



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0074962
(43) 공개일자 2018년07월04일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01) G06F 21/30 (2013.01)
(52) CPC특허분류
G06F 17/30082 (2013.01)
G06F 21/30 (2013.01)
(21) 출원번호 10-2016-0178777
(22) 출원일자 2016년12월26일
심사청구일자 2016년12월26일

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
허의남
경기도 용인시 기흥구 이현로29번길 86-21 201동401호(보정동,e편한세상대림아파트)
박준학
경기도 용인시 기흥구 덕영대로 1732 경희대학교 국제캠퍼스 전자정보대학 331호
(뒷면에 계속)
(74) 대리인
김홍석, 김동용

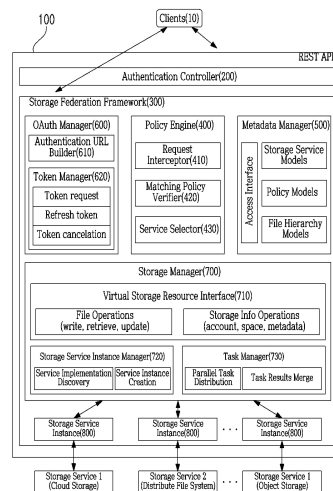
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 정책 기반의 스토리지 서비스 페더레이션 수행 방법 및 장치

(57) 요약

정책 기반 스토리지 서비스 연합 (policy based storage service federation) 수행 방법은, 클라이언트로부터 요청을 요청 인터셉터(Request Interceptor)가 수신하여, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별하는 정책 적용 가능 여부 식별 단계; 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하는 정책 획득 단계; 및 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 스토리지 서비스 집합 획득 단계를 포함하고, 빠른 어플리케이션 개발을 위한 스토리지 페더레이션 레이어 계층에 해당하는 프레임워크(architecture)를 제공할 수 있다.

대표도 - 도1



(72) 발명자

레네 헤인슨

경기도 용인시 기흥구 덕영대로 1732 경희대학교
국제캠퍼스 전자정보대학 331호

김윤곤

경기도 용인시 기흥구 덕영대로 1732 경희대학교
국제캠퍼스 전자정보대학 331호

이 발명을 지원한 국가연구개발사업

과제고유번호 1711035224

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 대학 ICT연구센터육성지원사업

연구과제명 실시간모바일클라우드서비스 플랫폼 연구개발

기 여 율 1/1

주관기관 경희대학교 산학협력단

연구기간 2016.01.01 ~ 2016.12.31

명세서

청구범위

청구항 1

정책 기반 스토리지 서비스 연합 (policy based storage service federation) 수행 방법에 있어서,
클라이언트로부터 요청을 요청 인터셉터(Request Interceptor)가 수신하여, 상기 요청에 대한 정책 적용 가능 (policy applicable) 여부를 식별하는 정책 적용 가능 여부 식별 단계;
상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하는 정책 획득 단계; 및
상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 스토리지 서비스 집합 획득 단계를 포함하는,
정책 기반 스토리지 서비스 연합 수행 방법.

청구항 2

제1항에 있어서,
상기 스토리지 서비스의 집합에 대한 서비스 자격 증명을 요청하는 서비스 자격 증명 요청 단계;
상기 서비스 자격 증명에 대한 인증을 수행하는 인증 수행 단계; 및
서비스 정보와 권한 인증을 상기 요청에 추가하고, 상기 요청을 스토리지 매니저로 전송하는 요청 송신 단계를 더 포함하는, 정책 기반 스토리지 서비스 연합 수행 방법.

청구항 3

제2항에 있어서,
가상 스토리지 자원 인터페이스가 상기 서비스 정보를 인스턴스 매니저로 전송하는 서비스 정보 전송 단계;
상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하는 서비스 구현 클래스 탐색 단계;
상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하고, 상기 서비스 구현 클래스 인스턴스를 상기 가상 스토리지 자원 인터페이스로 전송하는 서비스 구현 클래스 인스턴스 생성/전송 단계; 및
상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들을 상기 가상 스토리지 자원 인터페이스가 작업 매니저로 전송하는 요청/서비스 권한/서비스 구현 클래스 인스턴스 전송 단계를 더 포함하는, 정책 기반 스토리지 서비스 연합 수행 방법.

청구항 4

제3항에 있어서,
상기 작업 매니저가 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하는 작업 처리/응답 병합 단계; 및
상기 가상 스토리지 자원 인터페이스가 상기 작업 매니저로부터 수신한 응답을 상기 클라이언트가 수신할 수 있도록 최종 응답을 생성하는 최종 응답 생성 단계를 더 포함하는, 정책 기반 스토리지 서비스 연합 수행 방법.

청구항 5

제3항에 있어서,
상기 각각의 서비스에 대한 스토리지 서비스 프로파일은 가용성, 쓰기 성능, 읽기 성능, 가격, 및 사용한 공간에 해당하는 각각의 메트릭에 의해 정의되고, 그리고

상기 정책은 P1 내지 P4로 분류되고,

P1은 주로 수정 및 읽기/쓰기 작업을 수행하는 경우에 해당하고,

P2는 쓰기 작업은 거의 수행하지 않고 주로 읽는 작업만 수행하는 경우에 해당하고,

P3은 쓰기와 읽기 작업을 거의 하지 않는 경우에 해당하고,

P4는 중요도가 떨어지는 데이터. 사용 가능성이 거의 없지만 단지 보관을 위해 사용하는 경우에 해당하는 것을 특징으로 하는, 정책 기반 스토리지 서비스 연함 수행 방법.

청구항 6

제5항에 있어서,

상기 서비스 프로파일은 S1 내지 S4를 포함하고,

상기 S1 내지 S4 각각에 대하여, 상기 서비스 프로파일 별로 정규화된 가중치가 부여되고,

상기 P1 내지 P4 각각에 대하여, 상기 각각의 메트릭 별로 우선순위가 결정되고,

상기 정규화된 가중치와 상기 우선순위에 기반하여 상기 P1 내지 P4 각각에 대하여 상기 S1 내지 S4를 선택하는 순서를 결정하는, 정책 기반 스토리지 서비스 연함 수행 방법.

청구항 7

정책 기반 스토리지 서비스 연함을 수행하는 장치에 있어서,

클라이언트로부터 요청을 수신하고, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별하고, 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하고, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 요청 인터셉터(Request Interceptor); 및

상기 요청 인터셉터로부터의 정책 요청을 수신하고, 액티브 정책을 획득하고, 상기 요청 정보와 상기 정책 상태에 따라 일치하는 정책을 상기 요청 인터셉터로 반환하는 매칭 정책 검증자를 포함하는, 정책 기반 스토리지 서비스 연함을 수행 장치.

청구항 8

제7항에 있어서,

상기 정책 요청에 따라 상기 액티브 정책을 획득하고, 상기 액티브 정책을 상기 매칭 정책 검증자로 전송하는 메타데이터 매니저; 및

상기 정책을 기반으로 서비스 프로파일을 획득하고, 상기 서비스 프로파일에 기반하여 맞춤형 스토리지 서비스의 집합을 생성하는 서비스 선택자를 더 포함하는, 정책 기반 스토리지 서비스 연함을 수행 장치.

청구항 9

제7항에 있어서,

상기 요청 인터셉터는 상기 스토리지 서비스의 집합에 대한 서비스 자격 증명을 요청하고, 상기 서비스 자격 증명에 대한 인증을 수행하고, 서비스 정보와 권한 인증을 상기 요청에 추가하고, 그리고

상기 서비스 자격 증명에 대한 요청을 상기 요청 인터셉터로부터 수신하고, 상기 인증을 위한 토큰을 상기 요청 인터셉터로 제공하는 상기 인증 매니저(OAuth manager)를 더 포함하는, 정책 기반 스토리지 서비스 연함을 수행 장치.

