

ALAN 상에서 적응성 있는 웹 캐싱 서비스

홍성준*

*여주대학 정보통신과

sjhong@mail.yeojoo.ac.kr

The Adaptive web caching service on application level active network

Sung-June Hong*

*Department of Information & Communication , Yeojoo Institute of Technology

sjhong@mail.yeojoo.ac.kr

요 약

본 논문은 ALAN(Application Level Active Network)상에서 GME(Generic Modeling Environment)라고 불리는 컴포지션 메커니즘을 이용하는 AWCS(Adaptive Web Caching Service)에 관하여 언급하였다. ALAN 상의 웹 캐싱의 요구사항은 자치 구성성(self-configuration)과 자치 적응성(self-adaptation)이다. 이러한 요구사항을 지원하기 위해서 본 논문에서는 ALAN 상에서 IN(Intelligent Inetwork)에서 유래된 SCE(Service Creation Environment)와 유사한 서비스 컴포지션 메커니즘인 GME 를 이용하여 AWCS 의 설계 및 구현에 관하여 언급하였다.

1. 서론

웹 캐싱은 웹의 발전과 함께 성장했고 웹이 점점 커지고 복잡해 짐에 따라 웹 캐싱의 복잡성도 같이 커져왔다. 기존의 웹 캐싱은 관리자가 수동 구성(manual configuration)을 해야하는 단점이 있다. 그래서 자치구성(self-configuration)을 지원하는 AWC Adaptive Web Caching)[1]가 제안되었다. AWC 은 오버래핑 멀티캐스트 그룹(overlapping multicast group)을 구성하는 자치 구성 웹 캐싱 시스템을 지원하므로 웹 캐싱에 자치 구성성 지원을 하였다. 그러나 AWC 와 같은 낮은 수준의 튜닝과 같은 자치 구성성이 아닌 더 상위 수준의 자치 구성성을 위한 자치 적응성(self-adaptation)이 요구된다[2].

그러므로 본 논문에서는 ALAN(Application Level Active Network)[3]상에서 GME(Generic Modeling Enviroment)[4]이라고 불리는 서비스 컴포지션 메커니즘을 이용하여 자치 적응성(self-adaptation)위한 AWCS(Adaptive Web Caching Service)를 제안하였다.

AWCS 는 사용자가 기존 웹 캐싱과 기능과 더불어 적응성과 결합 허용성(fault-tolerance)을 요구할 경우 ALAN 의 자치 구성성의 특징과 GME 의 서비스 컴포지션 기능 그리고 PCM proxylet 의 자치 적응성 기능의 도움으로 쉽게 적응성과 결합 허용성을 추가 또는

삭제가 가능한 서비스를 의미한다. 이러한 자치 적응성 기술은 Ubiquitous Computing 에서도 필수적으로 요구하는 기술이기도 하며, 향후 AWCS 는 Ubiquitous Computing 에서도 사용될 수 있을 것이다.

컴포지션을 지원하는 도구는 MDA(OMG Model Driven Architecture)기반의 도구와 GME 도구가 있다. MDA 도구는 주로 비즈니스 응용에 사용되고 GME 도구는 주로 시스템 응용에 사용된다. 현재 GME 는 IN(Intelligent Network)에서 유래된 SCE(Service Creation Environment)를 직접적으로 지원하지 않지만, SCE 와 같은 컴포넌트 재구성 지원 메커니즘 즉 자치 적응성 메카니즘 개발에 관한 연구가 진행 중에 있다[6]. SCE 는 사용자가 서비스 로직만 제공하면 언제든지 응용 서비스를 제공할 수 있다. 서비스 로직은 복잡한 서비스 개발을 단순화 하면서 재사용될 수 있다.

