMI-IIOP and IDL Performance Comparison, Java Report, Vol. 6, No. 4, SIGS/101 Publications, Apr 2001
- [16] Juric, M., B., Rozman, I., Stevens, A., P., Hericko, M., Nash, S., Java 2 Distributed Object Models Performance Analysis, Comparison and Optimization, Proceedings of ICPDAS '00, Iwate, Japan, IEEE Computer Society Press, pg. 239-246, Jul 2000