청구항 10

제9항에 있어서,

상기 서비스 정보와 권한 인증이 추가된 상기 요청을 상기 요청 인터셉터로부터 수신하는 가상 스토리지 자원 인터페이스;

상기 가상 스토리지 자원 인터페이스로부터 수신된 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하고, 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하고, 상기 서비스 구현 클래스 인스턴스를 상기 가상 스토리지 자원 인터페이스로 전송하는 인스턴스 매니저;

상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들을 상기 가상 스토리지 자원 인터페이스로부터 수신하고, 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하고, 상기 응답을 상기 가상 스토리지 자원 인터페이스로 전송하는 작업 매니저를 포함하고,

상기 가상 스토리지 자원 인터페이스는, 상기 작업 매니저로부터 수신한 응답을 상기 클라이언트가 수신할 수 있도록 최종 응답을 생성하는, 정책 기반 스토리지 서비스 연합을 수행 장치.

청구항 11

제10항에 있어서,

상기 인스턴스 매니저와 상기 작업 매니저로부터의 상기 요청을 복수의 스토리지 서비스들의 각각을 수행하는 복수의 스토리지 장치들로 전송하는 복수의 스토리지 서비스 인터페이스를 더 포함하고,

상기 복수의 스토리지 장치들은 클라우드 스토리지, 분산 파일 시스템, 및 오브젝트 스토리지를 포함하는, 정책 기반 스토리지 서비스 연합을 수행 장치.

청구항 12

정책 기반 스토리지 서비스 연합을 수행하는 장치에 있어서,

정책 기반 스토리지 서비스 연합을 위한 요청을 전송하는 클라이언트; 및

상기 클라이언트로부터 상기 요청을 수신하고, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별하고, 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하고, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 정책 엔진을 포함하는, 정책 기반 스토리지 서비스 연합을 수행하는 장치.

청구항 13

제12항에 있어서,

상기 서비스 정보와 권한 인증이 추가된 상기 요청을 상기 정책 엔진으로부터 수신하고, 상기 수신된 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하고, 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하는 스토리지 서비스 매니저를 더 포함하는, 정책 기반 스토리지 서비스 연합을 수행하는 장치.

청구항 14

제13항에 있어서,

상기 스토리지 서비스 매니저는 상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들에 기반하여, 상기 작업 매니저가 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하고, 상기 병합된 응답을 상기 클라이언트로 전송하는, 정책 기반 스토리지 서비스 연합을 수행하는 장치.

발명의 설명

기술 분야

[0001] 본 발명은 정책 기반의 스토리지 서비스 페더레이션 수행 방법 및 장치에 관한 것이다. 보다 상세하게는, 정책 기반의 스토리지 서비스 페더레이션 프레임워크를 이용하여 정책 기반의 스토리지 서비스 페더레이션을 수행하는 방법 및 장치에 관한 것이다.

배경 기술

- [0002] IT 산업은 다양한 형식을 가지고 크기가 상당한 데이터들이 급격히 발생하는데 비해 이를 처리하는 속도는 상대적으로 저조하여 이를 저장 및 관리 방법이 필요한 상황이다.
- [0003] 이와 같은 데이터의 저장 및 관리에서, 엄청난 규모의 데이터가 발생하게 됨에 따라 데이터 분석 분야에 이점이 생김과 동시에 데이터의 보관 등의 문제 또한 발생하게 되었다. 이러한 빅 데이터는 크기가 상당하여 한 저장소에 모두 담기가 어렵다. 뿐만 아니라 빅 데이터 집합들(big datasets) 과 데이터 저장소는 데이터 액세스 성능을 감소시킨다는 문제점이 있다. 따라서 기하급수적으로 늘어나고 있는 데이터 생성 속도에 발맞추어 확장 가능성을 제공하면서 데이터가 저장되는 방법과 위치에 대한 자동적이고 정확한 결정을 할 수 있는 시스템이 필요하다.
- [0004] 이와 관련하여, Hewlett Packard Enterprise 3PAR Peer Motion software는, 여러 third party Store Server 시스템의 연합 기능을 허용하고 있다. 하지만, 가장 큰 문제점은 연합 기능이 타 시스템과의 호환이 불가능하는 점이다.
- [0005] 다음으로, 여러 스토리지 페더레이션 방법에 대한 연구가 있다. 하지만 대부분의 경우 벤더 종속을 방지하고 여러 계정의 관리를 단순화하기 위한 클라우드 저장소의 연합에 초점을 맞추고 있다. 이러한 접근 방법의 문제점은 오직 하나의 서비스 종류에만(e.g. 클라우드 저장소) 집중을 하고 있는 것이다.
- [0006] 마지막으로, 서비스 선택 접근에 대한 논의와 관련하여, 현재 존재하는 방법들은 정적으로 정해놓은 수치의 서비스만을 제공하는 한계점이 존재한다는 문제점이 있다. 또한, 대부분의 스토리지 페더레이션은 여전히 정책 기반의 서비스 선택이 부족하다는 문제점이 있다.

발명의 내용

해결하려는 과제

- [0007] 따라서, 본 발명에서 해결하고자 하는 과제는 빠른 어플리케이션 개발을 위한 스토리지 페더레이션 레이어 계층에 해당하는 프레임워크(framework)를 제공하는 데에 그 목적이 있다.
- [0008] 또한, 본 발명에서 해결하고자 하는 과제는 타 시스템과의 호환이 가능하면서, 정책 기반의 서비스 선택이 가능한 정책 기반 스토리지 서비스 프레임워크(framework)를 제공하는 데에 그 목적이 있다.

과제의 해결 수단

- [0009] 상기와 같은 과제를 해결하기 위한 본 발명에 따른 정책 기반 스토리지 서비스 연합 (policy based storage service federation) 수행 방법은, 클라이언트로부터 요청을 요청 인터셉터(Request Interceptor)가 수신하여, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별하는 정책 적용 가능 여부 식별 단계를 포함한다. 또한, 상기 정책 기반 스토리지 서비스 연합 수행 방법은, 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하는 정책 획득 단계를 더 포함한다. 또한, 상기 정책 기반 스토리지 서비스 연합 수행 방법은, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 스토리지 서비스 집합 획득 단계를 포함한다.
- [0010] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 방법은, 상기 스토리지 서비스의 집합에 대한 서비스 자격 증명을 요청하는 서비스 자격 증명 요청 단계; 상기 서비스 자격 증명에 대한 인증을 수행하는 인증 수행 단계; 및 서비스 정보와 권한 인증을 상기 요청에 추가하고, 상기 요청을 스토리지 매니저로 전송하는 요청 송신 단계를 더 포함할 수 있다.
- [0011] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 방법은, 가상 스토리지 자원 인터페이스가 상기 서비스 정보를 인스턴스 매니저로 전송하는 서비스 정보 전송 단계; 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하는 서비스 구현 클래스 탐색 단계; 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하고, 상기 서비스 구현 클래스 인스턴스를 상기 가상 스토리지 자원 인터페이스로 전송하는 서비스 구현 클래스 인스턴스 생성/전송 단계를 더 포함할 수 있다. 또한, 상기 정책 기반 스토리지 서비스 연합 수행 방법은, 상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들을 상기 가상 스토리지 자원 인터페이스가 작업 매니저로 전송하는 요청/서비스 권한/서비스 구현 클래스 인스턴스 전송 단계를 더 포함할 수 있다.
- [0012] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 방법은, 상기 작업 매니저가 비동기적으로 집

근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하는 작업 처리/응답 병합 단계; 및 상기 가상 스토리지 자원 인터페이스가 상기 작업 매니저로부터 수신한 응답을 상기 클라이언트가 수신할 수 있도록 최종 응답을 생성하는 최종 응답 생성 단계를 더 포함할 수 있다.

