

다중 Tier 온라인 서비스 상에서 대량 데이터 출력 성능 향상을 위한 비동기 통신 기법

김성룡*, 오재오**, 조운호**, 이상근**
 * 고려대학교 컴퓨터 정보통신 대학원
 ** 고려대학교 컴퓨터학과
 e-mail : stdragon@korea.ac.kr

Asynchronous Communication Technique for Heavy Data Output Performance Improvement on Multi Tier Online Service Environment

Sung-Lyong Kim*, Jae-Oh Oh**, Yoon-Ho Jo**, Sang-Keun Lee**

*Dept. of Software Engineering, Graduate school of Computer Information Communication, Korea University

**Division of Computer and Communication Engineering, Korea University

요 약

본 논문은 다중 Tier 상에서 온라인 서비스 대량 데이터 처리를 빠르고 정확하게 클라이언트에 전달하는 기법을 제안한다. Tier 가 많은 온라인 서비스상에서 대량의 데이터를 빠르게 처리하는 데에는 많은 어려움이 있다. Tier 간 지연 시간의 최소화, 네트워크 대역폭을 고려한 트랜잭션(Transaction)의 적절한 분할 통신, 이 기종간의 데이터 변환 시 처리속도 개선 등이 해결해야 할 주요한 요건이라고 할 수 있다. 하지만 이러한 문제들이 해결된다고 해서 관목할 만한 성능의 개선은 쉽게 나타나지 않는다, 그 이유는 바로 Partial Query 에 의한 데이터 통신이 꾸준히 반복 발생하기 때문이다. 온라인 서비스의 특성상 대량 데이터는 많은 사용자의 효율적인 트랜잭션 처리를 위하여 분할(Partial) 처리되어 통신하는 방식을 기준으로 사용하고 있다, 이러한 방식을 준수 하기 위해서는 데이터 사이즈에 비례하는 반복의 증가가 불가피하다. 그래서 반복 횟수를 줄이는데 포커스를 두고 온라인 서비스 대량 데이터 처리에 대한 성능 테스트를 진행한 결과 반복이 최소화 될수록 성능은 최대한으로 유지되며, 다른 어떤 기술적인 요소를 개선하는 것보다 큰 효과를 볼 수 있음을 알 수 있었다.

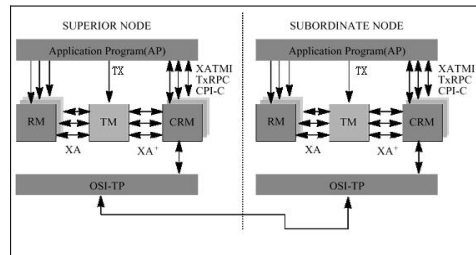
1. 서론

최근 차세대 Project 에서는 Web 서비스의 장점과 C/S 환경의 장점을 모두 가지는 다중 Tier 환경으로 구성된 개발 방식을 많이 사용하고 있다. 이러한 방식의 장점은 Web Browser 를 통한 다양한 UI(User Interface) 제공으로 사용자가 원하는 좀 더 많은 형태(Form)의 서비스 제공이 가능하고, X-internet 기능을 접목하여 Thin 클라이언트[6]와 Rich 클라이언트[7]의 장점을 모두 살릴 수 있다. 그리고 비즈니스 로직을 처리하는 Application Tier 에서는 Java 와 C 언어의 기능적인 이점을 모두 사용할 수 있어 다양한 비즈니스 환경에 적합한 기술구조로 사용되고 있다. 그러나 이러한 구조는 상대적으로 단일한 구조에 비해 물리적인 Tier 의 증가로 인한 성능이슈가 제기될 수 있고, 제기된 성능이슈는 온라인 서비스 대량 출력 요구사항이 있는 실시간 온라인 데이터 처리 업무에서는 반드시 해결해야 할 문제로 남게 된다. 본 논문에서는 이러한 문제를 해결하기 위한 다양한 통신 방식을 설계, 구현하였으며 비동기 통신 기법을 이용

한 성능 개선방법을 제시하고, 실험을 통해서 가장 효율적인 성능 개선방안을 제시한다.

2. 관련연구

이장에서는 본 논문의 논점을 설명하기 위한 배경 지식이 되는 분산 트랜잭션 처리를 규정한 X/OPEN DTP(Distributed Transaction Process), 즉 (그림 1)의 구조를 먼저 설명한다[1].



(그림 1) X/Open DTP 모델 구조도

X/OPEN DTP 모델은 <표 1>과 같이 다음 5 가지 기능적 구조로 나누어 진다.

