



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2011년06월28일
(11) 등록번호 10-1044296
(24) 등록일자 2011년06월20일

(51) Int. Cl.

G06F 15/16 (2006.01) *H04L 12/12* (2006.01)

(21) 출원번호 10-2008-0105818

(22) 출원일자 2008년10월28일

심사청구일자 2008년10월28일

(65) 공개번호 10-2009-0090992

(43) 공개일자 2009년08월26일

(30) 우선권주장

12/035,003 2008년02월21일 미국(US)

(56) 선행기술조사문헌

"Poster: Scalable Service-Oriented Telemetry Acquisition and Distribution," 배주희 외 4인, Proceedings of Mobile and ubiquitous Systems: Networking & Services, 2007, pp.1-3 (2007.08.10.)

(73) 특허권자

한국산업기술평가관리원

서울 강남구 역삼동 701-7

인터내셔널 비지네스 머신즈 코퍼레이션

미국 10504 뉴욕주 아몬크 뉴오차드 로드

(72) 발명자

배 주희

대한민국 서울

탁 영주

대한민국 서울

(뒷면에 계속)

(74) 대리인

신정건, 김태홍

전체 청구항 수 : 총 3 항

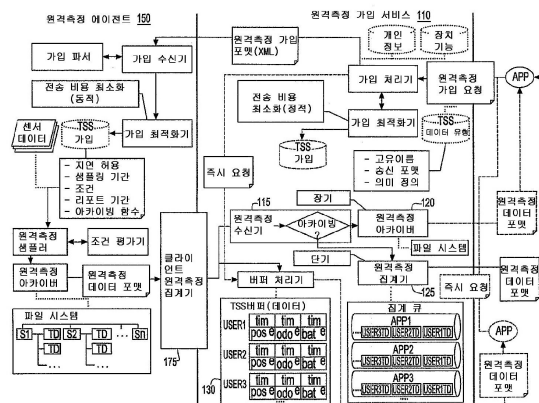
심사관 : 이종익

(54) 무선 네트워크를 통한 계기 시스템 엔티티들로부터의 크기 조정되는 원격측정 데이터의 획득

(57) 요약

원격측정 데이터 취득 관리 시스템은 센서 포함 시스템들과 인터페이스 하기 위해 시스템 프로세서 및 시스템 프로세서에 의해 제어되는 인터페이스 엔진을 포함하여, 센서 포함 시스템들로부터 원격측정 데이터를 수신하고, 이 데이터를 각각의 센서 포함 시스템의 프로토콜에 따라 원격측정 데이터를 처리하도록 동작하는 복수의 데이터 처리 시스템 및 애플리케이션에 분산시킨다. 이 원격측정 데이터 취득 관리 시스템은 복수의 데이터 처리 시스템 또는 애플리케이션이 "퍼지" 시의성 프로토콜을 포함하는 원격측정 요청들을 지정하게 할 수 있고, 시스템 부하에 상응하는 사용량에 따라 센서 포함 시스템들에게 비용을 청구하도록 하기 위한 요금부과 구조를 생성한다.

대표도



(72) 발명자

하우슨 크리스토퍼 휴
미국 뉴욕 슬리피 할로우
문선 조나단
미국 뉴욕 팻남 벨리

이 원일

대한민국 서울

특허청구의 범위

청구항 1

센서 포함 시스템들과 인터페이스하고, 상기 센서 포함 시스템들로부터 원격측정(telemetry) 데이터를 수신하여 원격측정 요청에서 제공된 애플리케이션의 지연-허용 사양(delay-tolerance specification)에 따라 복수의 데이터 처리 애플리케이션에 분산시키는 원격측정 데이터 취득 관리 시스템으로서, 상기 원격측정 데이터 취득 관리 시스템에 대해 청구된 상기 요청에 응답하는 부하에 따라 상기 데이터 처리 애플리케이션에 대해 비용을 청구하는 것인, 상기 원격측정 데이터 취득 관리 시스템에 있어서,

동일 센서 포함 시스템들로부터의 동일 데이터 항목들에 대한 다수의 애플리케이션의 가입들을 결합함으로써, 센서 포함 엔티티 통신을 최적화하기 위한 수단,

상기 센서 포함 시스템들에 대한 데이터 관독 조건들을 평가하여, 상기 데이터 처리 애플리케이션들이 상기 조건들을 충족시킬 때에만 센서 관독 데이터를 수신하도록 하기 위한 수단,

데이터 관독 조건들을 사전-검파일링하여 다수의 이벤트에 대해 관독 오버헤드를 확산시키기 위한 수단, 및

수신된 원격측정 데이터를 단기간 동안 버퍼링하고, 상기 애플리케이션의 지연-허용 사양을 위반하지 않으면서 단일의 벌크 트랜잭션내의 대응하는 요청 데이터 처리 애플리케이션에 전달하기 위한 수단

을 포함하는 원격측정 데이터 취득 관리 시스템.

청구항 2

센서 포함 시스템들과 인터페이스하고, 상기 센서 포함 시스템들로부터 원격측정(telemetry) 데이터를 수신하여 원격측정 요청에서 제공된 애플리케이션의 지연-허용 사양(delay-tolerance specification)에 따라 복수의 데이터 처리 애플리케이션에 분산시키며, 상기 요청에 응답하여 부과되는 부하에 따라 상기 데이터 처리 애플리케이션에게 비용을 청구하는 방법을 수행하기 위해 처리 장치에 의해 실행되는 명령어들을 저장하고, 상기 처리 회로에 의해 관독가능한 컴퓨터 관독가능 기록매체에 있어서, 상기 방법은,

동일 센서 포함 시스템들로부터의 동일 데이터 항목들에 대한 다수의 애플리케이션의 가입들을 결합함으로써, 센서 포함 엔티티 통신을 최적화하는 단계,

상기 센서 포함 시스템들에 대한 데이터 관독 조건들을 평가하여 상기 데이터 처리 애플리케이션들이 상기 조건들을 충족시킬 때에만 센서 관독 데이터를 수신하도록 하는 단계,

데이터 관독 조건들을 사전-검파일링하여 다수의 이벤트에 대해 관독 오버헤드를 확산시키는 단계, 및

수신된 원격측정 데이터를 단기간 동안 버퍼링하고, 상기 애플리케이션의 지연-허용 사양을 위반하지 않으면서 단일의 벌크 트랜잭션내의 대응하는 요청 데이터 처리 애플리케이션에 전달하는 단계

를 포함하는 것인, 컴퓨터 관독가능 기록매체.

청구항 3

제1항에 있어서,

상기 원격측정 데이터 취득 관리 시스템 프로세서는 센서 관독을 위한 센서 포함 시스템상들에서의 부하를 줄이고 무선 네트워크상에서의 부하를 줄이도록 동일 엔티티로부터의 동일 데이터 항목에 대한 다수의 애플리케이션의 가입들을 결합하고, 센서 포함 시스템들에 대한 데이터-관독 조건들을 평가하여 상기 데이터 처리 애플리케이션들이 상기 조건들을 충족시킬 때에만 센서 관독을 수신하고, 시스템 부하를 줄이고, 상기 센서 관독을 수신하는 상기 복수의 데이터 처리 시스템 또는 애플리케이션들 중 임의의 것의 처리 부하를 줄이고, 상기 데이터-관독 조건들을 사전 검파일링하고, 원격측정 이벤트를 처리함에 의해 처리 오버헤드를 다수의 이벤트에 대해 확산시키고, 단기간 동안 다수의 센서 포함 시스템들로부터의 관독에 의해 데이터 처리 시스템들 또는 애플리케이션들의 처리 부하를 줄이고 나서 관련된 데이터 처리 시스템 또는 애플리케이션에 전달하고, 사용자들의 그룹별 가입을 이용하여 데이터 처리 시스템들 또는 애플리케이션들의 처리 부하를 줄이는 것인, 원격측정 데이터 취득 관리 시스템.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 무선 네트워크를 통해 계측 엔티티로부터의 크기조정가능한 원격측정 데이터를 획득하는 것을 지원하는 시스템 및 방법에 관한 것이다.