- [0013] 일 실시예에 따르면, 상기 각각의 서비스에 대한 스토리지 서비스 프로파일은 가용성, 쓰기 성능, 읽기 성능, 가격, 및 사용한 공간에 해당하는 각각의 메트릭에 의해 정의될 수 있다. 또한, 상기 정책은 P1 내지 P4로 분류되고, P1은 주로 수정 및 읽기/쓰기 작업을 수행하는 경우에 해당하고, P2는 쓰기 작업은 거의 수행하지 않고 주로 읽는 작업만 수행하는 경우에 해당할 수 있다. 또한, P3은 쓰기와 읽기 작업을 거의 하지 않는 경우에 해당하고, P4는 중요도가 떨어지는 데이터, 사용 가능성이 거의 없지만 단지 보관을 위해 사용하는 경우에 해당하는 것을 특징으로 할 수 있다.
- [0014] 일 실시예에 따르면, 상기 서비스 프로파일은 S1 내지 S4를 포함하고, 상기 S1 내지 S4 각각에 대하여, 상기 서비스 프로파일 별로 정규화된 가중치가 부여될 수 있다. 또한, 상기 P1 내지 P4 각각에 대하여, 상기 각각의 메트릭 별로 우선순위가 결정되고, 상기 정규화된 가중치와 상기 우선순위에 기반하여 상기 P1 내지 P4 각각에 대하여 상기 S1 내지 S4를 선택하는 순서를 결정될 수 있다.
- [0015] 본 발명의 다른 양상에 따른, 정책 기반 스토리지 서비스 연합을 수행하는 장치는, 클라이언트로부터 요청을 수신하고, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별하고, 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하고, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 요청 인터셉터(Request Interceptor)를 포함한다. 또한, 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 요청 인터셉터로부터의 정책 요청을 수신하고, 액티브 정책을 획득하고, 상기 요청 정보와 상기 정책 상태에 따라 일치하는 정책을 상기 요청 인터셉터로 반환하는 매칭 정책 검증자를 더 포함한다.
- [0016] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 정책 요청에 따라 상기 액티브 정책을 획득하고, 상기 액티브 정책을 상기 매칭 정책 검증자로 전송하는 메타데이터 매니저; 및 상기 정책을 기반으로 서비스 프로파일을 획득하고, 상기 서비스 프로파일에 기반하여 맞춤형 스토리지 서비스의 집합을 생성하는 서비스 선택자를 더 포함할 수 있다.
- [0017] 일 실시예에 따르면, 상기 요청 인터셉터는 상기 스토리지 서비스의 집합에 대한 서비스 자격 증명을 요청하고, 상기 서비스 자격 증명에 대한 인증을 수행하고, 서비스 정보와 권한 인증을 상기 요청에 추가할 수 있다. 또한, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 서비스 자격 증명에 대한 요청을 상기 요청 인터셉터로부터 수신하고, 상기 인증을 위한 토큰을 상기 요청 인터셉터로 제공하는 상기 인증 매니저(OAuth manager)를 더 포함할 수 있다.
- [0018] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 서비스 정보와 권한 인증이 추가된 상기 요청을 상기 요청 인터셉터로부터 수신하는 가상 스토리지 자원 인터페이스를 더 포함할 수 있다. 또한, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 가상 스토리지 자원 인터페이스로부터 수신된 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하고, 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하고, 상기 서비스 구현 클래스 인스턴스를 상기 가상 스토리지 자원 인터페이스로 전송하는 인스턴스 매니저를 더 포함할 수 있다. 또한, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들을 상기 가상 스토리지 자원 인터페이스로부터 수신하고, 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하고, 상기 응답을 상기 가상 스토리지 자원 인터페이스로 전송하는 작업 매니저를 더 포함할 수 있다. 한편, 상기 가상 스토리지 자원 인터페이스는, 상기 작업 매니저로부터 수신한 응답을 상기 클라이언트가 수신할 수 있도록 최종 응답을 생성할 수 있다.
- [0019] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 인스턴스 매니저와 상기 작업 매니저로부터의 상기 요청을 복수의 스토리지 서비스들의 각각을 수행하는 복수의 스토리지 장치들로 전송하는 복수의 스토리지 서비스 인터페이스를 더 포함할 수 있다. 이때, 상기 복수의 스토리지 장치들은 클라우드 스토리지, 분산 파일 시스템, 및 오브젝트 스토리지를 포함할 수 있다.
- [0020] 본 발명의 또 다른 양상에 따른, 정책 기반 스토리지 서비스 연합을 수행하는 장치는, 정책 기반 스토리지 서비스 연합을 위한 요청을 전송하는 클라이언트; 및 상기 클라이언트로부터 상기 요청을 수신하고, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별하고, 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하고, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득하는 정책 엔진을 포함할 수 있

다.

[0021] 일 실시예에 따르면, 상기 정책 기반 스토리지 서비스 연합 수행 장치는, 상기 서비스 정보와 권한 인증이 추가된 상기 요청을 상기 정책 엔진으로부터 수신하고, 상기 수신된 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하고, 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하는 스토리지 서비스 매니저를 더 포함할 수 있다.

[0022] 일 실시예에 따르면, 상기 스토리지 서비스 매니저는 상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들에 기반하여, 상기 작업 매니저가 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하고, 상기 병합된 응답을 상기 클라이언트로 전송할 수 있다.

발명의 효과

[0023] 본 발명에 따른 정책 기반 스토리지 서비스 연합 수행 방법은, 수평 및 수직적 스케일링과 병렬처리 기술을 통해 데이터 처리량 및 성능을 향상시킬 수 있어, 빠른 어플리케이션 개발을 위한 스토리지 페더레이션 레이어 계층에 해당하는 프레임워크(framework)를 제공할 수 있다.

[0024] 또한, 본 발명에 따른 정책 기반 스토리지 서비스 연합 수행 방법은, 서비스 프로파일을 통한 정책 기반의 스토리지 서비스에 의해, 타 시스템과의 호환이 가능하면서, 정책 기반의 서비스 선택이 가능한 정책 기반 스토리지 서비스 프레임워크(framework)를 제공할 수 있다는 장점이 있다.

도면의 간단한 설명

[0025] 도 1은 본 발명에 따른 정책 기반 스토리지 서비스 연합을 수행하는 장치의 세부 구성도를 도시한다.

도 2는 본 발명에 따른 정책 기반 스토리지 서비스 연합을 수행 방법의 흐름도이다.

도 3은 본 발명의 예시에 따른 스토리지 서비스 프로파일을 도시한다.

도 4는 본 발명의 예시에 따른 정규화된 스토리지 서비스 프로파일을 도시한다.

도 5는 본 발명에 따른 정규화를 거친 서비스 프로파일과 정규화를 거치지 않은 서비스 프로파일을 비교한 도면이다.

도 6은 본 발명의 예시에 따른 정책에 대하여 각각의 메트릭들의 순위를 정의한 도면이다.

도 7은 본 발명의 예시에 따른 정책에 대하여 각각의 메트릭들의 비율을 막대 그래프로 도시한 도면이다.

도 8은 본 발명의 예시에 따른 정책 기반 서비스 거리를 방사형으로 도시한 도면이다.

도 9a 내지 도 9c는 본 발명의 예시에 따른 쓰기 작업 수행, 읽기 작업 수행, 및 삭제 작업 수행에서, 파일 크기와 블록 크기에 따른 스루풋(throughput)을 도시한다.

도 10은 본 발명의 예시에 따른 정책 기반 스토리지 서비스 선택을 위한 알고리즘의 구성을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0026] 상술한 본 발명의 특징 및 효과는 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이며, 그에 따라 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을 용이하게 실시할 수 있을 것이다. 본 발명은 다양한 변경을 가할 수 있고 여러 가지 형태를 가질 수 있는바, 특정 실시 예들을 도면에 예시하고 본문에 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 개시형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 본 명세서에서 사용한 용어는 단지 특정한 실시 예들을 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다.

[0027] 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용한다.

[0028] 제1, 제2등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.