한편, ALAN 은 AN(Active Network)의 특징과 Proxylet 이라 불리는 이동 에이전트의 특성을 가지면서 망내에서 자치조직성을 지원한다. ALAN 은 자치 구성성과 빠른 서비스 배치를 가능하게 하는 에이전트 기반의 AN 이다. 기존의 AN 연구가 네트워크 라우터 수준에서 진행되었다면, ALAN 은 응용 수준에서 진행되고 있다. ALAN 과 같은 액티브 서비스(active service) 접근 방법은 망의 낮은 수준에 영향을 미치지 않고 더욱 빠르게 서비스의 배치가 이루어 질 수

있다는 장점이 있다. 최근에는 ALAN 에 자치 적응성 (self-adaptation) 메커니즘을 적용하여 Ubiquitous Computing 에 적용하고자 하는 연구가 진행되고 있다 [5].

2 장에서는 관련연구로 ALAN 과 GME 에 관하여 언급하였다. 3 장에서는 ALAN 상에서 AWCS 의 설계에 관하여 언급하였고 4 장에서는 AWCS 의 구현에 관하여 언급하였다. 5 장에서 실행 결과 및 특성 비교를 하였고 6 장에서 결론을 맺었다

2. 관련 연구

2.1 ALAN[3]

AN 연구는 주로 네트워크 라우터 수준에서 개발 진행되고 왔다. 그러나 영국 UCL(University College London)에서 제안된 ALAN(Application Level Active Network)은 네트워크 수준에서 문제점을 보완하기 위해서 응용 수준의 액티브네트워크를 제안하였다. ALAN 은 웹 과 같은 클라이언트와 서버로 구성되어 있고 서버와 클라이언트사이의 통신은 EEP(Execution Environment for Proxylet)와 프로그래머블하고 다운로드 가능한 Proxylet 에 의해서 성능 향상된다. 현재의 EEP 는 웹 캐싱 하부구조를 가지고 있고, EEP 는 브라우저와 웹 서버 사이에서 가장 가까운 최적의 거리에 있는 캐시 데이터를 찾아서 연결시키는 기능을 한다. ALAN 관련 프로젝트로는 ALPINE(Application Level Programmable Inter-Network Environment), CASPIAN(Constructing A dynamic Service environment using Policies Integrated with Active Network) , CCML(Component Compatibility Markup Language) 등이 있으며 이 대부분의 ALAN 관련 프로젝트는 ALAN 의 out of band control 메커니즘을 이용하여 IN 과 같은 구조로 만들려는 접근 방법을 취하고 있다.

2.2 GME[4]

GME 는 도메인 기반 모델링과 프로그램 종합 환경 (Program Synthesis Environment)를 지원하는 구성 가능한 도구를 의미한다. GME 는 MIC(Model Integrated Computing)를 기반으로 하는 기술로 MIC 는 도메인 모델을 실제 구현 코드로 자동으로 변환시켜준다.

3. AWCS 의 설계

3.1 AWCS 의 구조

그림 1 은 ALAN 상에서 AWCS 의 구조를 보이고 있다. 제약조건(constraint)에는 사용자의 PC, 이동 터미널 등의 다양한 디바이스로부터 얻을 수 있는 자원 제약조건과 서비스 제공자가 다양한 서비스 제공을 위해서 요구하는 컴포지션 제약조건이 있다. 이런 자원 제약조건과 컴포지션 제약조건은 GME 에 의해서 모델링 된다. PCM(Proxylet Configuration Manager) proxylet 은 GME 에 의해서 제공된 정보에 따라 적합한 Pxoylet 들을 선택하고 조합 및 관리한다. PCM 에 의해서 조합된 AWCS 에는 많은 proxylet 들로 구

성된다. AWCS 의 Supervisor Proxylet 는 Intelligence management 모듈과 FT management 모듈을 포함한다. Worker Proxylet 는 ScanNetTraffic 모듈과 FT_detection 모듈이 포함되어 있다. Intelligence management 모듈은 가장 가까운 Worker Proxylet 로 연결 시키기 위한 우회(redirection) 기능과 부하분담(load balancing)기능이 있다. FT management 모듈은 결함 허용(fault-tolerant)기능을 관리하는 모듈이다. ScanNetTraffic 모듈은 각 네트워크의 대역폭 등의 네트워크 정보를 조사하는 기능을 하는 모듈이다. FT_detection 모듈은 중복 캐싱 서버에 결함이 발생했는지를 감지하는 기능을 하는 모듈이다