<표 1> X/Open DTP 모델 기능구조

	기능	기능 설명
1	응용 프로그램 (AP)	-사용자가 작성한 응용 Program Transaction 경계 정의
2	Resource Manager (RM)	-공유자원들을 관리 -자원에 대한 접근 방식 제공 -DB, ISAM File Access System
3	Transaction Manager (TM)	-전체 System에 대한 Transaction 관리 -Transaction ID 정의, Commit /Rollback 제어 -Transaction 장애 시 복구에 대한 책임
4	Communication Resource Manager (CRM)	-같은 TM 영역내 혹은 서로 다른 TM 영역에 존재 -분산 응용 Program 들간의 통신 제어담당
5	Open System Interconnection -Transaction Processing (OSI -TP)	-서로 다른 TM 영역과의 통신 담당

이 모델에서 표준으로 규정한 XATMI(X. Application Transaction Manager Process) Interface 는 Application 과 TP Monitor 사이에 통신방식으로 사용되며 <표 2>와 같이 기능을 정의하여 사용한다 [1,2,3].

<표 2> X/Open XATMI Interface

관련 기능	이름	기능
버퍼 관리	tpalloc ()	데이터를 송수신할 버퍼 할당
	tprealloc ()	버퍼 크기 변경
	tpfree ()	할당된 버퍼 해제
	tpatypes ()	버퍼의 크기와 형식에 대한 정보 제공
서비스 요구 및 응답 관련	tpcall ()	서비스를 요청하고 응답이 올 때까지 대기
	tpacall ()	서비스를 요청하고 다른 처리를 하다가 tpgetrply()호출 시 처리 결과를 수신
	tpcancel ()	서비스 요청 대한 응답 취소
	tpgetrply ()	tpacall() 호출에 대한 응답 수신
대화형 Mode	tpconnect ()	대화형 Mode에서 메시지 송/수신을 위한 연결
	tpdiscon ()	대화형 Mode 서비스와 연결을 비정상적으로 종료
	tprecv ()	대화형 Mode에서 Message 수신
	tpsend ()	대화형 Mode에서 Message 송신
서비스 종료	tpreturn ()	서비스 요청에 대한 응답을 클라이언트에게 보내고 서비스 루틴 종료

이 X/Open XATMI Interface 기반 하에 국내외에서 시장점유율이 가장 높은 제품별로 대용량 처리 시 성능 개선을 위한 다양한 사례를 분석하였다.

2.1. On-Demand 처리방식

tpcall interface[2,3]을 이용하여 작업을 요청하는 화면과 대량의 데이터를 처리하는 실시간 Batch 응용 프로그램(Application Program) 기능을 조합한 방식으로 작업을 요청한 사용자는 요청 작업 상태를 확인하며 다른 업무를 동시에 수행 할 수 있다. 작업 종료 후 X-internet Grid 형식(Format)에 맞춘 처리 결과 파일을 FTP(File Transfer Process) 기능을 이용하여 전송하는 방식으로 빠른 결과 전송이 가능하 다.

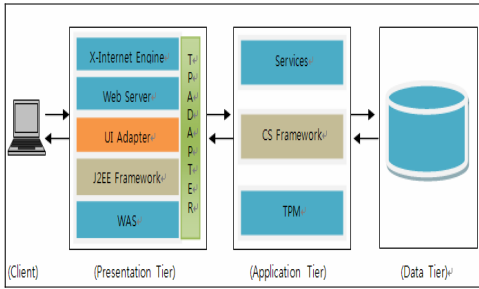
2.2. 동기 통신방식을 이용한 최적량 전송방식

클라이언트와 서버간에 tpcall 과 tpgetrply 를 주고 받는 동기 통신방식으로 Data 량을 조절해 가며 분할 통신하는 방식. 데이터 건수를 조절해가며 최소동기 시간이 걸리는 건수와 데이터 건수에 비례적으로 가감되는 반복횟수 중 두 측정값의 최적 수치를 찾아 통신하는 방식.