- [0029] 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는"이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0030] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미가 있다.
- [0031] 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않아야 한다.
- [0032] 이하의 설명에서 사용되는 구성요소에 대한 접미사 "모듈", "블록" 및 "부"는 명세서 작성의 용이함만이 고려되어 부여되거나 혼용되는 것으로서, 그 자체로 서로 구별되는 의미 또는 역할을 갖는 것은 아니다.
- [0033] 이하, 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 당해 분야에 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 설명한다. 하기에서 본 발명의 실시 예를 설명함에 있어, 관련된 공지의 기능 또는 공지의 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략한다.
- [0034] 이하, 본 발명에 따른 정책 기반 스토리지 서비스 연합을 수행하는 장치와 정책 기반 스토리지 서비스 연합을 수행하는 방법에 대해 살펴보기로 한다. 이와 관련하여, 본 발명에서 제시되는 프레임워크(framework)의 목적은 빠른 어플리케이션 개발을 위한 스토리지 페더레이션 레이어 계층을 제공하는 것이다. 본 프레임워크의 주요 목적은 충분한 양의 요구한 자원을 공급하면서 빠르게 어플리케이션을 개발할 수 있도록 페더레이션 스토리지 계층을 제공하는 것이다. 이것은 스토리지 서비스 인터페이스를 구현하는 것을 통해 모든 파일 기반의 스토리지 서비스에서 동작할 수 있다. 이것은 클라이언트에게 제한된 숫자의 구현 서비스를 제공하는 것 대신에 원하는 서비스를 본 프레임워크에 통합시킬 수 있도록 허용한다.
- [0035] 이와 관련하여, 도 1은 본 발명에 따른 정책 기반 스토리지 서비스 연합을 수행하는 장치의 세부 구성도를 도시한다. 도 1에 도시된 바와 같이, 스토리지 서비스 연합 수행 장치(1000)는 클라이언트(10), REST API(100), 인증 컨트롤러(Authentication Controller, 200), 스토리지 페더레이션 프레임워크(Storage Federation Framework, 300)를 포함한다. 또한, 상기 스토리지 페더레이션 프레임워크(300)은 정책 엔진(Policy Engine, 400), 메타데이터 매니저(Metadata manager, 500), 인증 매니저(OAuth manager, 600), 스토리지 매니저(Storage Manager, 700) 및 스토리지 서비스 인터페이스(800)를 포함한다.
- [0036] 상기 클라이언트(10)는 정책 기반 스토리지 서비스 연합을 위한 요청을 상기 REST API(100)를 통해 상기 스토리지 페더레이션 프레임워크(300)로 제공한다. 이때, 상기 요청은 인증 컨트롤러(200)를 통해 상기 스토리지 페더레이션 프레임워크(300)로 제공될 수 있다.
- [0037] 상기 REST API(100)는 상기 클라이언트(10)로부터 상기 스토리지 서비스 연합을 위한 요청을 수신하고, 상기 프레임워크(300)를 통해 스토리지 서비스 연합을 수행한다. 즉, 상기 REST API(100)는 프레임워크를 독립형 서비스 솔루션으로 변환하는 역할을 수행한다. 또한, 이는 아키텍처와 도메인 모델로 확장할 수 있다.
- [0038] 상기 인증 컨트롤러(200) 컴포넌트는 토큰 기반의 인증 절차를 제공하기 위해 통합되어 있다. 이 추가 구성 요소는 클라이언트 등록, 인증 및 API 엔드 포인트에 액세스하기 위한 인증 토큰을 제공한다. 한편 도메인 모델은 클라이언트 자신들의 서비스 인증, 정책 그리고 관련 파일을 정의할 수 있도록 확장된다.
- [0039] 상기 정책 엔진(400)은 상기 클라이언트(10)로부터 상기 요청을 수신하고, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별한다. 또한, 상기 정책 엔진(400)은 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하고, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득한다.
- [0040] 보다 구체적으로, 상기 정책 엔진(400)은 상기 프레임워크(300)의 두뇌 역할을 수행한다. 의사결정, 서비스 선택, 검증 요청 및 요청에 대한 보안을 담당한다. 자동화된 결정을 하기 위해서 정책 기반의 데이터 분산 알고리즘이 구현된다 두 가지의 정책 모델이 구현되었다. 첫 번째 모델은 서비스 프로파일 측정 항목의 중요도를 정의한다. 두 번째 모델은 주어진 조건의 복제본의 수를 정의한다. 이 모델은 클라이언트가 별개로 데이터 복제본의 수를 정의할 수 있도록 허용한다. 상기 정책 엔진(400)은 요청 인터셉터(Request Interceptor, 410), 매칭 정책 검증자(Matching Policy Verifier, 40), 및 서비스 선택자(Service Selector, 430)를 포함한다.
- [0041] 상기 요청 인터셉터(Request Interceptor, 410)는 상기 클라이언트(10)로부터 요청을 수신하고, 상기 요청에 대

한 정책 적용 가능 여부를 식별하고, 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득하고, 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득한다.

- [0042] 또한, 상기 요청 인터셉터(Request Interceptor, 410)는 상기 스토리지 서비스의 집합에 대한 서비스 자격 증명을 요청하고, 상기 서비스 자격 증명에 대한 인증을 수행하고, 서비스 정보와 권한 인증을 상기 요청에 추가할 수 있다.
- [0043] 즉, 상기 스토리지 매니저(700)에게 전송하기 전에 각 요청을 검증 및 보안을 담당한다. 요청을 보완하기 위해, 서비스 자격 증명에 반드시 포함되어야 한다. 그러므로 모든 요청은 상기 요청 인터셉터(410)에게 검증을 받아야 한다.
- [0044] 상기 매칭 정책 검증자(420)는 상기 요청 인터셉터(420)로부터의 정책 요청을 수신하고, 액티브 정책을 획득하고, 상기 요청 정보와 상기 정책 상태에 따라 일치하는 정책을 상기 요청 인터셉터(420)로 반환한다. 즉, 상기 매칭 정책 검증자(420)는 현재 파일과 모든 정책 조건이 일치하는지 확인하는 역할을 담당한다.
- [0045] 상기 서비스 선택자(Service Selector, 430)는 상기 정책을 기반으로 서비스 프로파일을 획득하고, 상기 서비스 프로파일에 기반하여 맞춤형 스토리지 서비스의 집합을 생성한다. 즉, 상기 서비스 선택자(430)는 정책과 복제 요인들의 집합에 따라 가장 좋은 서비스 그룹을 선택하는 역할을 담당한다.
- [0046] 상기 메타데이터 매니저(500)는 상기 정책 요청에 따라 상기 액티브 정책을 획득하고, 상기 액티브 정책을 상기 매칭 정책 검증자(420)로 전송한다. 보다 구체적으로, 상기 메타데이터 매니저(500)는 스토리지 서비스, 정책 및 파일의 계층 도메인 모델을 정의하는 역할을 수행한다. 또한, 다른 구성요소를 검색, 생성 그리고 모델의 정보를 수정할 수 있도록 하는 인터페이스를 제공한다. 상기 메타데이터 매니저(500)는 마이크로소프트 엔티티 프레임워크(Microsoft Entity Framework)의 코드 우선(code-first) 접근 방식을 사용할 수 있다. 이는 어떤 데이터베이스 설계 대신에 도메인 클래스를 생성하는데 초점을 맞춘 도메인 기반 설계 기술이다. 엔티티 프레임워크(Entity Framework)를 사용하는 것으로 새 도메인 클래스를 만들거나 기존의 것을 덮어쓰기를 수행하여 쉽게 도메인 모델의 확장이 가능하다.
- [0047] 상기 인증 매니저(600)는 상기 서비스 자격 증명에 대한 요청을 상기 요청 인터셉터(410)로부터 수신하고, 상기 인증을 위한 토큰을 상기 요청 인터셉터(410)로 제공한다. 보다 구체적으로, 상기 인증 매니저(600)는 OAuth 2.0 인증을 사용할 수 있도록 구현된다. 보통 클라우드 스토리지를 대상으로 하며, 두 개의 세부 컴포넌트로 구성되어 있다. 즉, 상기 메타데이터 매니저(600)는 인증 URL 생성자(Authentication Url Builder, 610) 및 토큰 매니저(Token Manager, 620)를 포함한다.
- [0048] 상기 인증 URL 생성자(610)는 주어진 서비스의 프로파일을 기반으로 하여 인증 URL과 redirect URL을 구성하는 역할을 수행한다. 또한, 상기 인증 URL 생성자(610)는 상기 구성된 URL이 유효한 것인지 확인한다. 상기 토큰 매니저(620)는 상기 토큰의 요청, 갱신 및 서비스 액세스 토큰을 취소하는 역할을 수행한다.
- [0049] 상기 스토리지 매니저(700)는 상기 서비스 정보와 권한 인증이 추가된 상기 요청을 상기 정책 엔진(400)으로부터 수신한다. 또한, 상기 스토리지 매니저(700)는 상기 수신된 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하고, 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성한다.
- [0050] 한편, 상기 스토리지 매니저(700)는 상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들에 기반하여, 작업 매니저(730)가 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리한다. 또한, 상기 스토리지 매니저(700)는 상기 서비스들의 응답을 병합하고, 상기 병합된 응답을 상기 클라이언트(10)로 전송한다.
- [0051] 즉, 상기 스토리지 매니저(700)는 중앙 저장소의 자원을 관리하는 컴포넌트이다. 상기 스토리지 매니저(700)는 단일 지점으로부터 복수의 스토리지 서비스 자원으로 접근을 할 수 있는 인터페이스를 제공한다. 또한, 실행 시간을 향상시키고 동시에 다수의 서비스에 접근할 수 있도록 병렬 처리 기술이 적용되어 구현되었다. 상기 스토리지 매니저(700)는 동적으로 구현 클래스를 생성하고 액세스 서비스를 제공하기 위해 반사 지향 프로그래밍을 사용한다. 상기 스토리지 매니저(700)는 아래와 같이 세 가지의 세부 컴포넌트로 나뉘어진다.
- [0052] 즉, 상기 스토리지 매니저(700)는 가상 스토리지 자원 인터페이스(Virtual Storage Resource Interface, 710), (스토리지 서비스) 인스턴스 매니저((Storage Service) Instance Manager, 720) 및 작업 매니저(Task Manager, 730)를 포함한다.