배경 기술

[0002] 센서 네트워크 미들웨어의 역할들 중 하나는 센서-포함(sensor-bearing) 엔티티로부터 데이터를 취득하고, 취득된 센서 데이터를 애플리케이션 요건에 따라 센서 데이터를 처리하는 각종 애플리케이션에 분산시키는 데 있다. 미들웨어는 서비스로서 배치되거나, 애플리케이션들에 의해 인터넷을 통해 액세스가능한 서비스들을 제공하는 것으로 알려져 있다. 미들웨어는 무선 네트워크를 통해 많은 수(예컨대, 수백만)의 센서-포함 엔티티(휴대 전화, 자동차, 등)에 대해 각종 애플리케이션 액세스를 제공한다. 미들웨어는 각종 애플리케이션과 센서-포함 엔티티 사이에서 게이트 키퍼(게이트웨이)로서 기능을 한다. 이런 형태의 배치(depoyment)는 미들웨어가 인터넷으로의 액세스, 및 미들웨어 서비스(들)를 액세스하기 위해 필요한 사용 권한을 갖는 임의 애플리케이션에 액세스가능 하기 때문에 바람직하다. 그러나, 이런 공지된 미들웨어 서비스를 배치함에 있어 단점이 없는 것은 아니다.

발명의 내용

해결 하고자하는 과제

[0003] 많은 수의 클라이언트에 대해 액세스를 제공하도록 미들웨어를 배치하고, 각종 애플리케이션이 그들의 특정 목적을 위해 빈번한 센서 판독을 요청할 수 있다는 사실을 감안해 볼 때, 미들웨어는 매우 높은 데이터 속도를 지원(서비스)해야 한다. 이런 매우 높은 데이터 속도는 "기존의(off-the-shelf)" 애플리케이션 서버 기술 및 고 처리량 요건에 맞도록 설계되어 있지 않은 애플리케이션들을 압도할 수 있다. 다수의 애플리케이션이 동일한 클라이언트로부터의 데이터에 가입할 수 있으므로, 클라이언트는 그가 가입들을 관리하는 것을 도와주는 기술이 없는 상태에서 오버로드될 수 있다. 따라서, 클라이언트와 미들웨어 서버 간에서의 무선 링크들은 오버로드될 수 있거나, 또는 클라이언트로부터 미들웨어로의 데이터 흐름이 최적화되어 있지 않은 경우에는 미들웨어 제공자에게 상당량의 비용을 들게 한다. 결과적으로, 애플리케이션들은 많은 수의 클라이언트의 원격측정 스트림에 의해 오버로드된다.

[0004] 미들웨어의 효율을 향상시키려는 노력은 애플리케이션에 도달하는 데이터의 시의성(timeliness)을 감소시킬 수 있어, 이런 상황 하에서 애플리케이션의 시의성 요건을 보존하기 위한 노력이 요구된다. 그리고, 센서 네트워크 미들웨어의 분야가 당 기술분야에서의 공지된 광범위한 단점을 해결하는 개선된 센서 네트워크 설계와 함께 활발하게 발전하고 있지만, 이런 형태의 확장성 과제를 해결하는 해결책은 알려져 있지 않다.

[0005] 그러므로, 당 기술분야에서는 종래 기술의 단점을 극복하고, 계측 엔티티로부터 원격측정 데이터를 취득하도록 동작하는 시스템 및 방법의 필요성이 명백하게 대두되고 있다.

과제 해결수단

[0006] 이런 목적을 위해, 본 발명은 애플리케이션들이 그들이 원격측정 요청에서 "퍼지(fuzzy)" 시의성 사양을 지정할 수 있게 하고, 제공자들이 네트워크 상에 청구하는 부하에 따라 제공자들에게 비용을 청구하는 요금부과 구조를 생성한다. 본 발명의 시스템은 센서 판독을 위한 엔티티에서의 부하 및 무선 네트워크에서의 부하가 줄어들도록, 동일 엔티티로부터의 동일 데이터 항목에 대한 다수의 애플리케이션의 가입을 결합한다. 서버 전용, 엔티티 전용, 서버/엔티티 결합을 포함하여 다수의 가입 "최적화" 메커니즘(최적화를 위해 엔티티에서의 능력에 의존)이 이용가능하다. 본 발명의 시스템은 엔티티에 대한 데이터-판독 조건을 평가하여, 애플리케이션들이 이 조건을 충족시킬 때에만 센서 판독을 수신하도록 하게 하여 센서 판독을 위한 엔티티에서의 부하 및 무선 네트워크에서의 부하를 경감시킨다. 본 발명의 시스템은 엔티티에서 단순한 조건 평가를 준비하도록 서버에서 상기 조건을 사전-컴파일링하고, 다수의 이벤트에 대해 원격측정 이벤트를 처리하는 오버헤드를 분배한다. 클라이언트 원격측정 집계기(시스템 모듈)는 클라이언트로부터의 원격측정 송신들을 단기간 동안 버퍼링한 후, 애플

리케이션의 지연-허용 사양을 위반하지 않으면서 그들을 단일의 벌크 트랜잭션으로 원격측정 수신기에 전달한다.

[0007] 본 발명의 시스템은 그룹 가입을 이용하여 애플리케이션의 처리 부하를 줄임으로써, 그룹 내의 엔티티에서의 센서 관독들이 거의 동기화되어, 서버에서의 수집이 비동기화된 경우보다는 많은 수의 엔티티로부터 데이터를 수집할 수 있어, 애플리케이션에 보내지는 데이터의 "밀도(density)"가 증가한다. 본 발명의 시스템은 엔티티에 대해 무선 네트워크를 횡단하는 비용이 드는 요청을 행하는 대신, 요청된 데이터가 정해진 "신선도(freshness)" 보증 기간 내에 관독되면 요청된 데이터를 반환하는 캐싱 메커니즘을 포함하여, 1-회 관독 요청을 효율적으로 지원한다(가입 또는 또 다른 1-회 요청을 만족시키기 위함). 본 발명의 시스템은 또한 무선 네트워크의 부하를 줄이기 위해 엔티티에 아카이빙(archiving) 함수를 분산시키는 아카이빙 메커니즘 및 관련된 요약화 기능들에 의해, 데이터 수집만을 필요로 하고 실시간 데이터 제공은 필요로 하지 않는 애플리케이션을 지원한다. 엔티티들은 가능한 많이 아카이빙을 수행하고, 최종 아카이빙을 위해 서버에 아카이브된 결과들을 부분적으로 전달한다.

[0008] 기존의 센서 네트워크 미들웨어(SNM)는 이런 기능들을 제공하지 않는다. 여타의 SNM 해결책을 이용하는 애플리케이션들은 이들 기능들을 그들 자신들이 제공해야 할 것이며, 대부분의 이들 기능들은 다수의 애플리케이션의 이용을 지원하므로, 애플리케이션-제공된 해결책이 최적은 아닐 것이다.