3. 다중 Tier 온라인 서비스 대량출력을 위한 구조 (Architecture)

다중 Tier 온라인 서비스 구조 (그림 2)는 X-Internet Grid 환경의 클라이언트, Web 서버와 WAS(Web Application Server)를 통한 Presentation 데이터를 구성하는 UI Tier, 비즈니스 로직을 구현하는 Application Tier, 데이터 I/O 를 관리하는 데이터 Tier 로 구성된다. X-internet Grid 형식의 클라이언트에서는 데이터 조회 범위를 정의하고 반복 분할 전송되는 데이터를 FirstRow 방식[4]으로 화면에 출력한다. FirstRow 방식을 사용할 경우 Presentation Tier 상에 X-internet 엔진 버퍼에 전달된 데이터를 좀 더 빨리 사용자에게 출력하여 보여 줄 수 있다. WAS 기반의 J2EE(Java 2 Enterprise Edition) Framework 은 Application Tier 의 서비스에서 전송한 문자열(String) 데이터를 X-Internet 에서 처리 할 수 있는 데이터 형식으로 변경하는 역할을 수행한다. TPM(Transaction Processing Monitor)과 CS Framework 은 클라이언트의 요청을 받아 Data Tier 에서 생성된 Partial 처리 결과를 Fetch 와 동시에 비동기 통신 전송을 수행한다. Fetch 후 바로 전송하는 이유는 대용량 Query 를 가능한 빨리 클라이언트로 전달하기 위해서 고안되었다. Application Tier 와 데이터 Tier 는 DBMS(Data Base Management System)에서 제공하는 Native Driver 를 이용하여 최적의 성능이 나도록 구성 하였다.

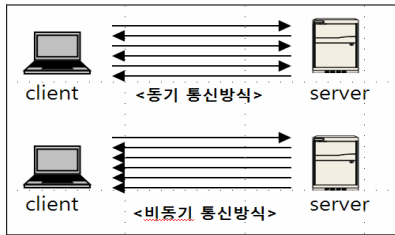
☞ Fetch : DBMS 에서 Cursor 를 선언하여 멀티 Row 를 받아서 Array 변수에 입력 처리하는 기능



(그림 2) 다중 Tier 온라인 서비스 구조

3.1 개선된 통신방식

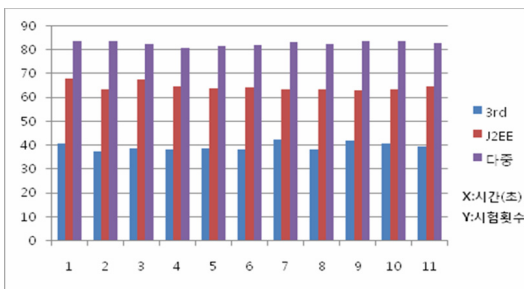
다중 Tier 온라인 서비스 대량출력에 적합하게 동기 통신방식을 개선하여 비동기 통신 방식으로 바꾸어 반복을 최소화 하였다. 즉, 한번의 요청으로 모든 데이터를 분할 전송하는 방식으로 개선 하였다.



(그림 3) 동기 비동기 통신방식 비교

(그림 3)의 전송방식은 tpcall 을 통하여 데이터 Tier 에서 Fetch 후 지정된 클라이언트로 비요청 데이터를 바로 전송하는 방식[5]이다. 결과적으로 통신 방식을 비교해 보면 비동기 통신방식의 경우 요청을 1 회만 하므로 동기 통신방식에 비해 요청에 소요되는 시간이 상당히 절약되는 것을 알 수 있다.

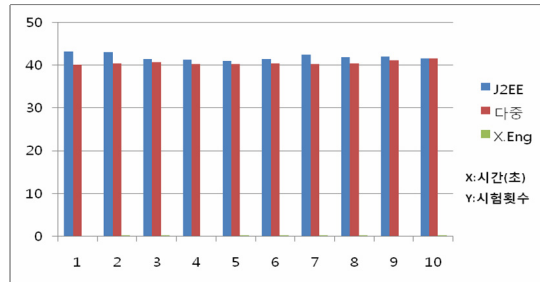
4. 성능평가



(그림 4) 1차 성능 평가 결과

다중 Tier 온라인 서비스 대량 출력 환경에서 성능 평가를 위하여 두가지 비교 실험 대상을 정하여 (그림 4)와 같이 평가를 진행하였다. 첫번째 비교대상은 Oracle 3rd Party 제품을 통한 직접 Query 후 출력된 대용량 데이터 처리 시간을 측정하였으며, 두번째 비교대상은 Presentation Tier 에서 J2EE

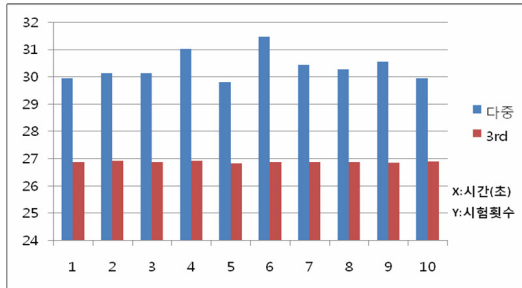
Framework[5]을 통하여 대량 데이터를 출력하도록 환경을 구성하였다. 두 비교대상 모두 물리적으로 간소화된 구조로 다중 Tier 온라인 서비스 대량출력을 위한 구조 보다 객관적으로 빠를 수 있는 구조를 가지고 있다. 1 차 성능 평가에서는 예상대로 3rd Party 제품이 다중 Tier 온라인 서비스 대량 출력 환경 보다 약 2 배 정도의 성능 우위를 보였으며 J2EE 환경으로 구성된 대량 데이터 출력 환경은 약 1.3 배 정도의 성능 우위를 보였다.