- [0053] 상기 가상 스토리지 자원 인터페이스(710)는 상기 서비스 정보와 권한 인증이 추가된 상기 요청을 상기 요청 인터셉터(410)로부터 수신한다. 즉, 상기 가상 스토리지 자원 인터페이스(710)는 원활한 서비스가 제공되고 있는 스토리지에 있는 모든 별개 스토리지를 변환하여 단일 액세스 지점을 제공한다.
- [0054] 상기 인스턴스 매니저(720)는 상기 가상 스토리지 자원 인터페이스(710)로부터 수신된 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색하고, 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성한다. 또한, 상기 인스턴스 매니저(720)는 상기 서비스 구현 클래스 인스턴스를 상기 가상 스토리지 자원 인터페이스(710)로 전송한다. 즉, 상기 인스턴스 매니저(720)는 서비스 구현 클래스의 발견 및 인스턴스를 만드는 역할을 수행한다. 상기 인스턴스 매니저(720)는 서비스 도메인 모델과 스토리지 서비스 인터페이스 상속에 정의된 고유한 이름에 따라서 각 서비스가 구현된 클래스를 반사 지향 프로그래밍을 통해 찾는다.
- [0055] 상기 작업 매니저(730)는 상기 요청, 상기 권한 인증과 연관된 서비스 권한 및 상기 서비스 구현 클래스 인스턴스들을 상기 가상 스토리지 자원 인터페이스(710)로부터 수신한다. 또한, 상기 작업 매니저(730)는 비동기적으로 접근하는 복수의 서비스들에 대한 작업을 처리하고, 상기 서비스들의 응답을 병합하고, 상기 응답을 상기 가상 스토리지 자원 인터페이스(710)로 전송한다. 이때, 상기 가상 스토리지 자원 인터페이스(710)는, 상기 작업 매니저(730)로부터 수신한 응답을 상기 클라이언트(10)가 수신할 수 있도록 최종 응답을 생성한다. 즉, 상기 작업 매니저(730)는 병렬적으로 접근하는 다수의 서비스 및 응답을 관리한다.
- [0056] 상기 스토리지 서비스 인터페이스(800)는 상기 인스턴스 매니저(720)와 상기 작업 매니저(730)로부터의 상기 요청을 복수의 스토리지 서비스들의 각각을 수행하는 복수의 스토리지 장치들(800)로 전송한다. 이때, 상기 복수의 스토리지 장치들(800)은 도시된 바와 같이, 클라우드 스토리지, 분산 파일 시스템, 및 오브젝트 스토리지를 포함할 수 있다. 즉, 상기 스토리지 서비스 인터페이스(800)는 각 스토리지 서비스가 본 프레임워크에 액세스하는 것을 가능하게 구현되어야 한다. 또한, 인터페이스의 구현은 동적으로 발견하고 반사 지향 프로그래밍을 사용하여 서비스 클래스의 인스턴스를 사용 가능하게 해야 한다.
- [0057] 이상에서는 정책 기반 스토리지 서비스 연합을 수행하는 장치에 대해 살펴보았다. 다음에서는, 정책 기반 스토리지 서비스 연합을 수행하는 방법에 대해 살펴보기로 한다. 이와 관련하여, 도 2는 본 발명에 따른 정책 기반 스토리지 서비스 연합을 수행 방법의 흐름도이다.
- [0058] 도 2에 도시된 바와 같이, 상기 정책 기반 스토리지 서비스 연합을 수행 방법은 i. 정책 엔진에 (Policy Engine)의해 수행되는 서비스 연합 과정 및 ii. 스토리지 매니저(Storage Manager)에 의한 서비스 연합 과정을 포함한다.
- [0059] i. 정책 엔진 식별 및 요청의 보완(Policy Engine verifies and complements the request)
- [0060] A. 정책 적용 가능 여부 식별 단계: 클라이언트로부터 요청을 요청 인터셉터(Request Interceptor)가 수신하여, 상기 요청에 대한 정책 적용 가능(policy applicable) 여부를 식별한다. 즉, 요청 인터셉터가 이 요청이 적용 가능한지 (policy-applicable) 를 식별한다. 이때 적용이 가능(policy-applicable)한 경우, 예를 들어 파일을 작성하거나 복제된 파일을 검색할 때에는 B 단계로 간다. 반대로, 적용이 불가능한 경우, 모든 서비스 혹은 특정 서비스로 요청이 전송된다. 예를 들면 복제되지 않은 파일을 읽거나 사용 가능한 공간을 제공받은 경우에는 D 단계로 간다.
- [0061] B. 정책 획득 단계: 상기 요청에 대한 요청 정보와 정책 상태를 평가하여 정책을 획득한다. 즉, 정책과의 연결은 정책 검증 연결자(Matching Policy Verifier)를 통해 이루어진다. 이는 메타데이터 매니저(Metadata Manager)에게서 Active 정책을 받아오고, 컴파일 및 요청 정보와 정책 상태를 평가하고 일치하는 정책이 있는지 반환한다.
- [0062] C. 스토리지 서비스 집합 획득 단계: 상기 정책을 기반으로 맞춤형 스토리지 서비스의 집합을 획득한다. 즉, 정책을 기반으로 하는 맞춤형 스토리지 서비스의 집합은 서비스 선택자(Service Selector)를 통해 얻는다. 이는 스토리지 서비스 프로파일(profile)을 메타데이터 매니저(Metadata Manager)로부터 가져온다. 그 후 요청한 절차에 대한 가장 적절한 서비스를 찾기 위해 정책 기반 알고리즘을 적용한다.
- [0063] 이와 관련하여, 각각의 서비스에 대한 스토리지 서비스 프로파일은 가용성, 쓰기 성능, 읽기 성능, 가격, 및 사용한 공간에 해당하는 각각의 메트릭에 의해 정의될 수 있다. 또한, 상기 정책은 P1 내지 P4로 분류될 수 있다. 이때, P1은 주로 수정 및 읽기/쓰기 작업을 수행하는 경우에 해당하고, P2는 쓰기 작업은 거의 수행하지 않고

주로 읽는 작업만 수행하는 경우에 해당한다. 또한, P3은 쓰기과 읽기 작업을 거의 하지 않는 경우에 해당하고, P4는 중요도가 떨어지는 데이터. 사용 가능성이 거의 없지만 단지 보관을 위해 사용하는 경우에 해당하는 것을 특징으로 할 수 있다.