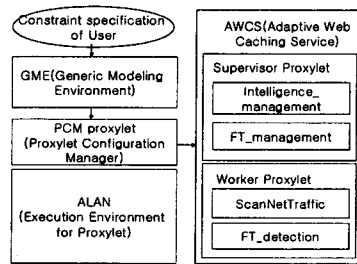


그림 1. ALAN 상의 AWCS 의 구조

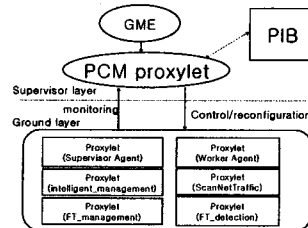


그림 2 Ground 계층의 Supervisor 계층의 통합

그림 2 는 GME 와 PCM proxylet 을 포함하는 Supervisor 계층과 각 컴포넌트 역할을 하는 다양한 proxylet 으로 구성된 Groud 계층의 통합된 구조를 보이고 있다. PCM 은 GME 와 PIB(Proxy Information Base)정보를 토대로 적당한 Proxylet 선택하고 조합하는 역할을 한다. 즉 자치 적응성(self-adaptation) 지원 기능을 한다. PCM 의 역할은 모니터링 (monitoring), 의사결정(decision-making), 재구성성 (reconfiguration)이다.

3.2 AWCS 의 동작

그림 3 은 ALAN 상에서 멀티캐스트 기반의 AWCS 의 동작을 보이고 있다. 본 논문의 웹 캐싱 서비스는 중복 캐싱 개념[7]을 도입하였다. 중복 캐싱은 Supervisor Proxylet 와 Worker Proxylet 가 있는 서버에 중복 된 데이터를 가지고 있도록 하는 것이다. ALAN 상의 AWCS 은 EEP(Execution Environment for Proxylet)와 Proxylet 으로 구성된다. EEP 는 AN 의 실

행 환경을 의미한다. 그리고 Proxylet 은 ALAN 상의 이동 에이전트를 의미한다. Supervisor Proxylet 는 일종의 SLP(Service Location Protocol)기능을 수행하는 것으로 어떤 중복 캐시가 가장 빠르게 응답하는지를 찾는다. Worker Proxylet 는 중복 서버로 파견되고 중복서버의 대역폭 등과 같은 네트워크 정보를 얻는다. 또한 클라이언트와 Supervisor Proxylet 사이의 프락시(proxy) 역할을 한다

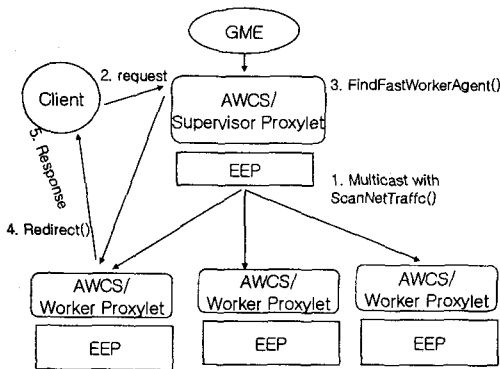


그림 3. ALAN 상의 AWCS 의 동작

1. Supervisor Proxylet 는 멀티캐스트 메시지를 보내고 중복 서버에 파견된 Worker Proxylet 로 부터 대역폭과 같은 네트워크 상태의 정보를 모을 수 있다.
2. 클라이언트가 데이터를 요청할 때 Supervisor Proxylet 는 Worker Proxylet 로부터 얻은 네트워크 상태 정보로부터 가장 빠른 Worker Proxylet 가 있는 중복 서버를 발견할 수 있다.
3. AWCS 는 Supervisor Proxylet 로 연결시키는 대신에 가장 빠른 Worker Proxylet 의 중복서버로 우회(redirect)시킨다.
4. 클라이언트는 가장 빠른 Worker Proxylet 의 중복 서버로부터 데이터를 얻는다.