발명의 실시를 위한 구체적인 내용

[0009] 본 발명의 상기한 목적 및 기타의 목적들, 특징들 및 이점들은 도면을 참조하여 본 발명의 실시예에 대한 이하의 설명으로부터 이해할 수 있을 것이다.

[0010] **개요**

[0011] 본 발명의 원격측정 데이터 취득용 시스템은 주된 두 유형의 요청: 원격측정 데이터의 주기적 전달 요청 및 원격측정 데이터의 1회 전달 요청을 지원한다. 도 1은 원격측정 데이터 취득의 두 유형을 강조하는 본 발명의 시스템에 대한 기능적인 설명이다. 1회 요청의 시스템 흐름은 대시선으로 도시되는 한편, 주기적인 전달 요청을 처리하는 시스템 흐름은 실선으로 도시된다.

[0012] **가입 요청(주기적)**

[0013] 클라이언트의 원격측정 데이터에 대한 가입이 수신되면, 원격측정 가입 서비스(TSS)(110)는 최적화된 집합을 생성하기 위해 서로에 상관하여 가입을 조정할 목적으로, 우선 그 가입을 동일 클라이언트 애플리케이션의 다른 가입에 대해 분석한다(섹션 2 참조). 이어서, TSS(110)는 다른 가입들에 대한 변경으로 인해 요구되는 임의 업데이트와 함께, 그 가입을 클라이언트 애플리케이션에 전송한다.

[0014] 클라이언트 애플리케이션에서, 원격측정 에이전트(150)는 가입들의 어느 샘플링 요건들을 병합시킬 수 있는가를 판단하기 위해 별도의 최적화 분석을 행한다. 가입을 할 때, 그 가입에 관련된 임의 조건식을 평가하며, 그 조건식이 반환되면, 즉 참(True)이면, 샘플링된 데이터는 가입에 아카이빙 사양이 존재할 경우에는 아카이브되거나, 또는 클라이언트 원격측정 집계기(175)로 즉시 전송된다.

[0015] 클라이언트 원격측정 집계기(175)는 원격측정 데이터 송신을 HTTP POST로서 수신하고, 원격측정 수신기(115)에 전달하기 전에 단기간 동안 데이터 송신을 버퍼링한다. 버퍼링의 목적은 많은 수의 개별 송신에 대해 원격측정 수신기의 호출별 오버헤드(per-call overhead)를 분배하기 위한 것이다. 원격측정 수신기는 데이터 송신이 아카이빙 가입인 경우에는 원격측정 아카이버(120)로 그 데이터 송신을 전달하거나, 원격측정 집계기(125)로 전달한다. 원격측정 집계기는 애플리케이션별로 클라이언트 원격측정 집계기(175)와 유사한 기능을 수행한다. 이 경우, 집계는 주로 애플리케이션을 도와주기 위한 것으로, 이는 집계기 없을 때보다 더 많은 원격측정 호출별 이벤트들을 처리할 수 있을 것이다.

[0016] 그룹 가입 요청들은 우선, 지정된 그룹 내의 사용자들을 열거하고 각 사용자마다의 단일-사용자 가입을 입력함으로써 처리된다. 추가적인 한 단계는 그룹 가입을 위한 별도의 가입 레코드를 생성하는 것을 포함한다. 이는 TelemetrySubscriptionServiceImpl 클래스의 "reportTelemetryForGroup" 메소드에 대한 설명에서 상세히 기술되어 있다.

[0017] **1회 요구**

[0018] 클라이언트의 데이터 또는 클라이언트 그룹의 데이터에 대한 1회 원격측정 샘플링 요구는 캐시된 데이터 관독이

얼마나 보존될 수 있는가를 상술하는 "신선도(freshness)" 파라미터를 포함하며, 애플리케이션에 대해 허용가능한 상태로 남아 있다. TSS(110)는 우선 데이터에 대해 TSSBuffer(130)를 검사한다. TSSBuffer에서 그 데이터가 발견되지 않거나, 발견된 데이터가 너무 오래된 것이면, 시스템은 클라이언트에 그 데이터에 대한 요청을 전송한다. 그 요구가 1회 요구인 것을 나타내는 필드가 참(True)으로 설정된 것을 포함하여, 요구에 대한 가입 레코드가 생성된다. 1회 요청의 원격측정 데이터가 원격측정 수신기(115)에 도달하면, 그 원격측정 데이터는 2번의 예외를 제외하곤 가입 요청에 대한 데이터와 같이 처리된다. 첫째는, 가입 레코드 캐싱되지 않는 데, 그 이유는 다시 판독되지 않을 것이기 때문이며, 둘째는, 가입 레코드가 삭제된다.

[0019] 가입 최적화

[0020] 본원에 기술된 바와 같은 데이터-취득 시스템에서는, 임의 한 시점에서 클라이언트의 원격측정 데이터를 요청할 수 있는 애플리케이션의 수는 제한적이지 않다. 각 애플리케이션의 원격측정 가입이 다른 모든 가입들과 독립적으로 실행되면, 클라이언트는 가입을 실행하는 작업(task)에 의해 압도될 가능성이 있으며, 무선 네트워크를 통해 데이터를 송신하는 비용은 터무니없이 비싸게 될 수 있다. 본 발명의 시스템은 "가입 최적화" 메커니즘 또는 프로세스를 포함함으로써 이러한 단점을 해결하며, 이 프로세스는 가입 집합을 조정된 방식으로 실행하여, 전체 비용(또는 시스템 부하결림)이 상당히 경감된다. 본 발명의 시스템은 가입 최적화를 위한 3 개의 독립된 기술 또는 프로세스를 포함한다.

[0021] 간격 범위 매칭

[0022] 각각의 가입은 원격측정 샘플들 간에서 원하는 간격, 즉 1) 최소 간격과 2) 최대 간격의 2 가지 사양을 포함한다. 간격 범위 매칭 프로세스는 각 그룹마다, (그룹의) 각 가입의 샘플링 범위가 그 그룹에서 다른 모든 가입과 겹치도록 가입 그룹을 생성한다. 이렇게 함으로써, 클라이언트는 그룹에서의 모든 가입을 동시에 샘플링 및 전달할 수 있다.

[0023] 서버는 클라이언트에 그룹화를 명시적으로 전달하지 않는다. 오히려, 가입이 속하는 그룹은 원격측정 요청 메시지의 "가입 및 갱신" 엘리먼트의 optInterval 속성에 의해 표시된다. 클라이언트들은 그들이 수신한 가입 집합 내의 각각의 optInterval의 다른 값의 가입 그룹을 유지한다.

[0024] 간격 범위 매칭 함수 또는 프로세스는 (애플리케이션이 원격측정에 가입할 때 인보크되는) 프로시저 addSubscription 및 (애플리케이션이 가입을 중지할 때 인보크되는) deleteSubscription에 의해 지정된다. 예를 들어, S는 한 클라이언트에 대한 원격측정 가입 집합으로서 정의되며, 여기서 S 내에서의 각각의 s는 최소 샘플링 간격 $i_{\min}$, 최대 샘플링 간격 $i_{\max}$, 및 "최적화된" 샘플링 간격 i_{opt} 를 갖는다. 그룹 G의 샘플링 범위는 다음과 같이 정의된다. 즉

[0025] $[i_{\min}, i_{\max}]$ 여기서 G 내에서의 각 s의 경우, $i_{\min} = s.i_{\min}$ 및 $i_{\max} = s.i_{\max}$

[0026] 각 G는 또한 $i_g = i_{\min}$ 및 $i_g = i_{\max}$ 가 되도록 선택된 그룹 샘플링 간격 i_g 를 갖는다.