- [0064] 한편, 상기 서비스 프로파일은 S1 내지 S4를 포함하고, 상기 S1 내지 S4 각각에 대하여, 상기 서비스 프로파일 별로 정규화된 가중치가 부여된다. 이때, 상기 P1 내지 P4 각각에 대하여, 상기 각각의 메트릭 별로 우선순위가 결정되고, 상기 정규화된 가중치와 상기 우선순위에 기반하여 상기 P1 내지 P4 각각에 대하여 상기 S1 내지 S4를 선택하는 순서를 결정한다. 이에 대해서는 아래에서 보다 상세히 살펴보도록 한다.
- [0065] **D. 서비스 자격 증명 요청 단계:** 상기 스토리지 서비스의 집합에 대한 서비스 자격 증명을 요청한다. 즉, 요청 인터셉터(Request Interceptor)에게 메타데이터 매니저(Metadata Manager)로부터 서비스 자격 증명을 요청한다. 이때 해당 서비스가 OAuth 프로토콜을 사용한다면 E 단계를 수행한다.
- [0066] **E. 인증 수행 단계:** 상기 서비스 자격 증명에 대한 인증을 수행한다. 즉, OAuth의 인증은 OAuth 매니저(OAuth Manager)를 통해 수행된다. 새로 고침 되었을 때 토큰이 만료된 경우 권한이 메타데이터 매니저(Metadata Manager)에 갱신되고 새로운 토큰이 요청 인터셉터(Request Interceptor)로 발급된다.
- [0067] **F. 요청 송신 단계:** 서비스 정보와 권한 인증을 상기 요청에 추가하고, 상기 요청을 스토리지 매니저로 전송한다. 즉, 서비스 정보와 권한 인증은 클라이언트의 요청에 추가되고 스토리지 매니저(Storage Manager)로 보내진다.
- [0068] ii. 스토리지 매니저(Storage Manager)는 요청을 일치하는 스토리지 서비스로 전달한다. 또한, 스토리지 매니저는 클라이언트에 대한 응답을 생성하게 된다.
- [0069] **A. 서비스 정보 전송 단계:** 가상 스토리지 자원 인터페이스(Virtual Storage Resource Interface)는 서비스 정보를 스토리지 서비스 인스턴스 매니저(Storage Service Instance Manager)로 전송한다.
- [0070] **B. 서비스 구현 클래스 탐색 단계:** 상기 서비스 정보에 기반하여 각각의 서비스에 대한 서비스 구현 클래스를 탐색한다. 즉, 스토리지 서비스 인스턴스 매니저(Storage Service Instance Manager)는 스토리지 서비스 정보 및 스토리지 서비스 인터페이스 상속에 정의된 고유한 이름에 따라 각 서비스의 구현 클래스를 찾을 때 반사 지향 프로그래밍방식을 사용한다.
- [0071] **C. 서비스 구현 클래스 인스턴스 생성/전송 단계:** 상기 탐색된 서비스 구현 클래스에 기반하여 서비스 구현 클래스 인스턴스를 생성하고, 상기 서비스 구현 클래스 인스턴스를 상기 가상 스토리지 자원 인터페이스로 전송한다. 즉, 각각 구현된 서비스 클래스 인스턴스들은 생성되어 가상 스토리지 자원 인터페이스(Virtual Storage Resource Interface)로 보내진다.
- [0072] **D. 요청/서비스 권한/서비스 구현 클래스 인스턴스 전송 단계:** 클라이언트의 요청, 서비스 권한 및 서비스 클래스들의 인스턴스들은 작업 매니저(Task Manager)로 보내진다.
- [0073] **E. 작업 처리/응답 병합 단계:** 상기 가상 스토리지 자원 인터페이스가 상기 작업 매니저로부터 수신한 응답을 상기 클라이언트가 수신할 수 있도록 최종 응답을 생성한다. 즉, 작업 매니저(Task Manager)는 비동기적으로 접근하는 모든 서비스를 처리하기 위해 병렬처리 기술을 사용한다. 그 후 모든 서비스들의 응답을 병합하여 가상 스토리지 자원 인터페이스(Virtual Storage Resource Interface)로 전송한다.
- [0074] **F. 최종 응답 전송 단계:** 가상 스토리지 자원 인터페이스(Virtual Storage Resource Interface)는 클라이언트의 요청에 응답하여 상기 최종 응답을 전송한다.
- [0075] 전술된 상기 정책 기반 스토리지 서비스 연함 수행 방법 및 장치는 다음과 같은 차별성을 갖는다.
- [0076] (가) 본 프레임워크는 통합된 다수의 파일 기반 이기종 스토리지 서비스에 대한 가능성을 제시한다. 예를 들어 클라우드 저장소, 분산 파일 시스템, 오브젝트 스토리지, ftp와 같은 스토리지의 데이터를 하나의 가상 스토리지에 단일 컨트롤 지점을 통해 관리하게 하는 것이다.
- [0077] i. 병렬처리 기술과 데이터 처리량 및 성능을 향상시킬 수 있다.
- [0078] ii. 수평 및 수직적 스케일링
- [0079] iii. 자동화된 결정
- [0080] (나) 클라이언트들의 요구사항을 최대한 정확하게 제공하기 위한 스토리지 영역의 번호를 정의한 정책 기반의

