ACSA : An Adaptive Content System Architecture[†]

Jae-Dong Lee* and Jin-Sung Kim**

Abstract Adapting content services to the needs of the users requires content adaptation for context and sensibility. Adapting to context may necessitate distinguishing between video delivery to a PC versus a smart phone, while adapting to sensibility may necessitate between video delivery outside, in a quiet environment, or at home, or offering different movie choices on a sunny summer day, a cold winter day or a holiday season. One key area to address in providing adaptive content services is the design of the delivery system architecture of the adaptive content server. This paper describes several alternatives for this architecture, and outlines some additional concerns that must be considered when the chosen architecture is implemented.

Key Words : Content Adaptation, Adaptive Content Services, Context, Sensibility

1. Introduction

The paradigm of content service is rapidly changing with the advent of the ubiquitous computing era. Ubiquitous computing leads to the increased consumption of digital contents. Ubiquitous computing environment also enables people to consume plenty of digital contents with diverse devices such as PC, PDA, cell phone, television at anytime and anyplace[1][2]. With the computing paradigm shifts, users want adaptive content service that supplies the appropriate contents for their context such as device, network, mood and situation.

Therefore, this paper aims to provide various multimedia content to users by adapting to the delivery technology (e.g., PC, tablet PC, smart phone, or heavily loaded vs. lightly loaded network)

as well as adapting to the user's mood or situation (e.g., depressed, hot day, rainy weather).

Content adaptation is the action of transforming contents to adapt to device capabilities[3]. Many studies have performed to adapt contents to users' context[4]. Most of the studies are about reduce contents to adapt web contents to mobile devices. Content adaptation is usually related to mobile devices that require special handling because of their limited computational power, small screen size, and limited network condition[5][6][7].

What is required is a content adaptation approach that could automatically generate any content version from one single original version such that the content is adapted to the device and the user preferences[7][8]. Content adaptation is illustrated in Figures 1 and 2. Figure 1 illustrates content adaptation to the user's context[9]. In this figure, multimedia content(e.g., a movie) is delivered full size to a PC but scaled down to a smaller size for viewing on a tablet PC or smart phone. Not shown,

[†] 이 논문은 2010년도 단국대학교 대학연구비 지원으로 연구되었음

* 단국대학교 소프트웨어학과, 주/교신저자

** 단국대학교 소프트웨어학과

but implied, is the lower resolution that may be required to deliver the movie over the more limited bandwidth of the tablet PC and smart phone. Throughout Asia and more recently Europe and the US, multimedia content is quickly becoming available on tablets and smart phones[10][11].

<Figure 1> Content Adaptation for Context (Different Device Types)

Figure 2 illustrates content adaptation to the user's sensibility and context. In this figure, multimedia content (a movie) is delivered to a smart pad, so the video resolution is reduced to

<Figure 2> Content Adaptation for Sensibility and Context (Different Locations, Different Device Types)

128K due to lower bandwidth. The user is initially outside, so both audio and text are delivered. As the user enters the library, audio delivery ceases and the user sees only the video and text. Eventually the user arrives at home, where the video resolution increases to take advantage of the user's higher-bandwidth home wireless network and both audio and text are again available.

Adaptive content system requires generally four key capabilities[12]:

- ① Context and sensibility detection
- ② User profile generation (e.g., device type, network bandwidth, personal sensibility)
- ③ Adaptive content generation
- ④ Adaptive content delivery

In this paper, we primarily address the last issue – adaptive content delivery. Section 2 describes the overall delivery system architecture of the adaptive content server, and the tradeoff between several different implementation alternatives. Section 3 outlines some additional concerns that must be considered when the chosen architecture is implemented. Section 4 presents our conclusions.

2. An Adaptive Content System Architecture

The overall architecture of the adaptive content server is shown in Figure 3. At the bottom is the Content Adaptation Subsystem (CAS), which adapts the content as necessary to meet various contexts and sensibilities. At the top is the Content Delivery Subsystem (CDS), which delivers the adapted content to individual users. Both subsystems can be scaled as necessary to provide the required service.

2.1 Subsystem Architecture of an Adaptive content system

2.1.1 Content Adaptation Subsystem(CAS)

The Content Adaptation Subsystem, which adapts the content as necessary to meet various contexts and sensibilities, is illustrated in Figure 4. Since multimedia re-encoding is a computationally challenging task, the key feature of this subsystem is the Compute Intensive Nodes capable of

quickly. Digital Video Recorders (DVRs) typically use custom hardware to encode broadcast movies into MPEG4 files at speeds sufficient to allow the user to watch, pause, and jump through the encoded movie essentially as it is received. The Adaptive Content Server's CAS must provide similar functionality.

<Figure 3> Architecture of an Adaptive Content Server

re-encoding the content quickly enough to meet the needs of the user.