4. AWCS 의 구현

AWCS 는 Window 2000 Server 상에서 자바 프로그램 언어를 이용하였고 UTS 와 BT 에 의해서 개발된 ALAN 공개 소프트웨어와 ISIS 와 Vanderbilt 대학에 의해서 개발된 GME 2000 도구를 이용하여 구현되었다.

4.1 GME 도구를 사용하는 AWCS

본 논문은 컴포지션 메카니즘으로 GME 를 사용한다. GME 는 응용 수준에서 meta modeling 이라는 모델 언어를 이용하여 설계하고 실제 코드는 자동 생성하는 환경을 제공한다. 그림 4 의 컴포지션 다이어그램은 각 컴포넌트간의 관계를 보이고 있는데 크게 클라이언트, Supervisor Proxylet, WorkerProxylet 로 구성

되어 있다.

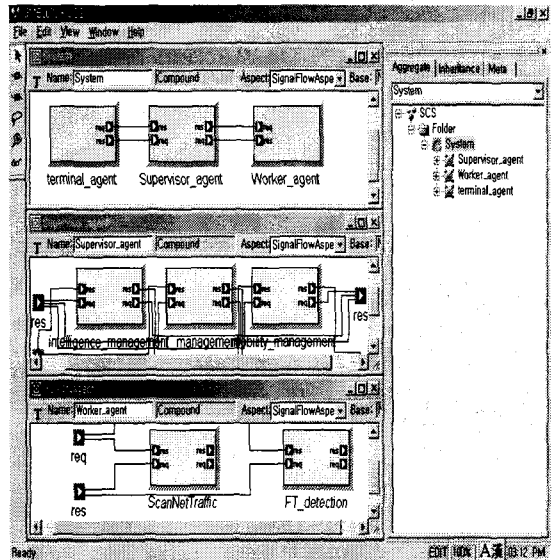


그림 4. GME 도구에 의한 컴포지션

4.2 AWCS Proxylet 구현 코드

```

public AWCS implements Proxylet {
    public AWCS() {}
    public Object Run( String param){
        String url = new String("PS");
        AWCS AWCSRef = (AWCS) Naming.lookup(url);
        String load_balancing = AWCSRef.supervisor();
        return "OK"
    }
}
    
```

그림 5. AWCS Proxylet 구현 코드

그림 5 는 AWCS Proxylet 구현 코드를 보이고 있다. AWCS Proxylet 은 Supervisor Proxylet 를 호출한다. 본문에서 PCM(Proxylet Configuration Manager)의 구현은 개발 중에 있으므로 GME 에 의해서 설계된 내용이 PCM 을 거치지 않고 바로 Proxylet 구성 및 구현으로 되어 있다.

5. AWCS 실행 결과 및 특성 비교

5.1 지능성 실험

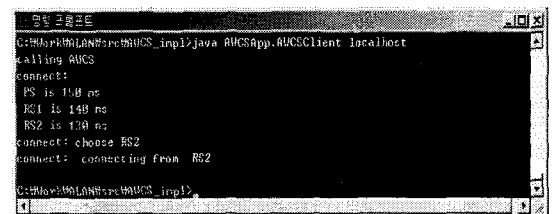


그림 6. 가장 빠른 중복 서버를 찾는 실험

그림 6 은 Supervisor Proxylet 가 있는 서버를 PS(Primary Server)라고 명명한다. 그리고 여러 개의 Worker Proxylet 가 있는 중복서버를 RS(Replication Server)라고 명명한다. 그리고 그 중에서 가장 빠른 Worker Proxylet 가 있는 RS 를 선택하는 실험을 보이고 있다. 수행 결과는 각 RS 의 대역폭을 보이고 그 중에서 가장 빠른 RS 를 선택하는 과정을 보이고 있다.