알고리즘

- [0081] (다) 클라이언트들이 자율적으로 데이터 분산 규칙을 정할 수 있도록 하는 스토리지 서비스 메트릭(metrics)을 기반으로 한 간단한 정책 모델 (예를 들어 가용성, 비용, 공간, 성능)
- [0082] 또한, 전술된 상기 정책 기반 스토리지 서비스 연합 수행 방법 및 장치는 다음과 같은 활용 분야에 적용될 수 있다.
- [0083] **활용분야(적용응용분야)**
- [0084] (1) 빅 데이터 저장소 (Big Data Storage)
- [0085] (2) 클라우드 연합 (Cloud Federation)
- [0086] (3) Storage as a Service (SaaS)
- [0087] (4) 병렬 데이터 처리 (Parallel Data Processing)
- [0088] (5) 기업용 저장소 (Enterprise Storage Solutions)
- [0089] 또한, 전술된 상기 정책 기반 스토리지 서비스 연합 수행 방법 및 장치는 다음과 같은 기대 효과를 갖는다.
- [0090] **(1) 정책 기반의 스토리지 서비스 선택**
- [0091] i. **스토리지 서비스 프로파일 정의:** 네 종류의 서비스 프로파일을 S1, S2, S3, S4로 정의하고 다섯 가지의 스토리지 메트릭(metrics)은 각 정보를 생성할 때 정의된다. 모든 메트릭(metrics)은 낮은 값일수록 서비스가 좋은 것으로 표현된다. 이를 바탕으로 메트릭(metrics)을 재정의 한다.
- [0092] A. 가용성(Availability): 서비스가 제공된 전체 사용 시간 중 정상적으로 서비스가 제공되지 않은 시간의 비율을 백분율로 나타내어 표현한다.
- [0093] B. 쓰기 및 읽기 성능(Write and Read Performance): 1MB 크기의 데이터를 처리할 때 걸리는 평균 시간으로 표현한다.
- [0094] C. 가격(Cost): 1GB 저장소를 사용할 때 드는 비용
- [0095] D. 사용한 공간(Used space): 현재 사용 중인 공간의 비율
- [0096] 이와 관련하여, 도 3은 본 발명의 예시에 따른 스토리지 서비스 프로파일을 도시한다. 또한, 도 4는 본 발명의 예시에 따른 정규화된 스토리지 서비스 프로파일을 도시한다. 또한, 도 5는 본 발명에 따른 정규화를 거친 서비스 프로파일과 정규화를 거치지 않은 서비스 프로파일을 비교한 도면이다.
- [0097] 도 3에 도시된 바와 같이, 서비스 프로파일 S1 내지 S4는 쓰기 성능(Write Performance), 읽기 성능(Read Performance), 가격(Cost), 사용한 공간(Used space) 및 가용성(Availability)에 대하여 서로 다른 가중치를 갖는다.
- [0098] ii. **스토리지 서비스 프로파일 정규화:** 도 4에 도시된 바와 같이, 서비스 프로파일은 선정 과정에서 각 메트릭(metric)의 기여를 공정하게 판단하기 위해 정규화를 수행해야 한다. 한편, 도 5에서 우측에 도시된 정규화를 거친 서비스 프로파일이, 그렇지 않은 경우보다 쓰기 성능(Write Performance), 읽기 성능(Read Performance), 가격(Cost), 사용한 공간(Used space) 및 가용성(Availability)에 대하여 어떤 경향이 있는 지 판별이 더 용이하다.
- [0099] 전술한 바와 같이, 본 발명에서는 동적이고 개인화된 정책을 기반으로 다수의 스토리지 서비스에 데이터를 분산 및 복제하고자 한다. 각 스토리지 서비스는 다수의 메트릭(metrics) 그룹으로 정의된다. 정책은 구체적인 데이터를 저장하기 위해 각 측정 항목의 중요도를 구체화한다. 마지막으로 많은 복제본이 제공된다. 스토리지 서비스는 유한한 개의 정의된 메트릭(metric)을 가지고 있다. 데이터의 저장 효율을 최고로 높이는 방법을 분석하기 위해 엄청난 양의 다른 메트릭(metric) 들이 있다. 그럼에도 불구하고 그것들의 값은 다른 범위일 뿐만 아니라 심지어는 값의 의미가 반대인 경우가 있어 비교할 때 혼란을 가중시킨다. 예를 들어 가용성은 높을수록 성능이 좋은 것이며, 비용은 값이 저렴할수록 좋은 것이다. 이처럼 가용성의 경우 사용자는 가능한 한 높은 가용성을 원할 것이다. 반대로 가격은 최소한으로 지불하기를 원할 것이다. 이러한 특성 때문에 객관적 비교를 위해 모든 값을 0과 1사이의 값으로 표현하는 정규화 작업이 필요하다. 따라서 본 특허에서는 모든 측정값에서 더 작은 값이 더 좋은 서비스라고 가정한다. 예를 들면, 가용성은 서비스가 제대로 제공되지 않은 비율로 정의한다. 따라

서 값이 낮을수록 가용성은 우수한 것이다.

[0100] 한편, 본 발명에서는 가장 적당한 스토리지 서비스 그룹을 정할 수 있도록 정책의 집합을 가지고 있다. 각 정책은 두가지 특성으로 정의된다. 첫째로 조건은 데이터와 함께 정책에 대응되어야 한다. 둘째로 중요도 순서에 따라 각 서비스의 측정 항목에 대한 가중치를 두는 방식으로 정의한다. 이는 첫 순서가 가장 높은 가중치를 갖는 것을 의미한다.

[0101] **iii. 정책 정의:** 4개의 정책은 다음 조건으로 정의된다.

[0102] A. P1, 주로 수정 및 읽기/쓰기 작업을 수행하는 경우이다. 이 경우 가용성이 가장 우선시 된다. 예를 들어 문서 작업을 수행하는 경우.

[0103] B. P2, 쓰기 작업은 거의 수행하지 않고 주로 읽는 작업만 수행하는 경우. 결과적으로 가장 중요한 메트릭(metrics)은 읽기 성능과 가용성이다. 예를 들어 미디어 파일 혹은 PDF 문서를 사용하는 경우.

[0104] C. P3, 쓰기와 읽기 작업을 거의 하지 않는 경우. 예를 들어 백업용으로 생성해 놓은 데이터. 그러므로 이때 가장 중요한 특성은 가용 저장 공간과 가격이다.

[0105] D. P4, 중요도가 떨어지는 데이터. 사용 가능성이 거의 없지만 단지 보관을 위해 사용하는 경우이다. 이 경우 가격이 가장 중요한 메트릭(metric)이다.

[0106] **iv. 스토리지 서비스 선택:** 최종적으로 가장 좋은 선택 순서를 알아내기 위해 정의된 정책 기반의 선택 알고리즘을 적용한다. 이는 도 8에 나타나 있다.

[0107] A. P1 와 P2의 스토리지 서비스 선택 순서: [S4, S2, S1, S3].

[0108] B. P3의 스토리지 서비스 선택 순서: [S2, S1, S3, S4].

[0109] C. P4의 스토리지 서비스 선택 순서: [S4, S3, S2, S1].

[0110] 다음으로, 전술한 바와 같이, 정책 기반 스토리지 서비스 페더레이션 수행을 위한 수학적 정의에 대해 살펴보기로 하자.

[0111] 스토리지 서비스의 집합을 $S=\{s_1, s_2, \dots, s_n\}$ 라고 정의하고, 정책의 집합을 $P=\{p_1, p_2, \dots, p_m\}$, 복제 요인을 (replication factor) r 로 정의한다. 우리는 P와 R사이의 거리를 최소화 하는 r 부터 S까지의 크기를 가지는 부분집합 R을 선택하고자 한다.

[0112] **i. 스토리지 서비스 모델**

[0113] 각 스토리지 서비스는 벡터로 정의할 수 있다: $K_{S_i} = \{\phi_1^i, \phi_2^i, \dots, \phi_k^i\}$

[0114] 서비스 i 에 대한 J 번째 metric의 값은 아래와 같이 나타낸다:

$$\phi_j^i, \quad \phi_j^i \in [0, 1]$$

[0116] **ii. 정책 모델**

[0117] 각 정책은 벡터로 정의할 수 있다:

$$K_{P_i} = \{\varphi_1^i, \varphi_2^i, \dots, \varphi_k^i\}$$

[0119] 정책 i 를 기반으로 한 j 번째 metric의 중요도는 아래와 같이 나타낸다.

$$\varphi_j^i, \quad \varphi_j^i \in \{1, 2, \dots, k\}$$

[0121] **iii. 메트릭 가중치(Metrics weight)**

[0122] 메트릭 가중치는 기하급수적으로 중요도 순서에 따라서 분배 된다. (The weight of metric) 정책 i 를 기반으로

하는 통계 가중치 j 는 아래와 같이 나타낸다.

$$f_W(p_i, j) = e^{-\lambda \varphi_j^i}$$

모든 정책 P 에 대한 통계 가중치 j 는 아래와 같이 나타낸다.

$$f_W(P, j) = \sum_{i=1}^m f_W(p_i, j)$$

한편, 서비스 선택 문제를 해결하기 위해서, 우리는 k 근접 이웃 알고리즘을 사용하였다. 일반적으로 k -NN은 근접한 이웃에 따른 질의 지점을 분류하는데 사용한다. 본 문제에 적용하기 위해서, 쿼리 포인트 Q 는 모든 스토리지 차원의 metrics을 최대화 시키는 공간의 중심에 있고, k (이웃의 수)는 r (복제 요인, replication factor)에 의해 결정된다. 거리 측정을 위해서는 유클리드 거리를 사용하였다. 각 특성은 정책에 대한 가중치가 반영된다.

$$D(s_i) = \sqrt{\sum_{j=1}^k (\phi_j^i \times f_W(P, j))^2}$$

이와 관련하여, 도 6은 본 발명의 예시에 따른 정책에 대하여 각각의 메트릭들의 순위를 정의한 도면이다. 또한, 도 7은 본 발명의 예시에 따른 정책에 대하여 각각의 메트릭들의 비율을 막대 그래프로 도시한 도면이다. 도 8은 본 발명의 예시에 따른 정책 기반 서비스 거리를 방사형으로 도시한 도면이다.