[0027] 프로시저 addSubscription(s)

[0028] 각 샘플링 그룹 G마다,

[0029] s의 샘플링 범위가 G의 샘플링 범위와 겹치면,

[0030] G에 s를 추가하고 G의 샘플링 범위를 갱신.

[0031] s가 어떤 그룹에도 추가되지 않으면, 새로운 그룹을 생성하여 s를 그것에 추가.

[0032] 원격측정 요청 메시지를 생성하여 그것에 "부가된 s"를 추가.

[0033] 그룹 s의 I_g 가 부가되어 결과적으로 변경되었으면,

[0034] G 내의 각 s에 대해 "갱신된 s_g "를 요청 메시지에 추가.

[0035] 요청 메시지를 클라이언트에 전송.

[0036] 프로시저 deleteSubscription(s)

[0037] S 및 그것이 존재하는 그룹 G에서 s를 제거.

- [0038] 원격측정 요청 메시지를 생성하고 그것에 "중지된 s"를 부가.
- [0039] 샘플링 그룹들을 재생성. 각 G 내의 각 가입 sg에 대해,
- [0040] $s_g.i_{opt}$ 가 변경되면, G 내의 각 s에 대해 "갱신된 s_g " 를 요청 메시지에 부가.
- [0041] 요청 메시지를 클라이언트에 전송.

[0042] **다음 마감일(Deadline)**

- [0043] 이 프로세스에서는, 샘플링에 대해 "영원한(for-all-time)" 스케줄을 생성하기보다, 클라이언트는 주어진 시점에서 모든 가입을 주목하여 가장 가까운 마감일을 찾는다. 그래서, 클라이언트는 타이머를 그 시점에서 그 자신을 웨이크(wake)하도록 설정한 후 슬립(sleep)으로 들어간다. 클라이언트가 웨이크될 때, 클라이언트는 그 시점에서 샘플링되기에 적절한 모든 가입을 샘플링하고, 그 다음 웨이크 시간을 계산하고 나서, 슬립으로 들어간다.

- [0044] 새로운 가입이 도달하면, 클라이언트는 그 가입의 샘플링을 언제 시작해야 할지를 결정할 수 있다. 클라이언트가 새로운 가입에 "가장 근사한" 다른 가입에 따라 간격 범위 매칭을 이용하여 샘플링-시작 시간을 결정한다.

[0045] **조건 평가**

- [0046] 애플리케이션들은 정해진 센서들을 자주 샘플링하길 원할 수 있지만, 정해진 조건이 발생할 때는 단지 그들의 값을 통지받을 수만 있다. TSS(110)는 조건식을 공통 관계 및 논리 연산자 뿐 아니라, 샘플 시퀀스에 대해 동작하는 특수 함수들, 즉 변경, percentChange, 평균값, 중간값 및 기타를 원격측정 요청에 포함시킬 수 있다.

[0047] **사양**

- [0048] 조건식은 데이터가 자주 샘플링되는 것을 허용하지만, 데이터가 정해진 조건을 만족할 경우에만 애플리케이션에 보고되도록 한다. 조건식은 논리 연산자, 관계 연산자 및 정해진 함수들의 집합으로 구성된 부울식(인픽스 infix 표기법으로)이다. 조건식에 대한 문법(grammar)은 다음과 같이 확인되며, 여기서 표기법 "[...]"는 0 또는 1회 발생을 의미하고, "{...}"는 0회 이상의 발생을 의미한다.

- [0049] $\langle \text{condition} \rangle \rightarrow \langle \text{relation} \rangle \{ \langle \text{logical op} \rangle \langle \text{relation} \rangle \}$
- [0050] $\langle \text{relation} \rangle \rightarrow \langle \text{term} \rangle [\langle \text{relational op} \rangle \langle \text{term} \rangle]$
- [0051] $\langle \text{term} \rangle \rightarrow \langle \text{primary} \rangle$
- [0052] $\rightarrow ! \langle \text{primary} \rangle$
- [0053] $\langle \text{primary} \rangle \rightarrow \langle \text{literal} \rangle$
- [0054] $\rightarrow \text{DataName} [. \text{DataPartName}]$
- [0055] $\rightarrow \langle \text{function} \rangle$
- [0056] $\rightarrow (\langle \text{condition} \rangle)$
- [0057] $\langle \text{function} \rangle \rightarrow \text{FunctionName} (\langle \text{primary} \rangle , \{ \langle \text{primary} \rangle \})$
- [0058] $\langle \text{logical op} \rangle \rightarrow \&\&$
- [0059] $\rightarrow \parallel$
- [0060] $\langle \text{relational op} \rangle \rightarrow ==$
- [0061] $\rightarrow !=$
- [0062] $\rightarrow <$
- [0063] $\rightarrow <=$
- [0064] $\rightarrow >$
- [0065] $\rightarrow >=$

[0066] 여기서,

[0067] DataName: 원격측정 데이터 사전으로부터의 원격측정 데이터 항목 이름.

[0068] DataPartName: 원격측정 데이터 부분의 이름(복합 데이터의 경우).

[0069] FunctionName: 표 1의 함수 이름.

[0070] TOPAZ는 다음의 조건 함수들을 정의한다. 클라이언트에게 보내진 포스트픽스(postfix) 식에서 간결한 형태의 함수 이름을 사용하며, 이것은 괄호에서 나타난다.

함수	파라미터	설명
Change(x, n)[CH]	x: 스칼라 식 n: 슬라이딩 윈도우내의 샘플들의 수(상수입)	슬라이딩 윈도우 내의 샘플들에 대해 x를 변경
percentChange(x, n)[PCH]		제1 샘플값에 상관한 백분율로서, 슬라이딩 윈도우 내의 샘플들에 대해 x를 변경
Average (x, n)		슬라이딩 윈도우에서 x의 산술 평균
Median(x, n)[MED]		슬라이딩 윈도우에서 x의 중앙값
min(x, n)[MIN]		슬라이딩 윈도우에서 x의 최소값
max(x, n)[MAX]		슬라이딩 윈도우에서 x의 최대값
countTrue(e, n)[CNT]	e: 부울식 n: 슬라이딩 윈도우에서의 샘플들의 수(상수입)	시간의 수 e는 슬라이딩 윈도우에서 참임. (우리가 파라미터 resetOnTrue를 추가해야 하는가?)
NumTrue(e1,...en)[NTR]*	e1,...en:부울식의 목록	참인 e1,...en에서의 식의 수
abs(x)	x: 스칼라식	x의 절대값을 반환

[0072] * numTrueIn 함수에 대한 인수의 변수 때문에, 식은 포스트픽스의 2진 형태로 변환된다. 논리적으로, 식

[0073] numTrueIn(e1, e2,...en)은

[0074] 우선적으로 incrTrue(e1, incrTrue(e2,...,incrTrue(en), 0)...)으로 다시 쓰여지며,

[0075] 으로 다시 쓰여지며,

[0076] 여기서, incrTrue(e, n)은 e가 참이면 n + 1을 반환하고, e가 거짓이면 n을 반환한다. 이어서, 이 식은 포스트픽스로 변환될 수 있다. incrTrue에 대한 포스트픽스 축약은 NTR이다.

[0077] 구현

[0078] TSS 클라이언트는, 가능하다면, 조건-식 평가를 구현해야 한다. 그러나, 모든 클라이언트가 이를 행할 수 있는 것은 아닐 것이며, 대역폭이 CPU 사이클보다 싼 경우가 있을 수 있다. 따라서, TSS는 클라이언트들이 원격측정 가입 요청을 수락할 때, 클라이언트들이 그들의 응답에서 그들이 조건식 평가를 수행할 것인지의 여부를 나타내는 모델을 이용한다. TSS는 가입을 위한 레코드에 그들의 응답을 기록한다. 클라이언트가 조건식을 평가하지 않기로 결정하면, 서버에서의 원격측정 수신기가 조건을 평가할 것이다.