5.2 결합허용성 실험

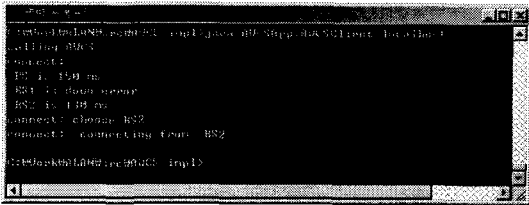


그림 7. 결합 허용성 실험

그림 7 은 AWCS 의 결합 허용의 실험 결과를 보이고 있다. Worker Proxylet 가 있는 RS 중에 하나가 시스템 결합이 발생했을 때 그 RS 는 서비스를 제공하지 못하기 때문에 그 RS 를 제외한 다른 RS 중에서 가장 빠른 RS 를 찾는 과정을 보이고 있다.

5.3 특성 비교 평가

표 1 은 AWC, ALAN 의 기존 웹 캐싱 그리고 본 논문의 AWCS 의 주된 특성 비교를 보이고 있다. AWC 는 웹 캐싱에 자치 구성성(self-configuration)을 제공한다. 본 논문의 AWCS 는 기존의 ALAN 의 웹 캐싱 보다 결합허용성, 자치 적응성(self-adaptability)을 더 지원하는 특징을 가진다.

표 1. 주된 특성의 비교

	AWC	ALAN 상의 기존 웹 캐싱	ALAN 상의 AWCS
제어신호 트래픽의 감소	X	O	O
데이터 트래픽 감소	X	O	O
자치구성성	O	O	O
지능성	X	O	O
결합허용성	X	X	O
자치 적응성	X	X	O

6. 결론 및 향후 연구

본 논문은 GME 라 불리는 응용 수준 컴포지션 메

카니즘을 이용하여 ALAN 상에서 적응성 있는 웹 캐싱을 위한 컴포지션 메커니즘을 제시하였다. 향후 연구로는 보다 지능화된 자치 적응성(self-adaptation)의 연구가 필요하다.

참고 문헌

- [1] Floyd, S., "Adaptive Web Caching," Boulder Cache Workshop '97, (June 9-10, 1997) in AWC, <http://www-nrg.ee.fbl.gov/floyd/web.html>.
- [2] Oliver Aubert and Antone Beugard,"Toward a fine-grained adaptivity in web caches", in proceedings of the 4th international web caching workshop, April 1999, <http://www.ircache.net/workshop/papers/aubert-0.ps.gz>.
- [3] I.W.Marshall, et, al, "Application-level Programmable Network Environment," BT Technology Journal, Vol. 17, No2 April 1999.
- [4] Ledeczki A., Maroti M., Bakay A., Karsai G., Garrett J., Thomason IV C., Nordstrom G., Sprinkle J., Volgyesi P, "The Generic Modeling Environment," Workshop on Intelligent Signal Processing, accepted, Budapest, Hungary, May 17, 2001.
- [5] K. T. Krishnakumar, M. Sloman, "Constraint-based Network Adaptation for Ubiquitous Applications," *Proceedings of the 6th International EDOC Conference*, Sep. 2002, pp 258-271, Laussane, Switzerland.
- [6] Karsai, G., Á. Lédeczi, J. Sztipánovits, G. Péceli, Gy. Simon, T. Kovács házy, "An Approach to SelfAdaptive Software based on Supervisory Control," Second International Workshop on Self Adaptive Software IWSAS 2001, May 17-19, Balatonfüred, Hungary, 2001.
- [7] M.Baentsch,A.Lauer, L.Baum,G.Molter,, S.Rothkugel, P.Sturm,"Enhanceing the Web's Infrastructure - From Caching to Replication," IEEE Internet Computing, Vol.1,No. 2, March-April 1997.