Consider the task of re-encoding the movie from a DVD to view on a tablet PC, smart phone, or iPod. The DVD will contain a set of .VOB files comprising 3-5 GB of high-quality video, audio in several languages, and subtitles in a few languages. To view this content on handheld device, the content might need to be re-encoded into a single MPEG4 file at a video size of 320x240, bit-rate of 700 kbps, and audio sample rate of 44.1 KHz – a process that might take an hour or two on a standard laptop.

However, a more powerful computer or dedicated hardware can re-encode the movie content more

The CAS must also provide sufficient storage for huge multimedia content it must provide. At 4GB per movie, a home library of only 200 DVDs would comprise 0.8TB of data if stored directly as .VOB files, so the multimedia content of the CAS must scale into the 100's or 1000's of tera-bytes. Further, the data should be replicated to allow multiple compute nodes to more efficiently access the same content simultaneously (e.g., to re-encode the same movie for two different users with different requirements). While each compute node may not need access to every single piece of content, some duplication would be necessary. Perhaps as an initial design each piece of content should be replicated and available from at least two

<Figure 4> Content Adaptation Subsystem of an Adaptive Content Server

<Figure 5> Content Delivery Subsystem of an Adaptive Content Server

separate disks.

2.1.2 Content Delivery Subsystem(CDS)

The Adaptive Content Server's Content Delivery Subsystem (CDS), which delivers the adapted content to individual users, is illustrated in Figure 5. Since simultaneous delivery of content to many users requires high I/O bandwidth, the key feature of this subsystem is the I/O Intensive Nodes capable of delivering the content quickly enough to multiple users.

Each node of the CDS must interact with the nodes of the CAS to get the adapted content, and then must deliver that adapted content to one or more users. At the same time, the node will also be delivering other adapted content to other users. A single CDS node should be able to serve the needs of 10's or possibly 100's of users, provided there are sufficient CAS nodes to provide adapted content. Of course, there must also be sufficient I/O bandwidth between the system and the users to

deliver that content.

Though this paper focuses primarily on adaptive content delivery, the CDS would also be responsible for the first two key capabilities identified at the end of Section 1: context and sensibility detection, and user profile generation and management.

2.2 Implementation Alternatives

The overall architecture of the adaptive content system shown in Figure 3 can be implemented on a variety of platforms. When choosing the implementation platform, it is important for the system designer to consider the number of users, available bandwidth to the users, type of content delivered, type of content adapted, and size of content adapted and delivered. In addition to the platform alternatives described below, it may be possible to further increase performance for a given platform using special-purpose hardware to perform or assist with the content adaptation.

2.2.1 Standard Workstation/PC

In low-volume situations with light content adaptation requirements, the Adaptive Content System could be implemented using a standard workstation or personal computer. Although a typical workstation has only 160-250 GB of hard disk space, external storage could easily be added in units of 500 GB or 1 TB for a modest increase in cost.

However, though this implementation platform will likely suffice to implement the CAS for a small number of users, it will likely be insufficient to implement the CDS for all but simple Adaptive Content Systems. For more demanding content adaptation such as video re-encoding, a separate processor will likely be necessary.

2.2.2 Replicated Systems with Load Balancing

In medium-volume situations with moderate content adaptation requirements, the Adaptive Content System could be implemented using (possibly dual-core or quad-core) replicated servers with load balancing. One set of systems implements the CAS, another set implements the CDS, and dedicated load-balancing hardware can distribute tasks between the nodes of the CAS and the nodes of the CDS. Again, such load balancers are commercially available and come in rack-mounted configurations. The overall implementation resembles the architectural diagram of Figure 3.

A replicated system with load balancing configured in this manner will use a similar structure to a moderate-volume web server or database server and should suffice to implement the CAS and CDS for a hundreds of users. Further, it can be scaled up in response to system needs as resources permit. However, it may still lack sufficient processing power for the most demanding content adaptation, and does not consider the unique needs of the CAS vs. the CDS.

2.2.3 Dedicated CAS/CDS Blade Systems with Load Balancing

In high-volume situations with moderate-to- high content adaptation requirements, the Adaptive Content System could be implemented using (possibly dual-core or quad-core) dedicated CAS/CDS blade servers with load balancing. As in the replicated system alternative, one set of servers implements the CAS, another set implements the CDS, and dedicated load-balancing hardware can distribute tasks between the nodes of the CAS and the nodes of the CDS. Again, the overall implementation resembles the architectural diagram of Figure 3.

However, this implementation alternative differs by using customizable 'blade' servers (also available in rack-mounted configurations) configured differently for the CAS and CDS. For the I/O-intensive CAS, 'thin' blades are used with single-core processors and higher-capacity I/O. For the compute-intensive CDS, 'thick' blades are used with dual-core or quad-core processors and additional memory and lower-capacity I/O. These customizable blade servers are available from vendors such as Sun Microsystems.