[0079] 대부분의 조건식 함수들은 "윈도잉(windowing)" 함수들인데, 그들은 고정된 크기의 데이터 이력에 기초하여 그들의 값을 계산한다. 따라서, 이들 함수는 상태를 유지할 필요가 있다. 이를 위한 다수의 가능한 메커니즘이 있다. 본 실시예에서는, 조건식이 식 트리로서 실현되며, 여기서 트리 내의 각 연산자 및 함수는 (논리적으로) 별개인 객체로서 실현된다. 각 윈도잉-함수 객체는 그 자신의 상태를 저장한다.

[0080] 도 2는 조건식 처리를 도시한다. 서버에서 "requestTelemetry"를 수신하면, 본 발명은 인픽스 식을 구문분석하고(parse), 그것의 유효성을 검증하고, 그것을 포스트픽스 형태로 변환하고, 가입 데이터베이스에 포스트픽스 형태를 저장하고, 포스트픽스 형태를 클라이언트에 전송한다. 클라이언트가 식을 수신하면, 본 발명은 TOPAZ_TelemetryConditionExpression의 클래스들을 이용하여 포스트픽스 형태를 식 평가 트리로 구문분석하고, 식-트리 객체를 가입 객체에 첨부한다. 가입 데이터가 클라이언트에서 샘플링될 때, 본 발명은 식 트리 상의 "evaluate(TelemetryDataSet)"를 호출함으로써 조건을 평가한다. 결과가 참이면, 전과 같이 전송한다. 서버에서 원격측정이 수신되는 경우, 처음번째이면, 포스트픽스 식을 식 평가 트리로 구문분석한다(다음 슬라이드 상

의 노트를 참조). 식-트리 객체를 캐싱된 가입 객체에 첨부한다. 식은 상태 유지될 수 있다. 그러나, 우리는 com.ibm.websphere.cache.DistributedMap으로 관리되는, 다수의 TelemetryReceiver 노트들을 가질 것이다. 식 트리 상의 "evaluate(TelemetryDataSet)"를 호출함으로써 조건을 평가한다. 결과가 참이면, 아카이빙 또는 집계로 진행한다. 평가 결과에 관계없이 데이터에 대한 비용을 청구한다.

[0081] 파서(parser) 구현은 JavaCC를 이용하여 Infix 파서가 생성된 것을 포함하며, Postfix 파서는 단순히 StringTokenizer 및 반복 하향 기술(recursive descent technique)을 이용한다. 식이 특정 클라이언트에 대해 유효한가를 검사하기 위함(각 장치의 유형은 정해진 원격측정 집합만을 제공할 수 있으므로, 각 클라이언트 버전은 정해진 함수 집합만을 구현하고, 일부 클라이언트는 조건식을 전혀 지원하지 않을 수 있다). 가입 요청이 수신되면, 유효성 검사가 행해진다. 이런 검사에 의해 애플리케이션에 대해 직접적인 오류 메시지가 인에이블링되어, 조건 평가 시에 이들 오류를 처리하는 것을 피하게 된다. 각 장치/클라이언트가 행할 수 있는 것이 기록된다. 사용자/장치 리소스 서비스는 이를 행한다. 본 발명은 유형별로, 장치별로, 또는 TOPAZ-클라이언트 버전별로 기록한다. TA는 별도의 버전 넘버를 가질 수 있다.

[0082] 본 발명은 원격측정 수신기가 언제 그것이 식을 평가할 필요가 있는지를 알아야 하는 것을 제공한다. 가입을 수신하여 처리할 때 클라이언트 기능성을 검사하므로, 판정은 그 때 행해질 수 있다. 조건식은 TR이 조건식을 평가할 필요가 있으면 DB에만 저장된다. 또는, 조건식을 어떤 식으로든 기록하면, 레코드에 부울 필드를 추가하거나, 조건식에 "!"와 같은 것을 말머리에 추가시킨다. 또는, 클라이언트 응답은 가입을 평가하기 위해 그 능력 및 태도를 나타낸다. 본 발명은 오류 메시지를 포함하는 특수 원격측정 리포트를 정의함으로써 클라이언트에서의 평가 예외를 제공한다.

[0083] 도 3은 평가 식 트리이다. 각 노드는 ConditionExpressionNode 인터페이스를 구현한다. 예를 들어, 한 메소드에서는, public void evaluate(TelemetryDataSet)를 구현한다. 또한, 각각은 적절한 유형을 반환하는 agent<type>Result() 메소드를 구현한다(boolean, float, or int... need double?). 더 많은 소정의 노드들이 상태 유지된다(원도잉 함수를 구현하는 것들). 그러므로, 논리 연산자들은 간단한 평가를 행하도록 구현되어야 한다. 평가 프로시저에서, 본 발명은 루트 노드 상의 evaluate()를 호출한다. 이는 새끼 노드들 상의 evaluate()를 호출하고, 이는 그들의 새끼 노드들 상의 evaluate()를 호출 등을 행한다. evaluate()에 대한 각 호출은 get<type>Result에 대한 호출을 추종하여 평가의 결과를 검색한다.

[0084] 클라이언트 원격측정 집계

[0085] 개별 클라이언트의 관점으로부터, 각각의 원격측정 송신은 별개의 이벤트이다. 수백만의 클라이언트로부터의 원격측정 스트림을 처리할 수 있는 서버의 경우, 별개의 이벤트로서 각 원격측정 송신을 처리하는 것은 이벤트를 처리함에 있어 초래되는 오버헤드로 인해 매우 불충분하다. 오버헤드는 I/O 버퍼들로부터 데이터를 판독하는 것과, 이들 버퍼로부터 애플리케이션 서버 판독을 인보크하는 것을 포함하고, 애플리케이션 서버는 데이터를 처리하기 위한 코드를 인보크하는 데, 이들은 가장 개략적인 수준에서의(at the grossest level) 단계들이다. 따라서, 대다수의 클라이언트를 비용 효율적으로 처리하기 위해서는, 대다수의 이벤트에 대해 대다수의 클라이언트로부터의 원격측정을 처리하는 비용을 분배시키는 메커니즘을 제공할 필요가 있다. 본 발명은 이를 행한다.

[0086] 집계 아키텍처

[0087] 원격측정 가입 서비스 아키텍처는 수백만 클라이언트의 집단(population)을 지원하는 것을 목표로 한다. 이 아키텍처는 느슨하게 결합된 웹 서비스 노드 집합을 이용하며, 이들 각각의 서비스 노드는 클라이언트 집단의 부분집합(subset)을 서비스한다. 개별 클라이언트들이 단문 이벤트 메시지를 송신하므로, 집합체에는 매우 많은 수의 이벤트가 존재한다. 이 아키텍처는 크로싱(crossing) 시스템이 경계들과 인터페이싱할 때 이벤트들을 더 큰 메시지로 집계하길 시도한다. 이 아키텍처는 서로 다른 스테이지들에 시스템 기능을 분산시켜 이벤트들의 누적을 집합체로 하고, 이 집합체가 스테이지 사이에서 전달된다. 부하 밸런싱 및 높은 가용성을 지원하기 위해 클러스터 내의 여러 노드 그룹에 걸쳐 여러 스테이지를 분산시킨다. 클러스터 멤버들은 수평 크기조정 토폴로지를 지원하는 다수의 머신들 간에 위치된다. 이것에 의해, 여러 머신들이 단일 논리 이미지로서 제공될 수 있다.