(그림 5) 2차 점검 Tier 별 성능 평가 결과

2 차 성능 평가에서는 각 Tier 별 성능에 문제를 일으키고 있는 구간을 점검하였다. (그림 5)에서 보이는 바와 같이 1 차적으로 렌더링(Rendering) 상의 시간이 걸릴 것으로 예상되었던 X-internet 엔진 구간은 Bypass 수준의 빠른 처리를 보였다. 전체적으로 부하를 많이 일으키는 구간은 Presentation Tier 와 Application Tier 로 확인 되었으며 그 중에서도 기능상 성능 개선 포인트로 Presentation Tier 내 UI Adapter 의 데이터 변환기능과 Application Tier 내 동기통신 반복이 문제점으로 확인되었다. UI Adapter 기능이 대상이 되었던 이유는 문자열 형식으로 전송된 데이터가 X-internet GRID 형식으로 데이터 변환이 일어났기 때문이었으며, Application Tier 내 동기통신 반복이 문제가 대상이 되었던 이유는 바로 통신방식 자체에 문제가 있다고 결론 지을 수 있었다. 즉 tpcall 후 클라이언트가 응답을 받을 때까지 기다리는 시간이 문제로 확인되었다. 이로써 크게 두가지 개선 포인트를 가지고 성능 개선에 들어갔으며 Presentation Tier 에서는 변환 알고리즘 개선과 Application Tier 에서는 동기 통신방식의 반복 최소화에 초점을 맞추어 기능 개선을 모색하였다. Presentation tier 에서 기능 개선 작업 후 즉, 데이터 변환 알고리즘 개선 후에도 물리적인 소요 시간을 개선하기에는 충분하지 않았다. 그래서 남은 동기 통신방식의 반복을 줄이는 부분에 초점을 두어 비동기 통신방식을 통한 반복 최소화 방안을 적용하도록 하였다. 이것이 바로 비동기 통신방식을 이용한 대용량 트랜잭션 처리 기법이다. 이 방법은 한번에 모든 대용량 데이터를 반복 Fetch 후 바로 바로 호출한 클라이언트로 전송하는 방식으로 비동기 통신방식으로 전송되므로 동기 통신방식처럼 지연시간이 없는 것이 큰 장점이다. 이 방법을

적용 후 (그림 6)과 같이 현격하게 시간이 절감되는 것을 알 수 있다.



(그림 6) 반복 개선 후 Tier 성능 평가 결과

5. 결론

지금까지 대부분 실시간 대용량 온라인 서비스를 처리하는 응용 프로그램들은 동기 통신방식을 이용한 다양한 성능 개선방안을 연구해왔다. 본 논문에서는 다중 Tier 온라인 서비스 환경에서 대용량 데이터 처리 성능 개선을 위해 기존에 방식과는 다른 관점에서 비동기 통신방식을 통한 성능 개선을 수행하였고 동일한 동기 통신방식보다 약 2.5 배 정도의 성능 개선을 할 수 있었다. 앞으로 좀 더 다양한 기술관점에서 성능 개선 포인트 연구가 필요하며 아울러 은행과 같은 대량의 트랜잭션 하에서 안정적인 대용량 처리 성능 확보를 위한 연구가 필요하다.

참고문헌

- [1] The Open Group, Distributed Transaction Processing: The XA+ Specification, Version 2, June 1994
- [2] TmaxSoft Co. Ltd, Tmax Application Development Guide, Dec 07, 2007
- [3] BEA Systems, Inc. Programming a BEA Tuxedo ATMI Application Using C, September 28, 2007
- [4] Gauce FirstRow, <http://bowwowmew.egloos.com/547609>
- [5] TmaxSoft Co. Ltd, ProFrame C V4 Developer Guide.pdf, January 10, 2008
- [6] Joel P. Kanter, "Understanding Thin-Client/Server Computing (Strategic Technology Series)", pp.3-14, Microsoft Pr, January 1998
- [7] Tim Boudreau, Jaroslav Tulach, Geertjan Wielenga, "Rich Client Programming: Plugging into the NetBeans(TM) Platform", pp.25-40, Prentice Hall PTR, May 10, 2007