A replicated system with load balancing configured in this manner will use a similar structure to a high-volume application system and should suffice to implement the CAS and CDS for a hundreds of users. It can be scaled up in response to system needs as resources permit, and it is a more cost-effective solution since it considers the unique needs of the CAS vs. the CDS. However, though enough thick blades can be added to perform the necessary content adaptation, using expensive general-purpose hardware may not always be the best implementation alternative for the CDS subsystem.

2.2.4 CAS MIMD Multimedia Servers/ CDS Blade Server

In very high-volume situations with moderate-to-high content adaptation requirements, the Adaptive Content System could be implemented using CAS MIMD multimedia servers with CDS blade servers. As in the dedicated CAS/CDS blade server alternative, thick blades are used for the compute-intensive CDS. However, the blade servers in the CAS are replaced with MIMD servers optimized for media streaming.

MIMD multimedia systems such as the nCUBE MediaCUBE-4 have been used to provide video-on-demand via cable to users for the past years. These systems are optimized to deliver high-bandwidth media content to hundreds of users simultaneously, in a variety of formats, with minimal replication of the content.

2.2.5 CDS with Hardware Accelerators for Content Adaptation

In situations with high content adaptation requirements, the Adaptive Content System's CDS could be augmented with hardware accelerators for content adaptation. Just as home Digital Video Recorders(DVRs) typically use custom hardware to encode broadcast movies into MPEG4 files for storage, any of the CDS implementations described previously could be augmented with custom hardware for content adaptation.

For systems with a high content adaptation requirement, it may be necessary to design specialized hardware accelerators optimized for the specific content adaptation and use those accelerators to augment the commercial processors in the CDS.

3. Additional Architectural Concerns

The computer system designed for the Adaptive Content System must have three key characteristics. It must be expandable to handle more and more users, and more and more content. It must be adaptable to handle changes in technology. It must provide security against attacks by thieves. Each of these issues will be considered in turn.

3.1 Expandability

The computer system must be able to handle increases in traffic as the service gets more and more users and stores more and more content. It needs more and more disk storage, more and more computers, and more and more I/O ports to the outside world.

3.2 Adaptability

Computer technology changes rapidly. Computer equipment gets faster and faster, gets more and more capacity, and gets more and more economical. What you bought before is obsolete, and what you buy today will become obsolete. At any time the system will be a mixture of old technology and new technology.

Example 1: Telephones. Telephone switching technology has changed dramatically over the years. Operators placed calls for customers in the earliest years, then callers placed their own calls using rotary-dial phones sending pulses to the switchboard. Even later callers could place their calls using push-button phones sending tones. These changes were made incrementally — the telephone system still handles the oldest technology along with the newest technology.

Example 2: The MPP. Goodyear Aerospace Corp. delivered the MPP to the customer in 1983. At that

time it was a very fast SIMD machine with 16,384 PEs (Processing Elements), built using a custom VLSI chip with 8 PEs per chip. But now, about 30 years later, a single Pentium chip runs faster than the whole MPP, and it costs too much to re-design the VLSI chip, so the MPP has never been redesigned as a faster machine.

One lesson learned from the MPP is that systems should be designed using Commercial - Off-The-Shelf (COTS) parts. COTS parts are continually improved by their manufacturers as technology advances. In contrast, the performance of custom-designed parts won't be improved unless the customer is willing to pay to have them redesigned. Yet even then, the newer parts will soon become obsolete.

One lesson learned from the telephone system is that systems should use popular interfaces. New equipment should handle the same popular interfaces as the old equipment. The bandwidth of popular interfaces will be improved as technology advances occur. Today most Local Area Networks (LANs) use Ethernet implemented with COTS routers, and the networks are expandable and adaptable as the technology advances.

Systems should also use generic software. Changes in hardware should not require changes in software - don't use software that is specific to only one manufacturer. Use a generic operating system like Linux, and use a generic source language like C++ or Java. Changing the DBMS should not require a major change in the data-bases. Use generic library routines, not routines that are specific to a particular manufacturer.

3.3 Security

The Adaptive Content System will have a large store of multimedia content such as movies,

animations, music, etc. We can expect that there will be many who will try to bypass its security to steal its content, so the system must have excellent security to defend the service against these hackers.

The content output should be encrypted so that others can't steal it from cell phone signals and other wireless networks. The encryption should be strong enough to prevent wrongful decryption. Subtle changes to the content should be made so that you can identify a user who copies it to CDs or DVDs to sell to others.

For example, cell phones have very poor security. There are many cases of cell phones being cloned by persons listening to their signals. The users don't realize that their cell phones have been cloned until they get a bill showing a large number of calls made to foreign lands. Similarly, the Internet has very poor security.