[0088] 현 시스템에는 주된 3 개의 스테이지가 있다. 원격측정 집계 스테이지는 클라이언트로부터의 이벤트에 대한 초기 진입점이다. 이곳으로부터 집합체 메시지가 주 TOPAZ 서비스 스테이지로 전달된다. 지속성 스테이지를 이용하여 데이터베이스로의 액세스를 중재한다.

[0089] 클라이언트 통신

[0090] TOPAZ 통신 모델에서, 단일 클라이언트는 TOPAZ 서버로 진행되는 여러 개의 원격측정 스트림을 가질 수 있다. 그러나, 시스템을 단순화하기 위해, TOPAZ와의 모든 클라이언트 통신은 동일한 TOPAZ 노드를 목표로 한다. 참조 구현을 위해, 클라이언트는 TCP를 통한 HTTP를 이용하여 TOPAZ와 통신한다. 그래서, 클라이언트로부터의 모든 애플리케이션 스트림 이벤트들이 동일한 TCP 연결을 통해 전송된다.

[0091] 캐싱 프록시

[0092] 원격측정 집계 에이전트(TAA)는 제1 스테이지이며 캐싱 프록시를 이용하여 구현한다. 이는 J2EE 애플리케이션 서버 앞에 위치하는 WebSphere 서비스로서, 이것의 목적은 공통의 요청 수신을 신속하게 만족시키기 위한 것이다. 수신 이벤트는 원격측정 집계기에 도달하고, 유효성을 검사하고, 그들의 목표 TOPAZ 노드에 전송되도록 큐잉된다. 집계 요청은 원격측정 집계기로부터 목표 TOPAZ 노드로 전송된다. 집계 응답을 수신하면, 그 요청에 대해 계류중인 이벤트 응답은 각자 클라이언트에 반송되고, 임의 큐잉된 요청들이 집계되어 목표 노드에 전달된다. 목표 노드에 대해 남아 있을 수 있는(outstanding) 최대수의 집계 요청이 있다.

[0093] TSS 아키텍처는 주어진 클라이언트의 모든 원격측정이 클라이언트의 활동 세션의 오랜 기간 동안 하나의 TOPAZ 노드에 전달될 것을 요구한다. 이것에 의해, 원격측정 수신기 노드는 원격측정 가입 레코드를 캐싱할 수 있으므로, 데이터베이스로의 과도한 액세스가 방지된다. TAA 상에 이것이 제공되는 부담은 TAA에 의해 서비스되는 클라이언트로부터 데이터를 수신하는 각 TOPAZ 노드마다 클라이언트 원격측정을 따로따로 큐잉해야 한다는 것이다. TAA는 세션 친화성 정보, 즉 특정 TOPAZ 노드에 클라이언트 원격측정 스트림을 할당하는 JSESSIONID 쿠키들을 이용한다.

[0094] J2EE 애플리케이션 서버

[0095] 주된 TOPAZ 처리는 J2EE 애플리케이션 서버의 제2스테이지에서 행해진다. TOPAZ 처리에는 집계 요청을 수신하고, 데이터베이스로부터 임의 필요한 정보를 검색하고, 이벤트에 대한 임의 필요한 요금부과 또는 로깅을 적용하고, 임의 조건 이벤트 처리를 적용하고 및 임의 가입된 애플리케이션에 전달되거나 아카이브될 이벤트를 큐잉하는 것이 포함된다.

[0096] 지속성

[0097] 데이터베이스와의 대화는 유사한 요청들이 단일 쿼리(query)로 집계될 수 있는 제3스테이지로 제한된다. 업데이트 또는 삽입은 함께 배치(batch)되어 성능을 개선시킨다.

[0098] 부하 밸런서

[0099] 부하 밸런서는 여러 개의 TOPAZ 클러스터 노드 중에서 클라이언트로부터 클러스터 멤버로의 HTTP 요청에 대한 부하 밸런싱을 허용한다. 세션 쿠키를 이용하여 동일 클라이언트 노드로부터의 HTTP 요청에 대한 노드 친화성을 제공한다. 그러나, 이들 세션의 부하는 클라이언트에 대한 가입에 종속하여, 시간이 지남에 따라 변할 수 있으므로, TOPAZ 노드마다 세션을 이동할 수 있는 것은 가치가 있다. 이것은 데이터베이스를 통해 행해지는데, 즉 세션이 이동될 때 세션의 현재 상태에 관한 정보가 레코드에 설정된다.

[0100] TSS 데이터 처리 및 애플리케이션 원격측정 집계

[0101] 일단 데이터가 원격측정 수신기에 의해 수신되면, 이 데이터는 초기 가입의 파라미터에 따라 처리되어야 한다. 필요할 경우, 본 발명은 조건식을 평가하고, 이 데이터는 애플리케이션으로의 송신을 위해 큐잉되거나 아카이브에 기입된다. 원격측정 수신기가 수백만의 클라이언트를 처리할 때, 원격측정 수신기는 대다수의 데이터 이벤트를 수신하기 때문에, 시스템 전체 성능에서 객체 할당의 효율이 중요한 요소가 된다. 객체 할당 메커니즘에 의해, 시스템의 정상 상태 실행 중에 대부분의 객체 할당 및 이로써 가비지 수집(garbage collection)을 피하게 된다.

[0102] 데이터 객체 할당

[0103] 다수의 원격측정 리포트를 서버 측 상의 버퍼에 기입할 경우, 라이브러리는 대형의 리포트 어레이를 할당하길 시도하여, 이는 결국에는 램(ram)을 이용함에 있어 대기 시간(latency)을 유발시킨다. 이를 방지하기 위해, 본 발명은 원격측정 리포트를 버퍼 전체에 원-샷으로(in one-shot) 기입하지 않는다, 그 대신, 본 발명은 각 리포트를 따로따로 작성하여 그것을 버퍼에 복사해 넣는다. 이와 같이 하여, 버퍼에 하나의 소규모 바이트 어레이를 할당하여 대기 시간을 줄인다. 본 발명은 또한 원격측정을 출력 스트림에 송출함으로써, 모든 헤더가 클라

이언트로부터 서버로 전송된 것을 확인한다. 청크된 전송(chunked transfers)의 지원은 또한 요청들을 송신하기 위한 대규모의 내부 버퍼의 할당을 디스에이블시킨다.

[0104] **애플리케이션 원격측정 집계**

[0105] 클라이언트 원격측정 집계는 데이터의 최종 목적지에 관계없이 다수의 클라이언트로부터의 원격측정을 집계한다. 일단 원격측정 수신기 내에서, 데이터는 데이터를 요청한 하나 이상의 애플리케이션에 라우팅되어야 한다. TSS와 같이, 애플리케이션들은 또한 원격측정 데이터 이벤트를 처리함에 있어 소정의 오버헤드를 가지므로, 애플리케이션들에게는 그들에게 청크로 원격측정을 제공하는 것이 도움이 된다. 원격측정 집계의 목적은 이벤트를 대규모 청크로 집계함으로써 각 이벤트마다의 계층 천이의 오버헤드를 줄이기 위한 것이다. 본 발명은 클라이언트 원격측정 집계기(175)를 이용하여 소규모 이벤트 메시지를 대규모 이벤트 그룹으로 집계한다. 따라서, 본 발명은 TOPAZ 서버 내에서 HTTP 구문분석의 오버헤드 및 서블릿 설정 호출(servlet setup calls)의 오버헤드를 줄인다. HTTP를 구문분석하고 서블릿 호출을 설정하는 비용은 집계된 그룹 내의 모든 이벤트들이 분담한다. 집계를 위한 두 가지 기준이 있는 데, 즉 집계된 그룹의 시기 및 크기이다. 대부분은, 이벤트들은 정해진 시간 윈도우 동안 집계되며, 이 윈도우는 집계된 이벤트 그룹이 서버에 전송되기 전에 도달하는 이벤트들이 원격측정 집계기 상에서 집계될 수 있는 기간을 지정한다. 이벤트들이 그 시간 기간 동안 거의 집계되지 않으면, 집계기는 최소 크기의 이벤트가 도달될 때까지 대기한 후, 집계된 이벤트 그룹을 전송한다.

[0106] **원격측정 아카이빙**

[0107] 원격측정 리포트는 즉시 전달 또는 아카이브될 수 있으며, 그 후 지정된 간격으로 전달될 수 있다. 이는 데이터의 규칙적인 수집은 필요로 하지만 실시간 응답을 위해 규칙적인 수집은 필요로 하지 않는 애플리케이션을 지원한다. 아카이브된 데이터는 대역폭 및 애플리케이션-서버 I/O 리소스의 보다 효율적인 이용을 위해 전달되도록 압축될 수 있다. 아카이브된 데이터의 경우, TOPAZ 데이터-집합에 대해 전체로서 동작하는 다수의 함수를 제공한다. 함수들은 min, max, sum, 및 count 뿐 아니라, average 및 median 등의 통계적인 함수들을 포함한다.

[0108] **사양**

[0109] 아카이빙 기간은 TOPAZ가 아카이브된 원격측정을 애플리케이션에 전달할 때의 규칙적인 시간을 지정한다. 포맷은 다음과 같다.

[0110] <아카이브-기간> → <매월> | <매주> | <매일> | <매시> | <매분>

[0111] <매월> → <num>M@<날짜> <시각>

[0112] <매주> → <num>W@<요일> <시각>

[0113] <매일> → <num>D@<시각>

[0114] <매시> → <num>H@<분>

[0115] <매분> → <num>m

[0116] <num> → 리포트들 간에서의 월/주/일/시/분의 수

[0117] <날짜> → 1..31 (주어진 날짜의 날짜를 갖지 않는 월에서는, 그 달의 마지막 날짜가 사용될 것임)

[0118] <요일> → 월 | 화 | 수 | 목 | 금 | 토 | 일

[0119] <시각> → hh:mm (서버의 현지 시간으로, 24시간 형태)

[0120] <분> → 0..59

[0121] 다음 예들은 이런 포맷팅을 강조한다.

[0122] 1M@15 00:00 매월 15일의 자정에 아카이브를 송신

[0123] 2W@Wen 08:00 2주마다의 수요일 오전 8시 30분에 아카이브를 송신

[0124] 1D@18:00 매일 오후 6시에 아카이브를 송신

[0125] 3H@30 3 시간마다 30분에 아카이브를 송신

[0126] 20m 매 20분 마다 아카이브를 송신

[0127] 아카이빙 함수들

[0128] archiveTelemetry... 메소드들은 호출자가 아카이브 함수들의 목록(list)을 제공할 수 있게 하고, 이들 각각은 그들이 데이터 리포트 집합에 적용되고 싶어한다. TOPAZ는 다음의 함수 집합을 상술한다. 플랫폼 연산자들은 다음 표에서 도시된 바와 같이 추가 함수들로 이 집합을 증가시킬 수 있다.

[0129] 함수 설명(x: 원격측정 데이터 항목)

[0130] max(x) 리포트에서 x의 최대값

[0131] min(x) 리포트에서 x의 최소값

[0132] median(x) 리포트에서 x의 중앙값

[0133] average(x) 리포트에서 x의 평균값(평균)

[0134] sum(x) 리포트에서 x의 총합

[0135] count(x) x의 리포트의 수

[0136] 아카이빙 함수들은 스칼라(scalar)에 대해서만 동작한다. pos 등의 다치화된 원격측정 데이터 항목들의 경우, 호출자는 도트 표기법을 이용하여 부분들(part)을 명명해야 한다. 예를 들어, "min(pos.lon); min(pos.lat); max(pos.lon); max(lat)"은 엔티티의 이동의 바운딩 박스(bounding box)를 계산한다.

[0137] 구현

[0138] 도 5는 서버측 상에서의 TSS 아카이빙을 도시한다. 즉, 각 TOPAZ 설비는 아카이빙 서비스 애플리케이션(servlet)을 갖는다. TSS 서비스는 아카이빙 가입을 수신할 때 이 서비스를 메시지로써 전달한다. 이 메시지는 아카이빙 함수들 및 리포트 기간을 포함한다. 이 서비스는 아카이브된 데이터의 전달을 위해 스케줄링된 작업을 시작한다(set up). TimerTask는 스레드 풀에서 한 스레드를 얻어, 그것에 ProcessTelemetryArchive Runnable을 제공하여, 그것을 개시한다. 각 TSS 수신기는 모든 TSS 수신기 노드들에 의해 공유되는 파일에 아카이브된 원격측정 데이터를 작성한다. 아카이브된 데이터 파일들은 애플리케이션 스트림 ID에 의해 명명된 디렉토리 내에 있다. 파일의 이름은 [clientstreamID].data이다. 파일의 내용은 일련의 <report> 객체들이다. 클라이언트 스트림 파일은 단지 하나의 TSS 수신기에 의해서만 액세스되어야 하지만(클라이언트의 데이터는 항상 동일 수신기에 전달되어야 하기 때문임), TSS 수신기는 동일 수신기 노드 상의 다수의 스레드로부터의 액세스인 경우에는 파일을 로킹(lock)할 것이다. 아카이브-작성 작업은 계측 프로세스와 동일한 기술을 이용한다. 서블릿은 데이터를 "채집" 스레드에 의해 공유되는 데이터 구조에 넣고, 이 스레드는 이 데이터를 아카이브 파일에 기입한다. 스케줄링된 아카이브-전달 프로세스는 아카이브 파일을 판독하고, 그것을 Java 객체들로 구문 분석하고, 아카이브 함수들을 계산하고, 그리고 그 결과들을 애플리케이션에 전송한다. 도 6은 프로토콜을 기술한 것이다.

[0139] 리포트 동기화

[0140] 애플리케이션들은 다수의 가입자로부터의 원격측정 요청이 소규모의 보다 빈번한 배치(batch) 보다는 대규모의 극소수의 배치로 전달되는 것을 요청할 수 있다. 이는 원격측정 리포트를 수신하는 애플리케이션 서버에서의 부하를 줄여준다. maximizeAggregation 파라미터를 통해, 애플리케이션들은 TSS가 그룹 가입을 위한 원격측정을 가능한 적은 송신 및 가능한 많은 송신으로 전달할 것을 요청할 수 있다. 이상적인 경우에, 이는 활동하는 모든 그룹 멤버들의 원격측정을 포함하는, 각 샘플링 간격으로의 단일 송신으로 이어질 것이다. 저-성능 장치들, 무선 네트워크들, 및 네트워크 게이트웨이들을 수반하는 실세계 시스템에서는, 이런 이상적인 경우는 거의 가능성이 없을 것이다. TSS는 이것에 대한 적합한 근사화를 시도한다.