The system must be able to authenticate the user - to distinguish between legitimate users devices and clones. One possibility is to give each user device a number of codes and ask the device to return a different particular code at each login. At worst, others trying to clone the device will only learn one of the codes.

4. Test Example

This section shows the test example based on the change of context.

As an example, both video and audio are delivered while the user is initially outside.(Figure 6. (a)) As the user enters the library, audio delivery ceases and the user sees only the video and text.(Figure 6. (b)) Eventually the user changes the tablet for smart phone, so the video resolution is reduced to 128K.(Figure 6. (c))

<Figure 6> (a) Tablet Screen
(original content : video on, sound on)

<Figure 6> (b) Tablet Screen
(outside→library : Sound off, Subtitle on)

<Figure 6> (c) Tablet Screen
(device change : screen size change)

5. Conclusions

This paper designed the overall adaptive content system architecture (ACSA) of the server. This system is composed of Content Adaptation Subsystem(CAS) and Content Delivery Subsystem

(CDS). Both subsystems could be scaled as necessary to provide the adaptive content service. We also described five alternatives for the architecture and outlined additional concerns that must be considered when the chosen architecture is implemented. The overall adaptive content system does not need any custom-designed parts - it's best to use the COTS parts available from a large number of manufacturers so the system is expandable and adaptable to changes in technology.

참 고 문 헌

- [1] Jean-Marie Hullot, "Ubiquitous Devices, Mobility and Context Awareness", LNCS Springer, Vol. 4866, pp.123-124, 2007.
- [2] 전우락, 소수환, 이재동, "콘텐츠 적응화 서비스를 위한 상황정보 수집 시스템의 설계 및 구현", 한국산업정보학회논문지, Vol.14, No.2, pp.1-7, June, 2009.
- [3] Zhijun Lei, Nicolas D. Georganas, "Context-based Media Adaptation in Pervasive Computing", In Proc. of Canadian Conference on Electrical and Computer Engineering, pp. 913-918, May, 2001.
- [4] Anna Carreras et al., "A Platform for Context-Aware and Digital Rights Management-Enabled Content Adaptation", IEEE Computer Society, pp.74-89, 2010.
- [5] Yamini Nimmagadda, Karthik Kumar and Yung-Hsiang Lu, "Adaptation of Multimedia Presentations for Different Display Sizes in the Presence of Preferences and Temporal Constraints", IEEE Transaction on Multimedia, Vol.12, No.7, pp.650-664, 2010.
- [6] Zhu Liu, David C. Gibbon, Harris Drucker, Andrea Basso, "Content Personalization and Adaptation for Three-Screen Services", pp. 635-644, CIVR'08, July 7-9, 2008.

- [7] 나문성, 김승훈, 이재동, "클라우드 환경에서 대규모 콘텐츠를 위한 효율적인 자원처리 기법", 한국산업정보학회논문지, Vol.15, No.4, December 2010.
- [8] Mohd Farhan Md Fudzee, Jemal Abawajy and Mustafa Mat Deris, "Multi-criteria Content Adaptation Service Selection Broker", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp.721-726, 2010
- [9] F. Pereira and I. Burneet, "Universal Multimedia Experience for Tomorrow", IEEE Signal Processing, Vol.20, No.2, pp.63-73, Mar. 2003.
- [10] Jiang He et al., "A Flexible Content Adaption System Using a Rule-Based Approach", IEEE Transactions on Knowledge and Data Engineering. Vol.19, No.1, 2007.
- [11] Rakesh Mohan et al., "Adapting Multimedia Internet Content for Universal Access", IEEE Transactions on Multimedia, Vol. 1, No.1, 1999.03.
- [12] Kyung-Sik Kim and Jae-Dong Lee, "Management Technology of Profiles for Providing Adapted Contents to an User in the Ubiquitous Environment", Journal Computing Practices and Letters(KIISE), Vol.13, No.6, pp.343-357, 2007.

이 재 동 (Jae-Dong Lee)

- 정회원
- 인하대학교 전자계산학과 학사
- Cleveland State University 석사
- Kent State University 박사
- 단국대학교 공과대학 소프트웨어학과 교수
- 단국대학교 미디어콘텐츠기술센터 센터장
- 단국대학교 모바일연구소 소장
- 관심분야 : Mobile Technologies/Applications, Ubiquitous Computing, Content Technologies

김 진 성 (Jin-Sung Kim)

- 학생회원
- 단국대학교 컴퓨터과학과 학사
- 단국대학교 컴퓨터과학전공 석사
- 단국대학교 소프트웨어학과 박사과정
- 관심분야 : Ubiquitous Computing, Mobile

논문접수일 : 2011년 04월 15일

1차수정완료일 : 2011년 05월 15일

게재확정일 : 2011년 05월 25일