[0141] 본 발명은 그룹 내의 모든 클라이언트에서의 원격측정 데이터의 샘플링 및 송신을 동기화함에 의해 접근한다. 데이터의 샘플링을 동기화하지 않으면-클라이언트로부터 수신된 데이터가 단순히 그 다음 송신 시간까지 버퍼링되어 있으면, 가장 이르게 버퍼링된 데이터는 그것이 애플리케이션에 도달하는 시간까지는 진부(stale)해질 것이다. TSS는 모든 클라이언트에 전송된 원격측정 가입에서의 syncTime 속성을 그것이 결정하는 UTC 시간으로 설정함으로써 원격측정 스트림들 동기화한다. 서버는 syncTime을 위해 현재 시간을 단순히 이용할 수 있고, 각

클라이언트에게 그것이 샘플링을 시작할 수 있는 미래 시간을 산출하기에 충분한 maxInterval 의 배수를 부가함으로써 개시 시간을 계산하게 할 수 있다.

[0142] 본 발명의 몇몇 예에 대해서만 기술 및 도시하였지만, 당업자라면 첨부된 청구범위 및 그 등가물에 기재된, 본 발명의 원리 및 사상을 벗어나지 않는 한 이들 실시예에 대한 변경이 이루어질 수 있음을 인식할 수 있을 것이다.

도면의 간단한 설명

[0143] 도 1은 본 발명의 원격측정 데이터 취득용 시스템을 나타내는 시스템 레벨도.

[0144] 도 2는 본 발명에 의한 조건식 처리를 나타내는 시스템 흐름도.

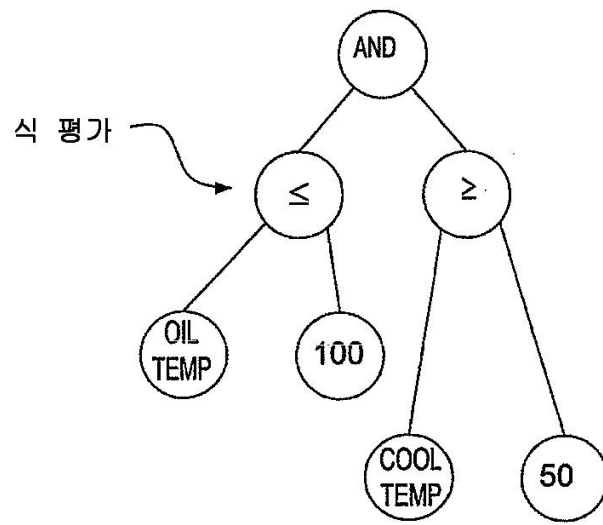
[0145] 도 3은 본 발명에 의한 조건식 처리를 나타내는 시스템 흐름도.

[0146] 도 4는 클라이언트 네트워크로부터 DMZ 네트워크를 통하여 애플리케이션 네트워크의 작업 흐름(workflow)을 설명하는 시스템 흐름도.

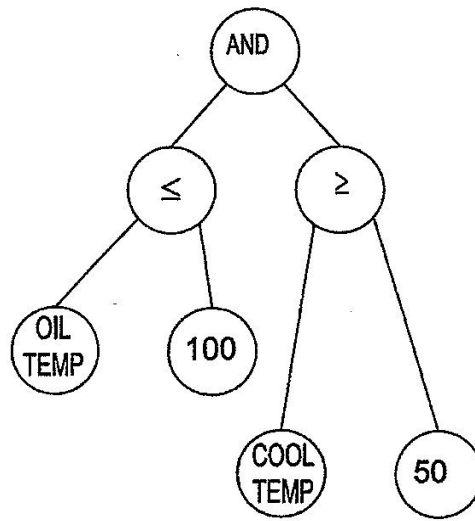
[0147] 도 5는 서버 측 상에서의 TSS 아카이빙(archiving)을 나타내는 시스템 흐름도.

[0148] 도 6은 시스템 프로토콜을 나타내는 도면.

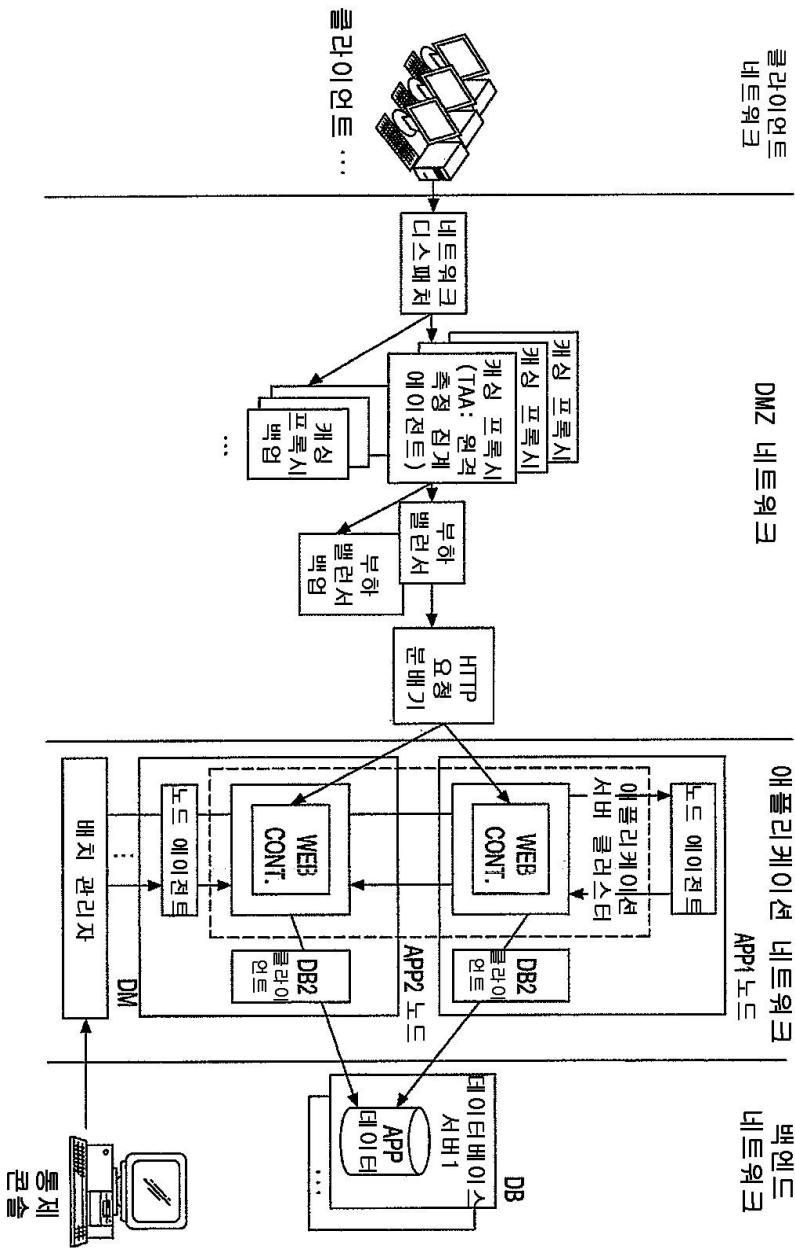
도면2



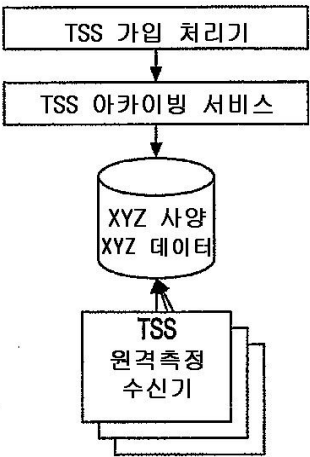
도면3



도면4



도면5



도면6