도 6을 참조하면, $P1$ 및 $P2$ 에 대해서는 가용성(Availability)이 가장 중요하고, $P3$ 에 대해서는 사용한 공간(Used space)이 가장 중요하고, $P4$ 에 대해서는 가격(Cost)이 가장 중요함을 알 수 있다.

도 7을 참조하면, $P1$ 내지 $P4$ 에 대하여 각각 쓰기 성능(Write Performance), 읽기 성능(Read Performance), 가격(Cost), 사용한 공간(Used space) 및 가용성(Availability)이 어떠한 비율로 그 중요도가 정의되는지를 일목요연하게 알 수 있다.

(2) 병렬 처리 성능

본 프레임워크에서 striped files 과 no-striped files의 병렬 처리의 성능을 평가하기 위해서 우리는 stripe 된 다른 블록 크기의 여러 파일 사이즈를 사용하였다. 이와 관련하여, 도 9a 내지 도 9c는 본 발명의 예시에 따른 쓰기 작업 수행, 읽기 작업 수행, 및 삭제 작업 수행에서, 파일 크기와 블록 크기에 따른 스루풋(throughput)을 도시한다. 도 9a 내지 도 9c에 도시된 바와 같이, 각 블록 사이즈 별 읽기 쓰기 및 삭제를 수행 및 평가하였다. 수행 결과 Striping과 병렬 처리를 조합한 방법이 성능과 데이터 처리량을 향상시킨다는 것을 확인하였다.

이상에서는, 정책 기반 서비스 페더레이션 프레임워크와 상기 프레임워크를 이용하는 정책 기반 서비스 페더레이션 수행 방법 및 장치에 대하여 살펴보았다. 상기 프레임워크가 설치되는 페더레이션 수행 장치는 단말, 사용자 기기이고, 상기 단말, 사용자 기기를 통해 복수의 스토리지 서비스 장치(예를 들어, 클라우드 스토리지, 분산 파일 시스템 및 오브젝트 스토리지)에 연결될 수 있다.

일 수 있다. 또한, 상기 프레임워크가 설치되는 페더레이션 수행 장치는 게이트 웨이이고, 상기 게이트웨이를 통해 복수의 스토리지 서비스 장치(예를 들어, 클라우드 스토리지, 분산 파일 시스템 및 오브젝트 스토리지)에 연결될 수 있다.

도 10은 본 발명의 예시에 따른 정책 기반 스토리지 서비스 선택을 위한 알고리즘의 구성을 도시한다. 도 10에 도시된 바와 같이, 스토리지 서비스 집합을 S 라 하고, 메트릭들의 수를 k , 복제 요인을 (replication factor) r 로 정의한다. 즉, 전술된 바와 같이, 스토리지 서비스의 집합을 $S=\{s_1, s_2, \dots, s_n\}$ 라고 정의하고, 정책의 집합을 $P=\{p_1, p_2, \dots, p_m\}$, 복제 요인을 (replication factor) r 로 정의한다. 우리는 P 와 R 사이의 거리를 최소

화 하는 r 부터 S 까지의 크기를 가지는 부분집합 R 을 선택하고자 한다.

[0136] 본 프레임워크는 페더레이션 레이어 컴포넌트를 제공하여 손쉽게 어플리케이션을 개발할 수 있게 한다. 한편 상 기 구현한 REST API를 기반으로 독립 서비스를 제공할 수 있다.

[0137] 본 발명의 적어도 일 실시예에 따르면, 수평 및 수직적 스케일링과 병렬처리 기술을 통해 데이터 처리량 및 성 능을 향상시킬 수 있어, 빠른 어플리케이션 개발을 위한 스토리지 페더레이션 레이어 계층에 해당하는 프레임워 크(framework)를 제공할 수 있다.

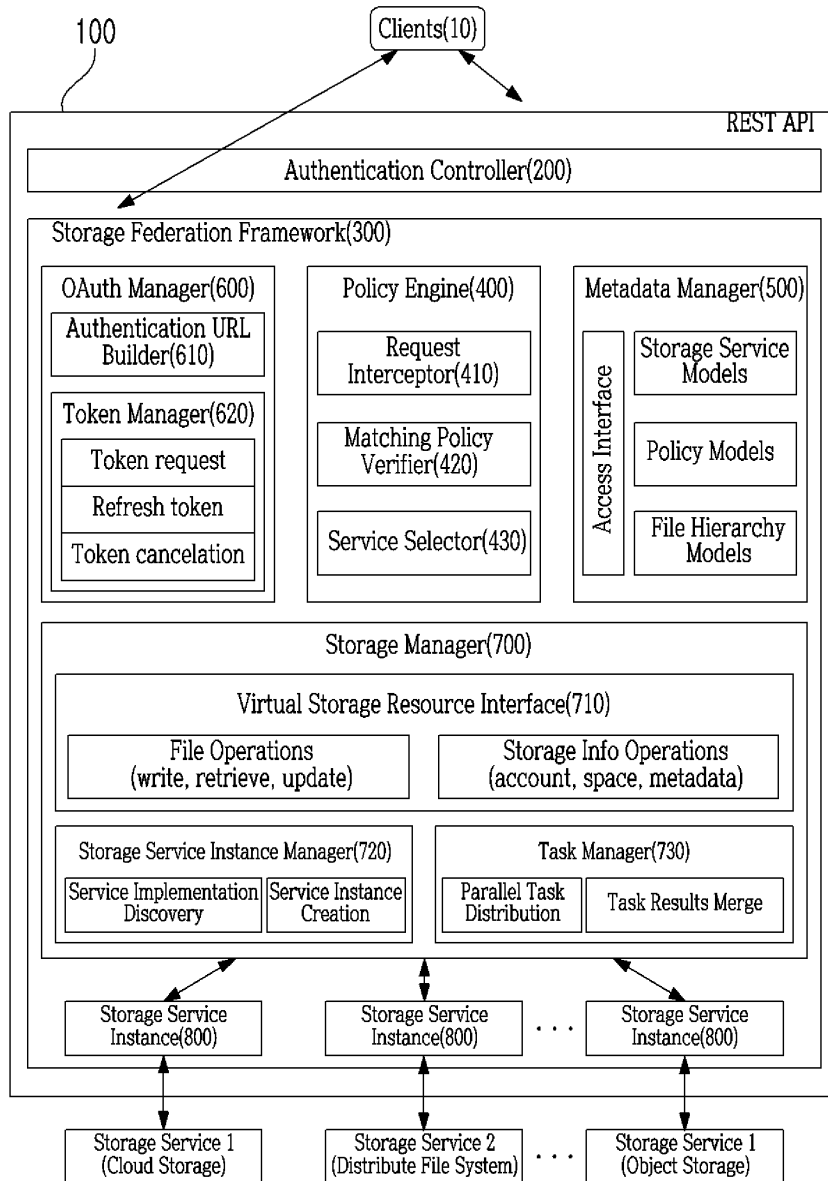
[0138] 또한, 본 발명의 적어도 일 실시예에 따르면, 서비스 프로파일을 통한 정책 기반의 스토리지 서비스에 의해, 타 시스템과의 호환이 가능하면서, 정책 기반의 서비스 선택이 가능한 정책 기반 스토리지 서비스 프레임워크 (framework)를 제공할 수 있다는 장점이 있다.

부호의 설명

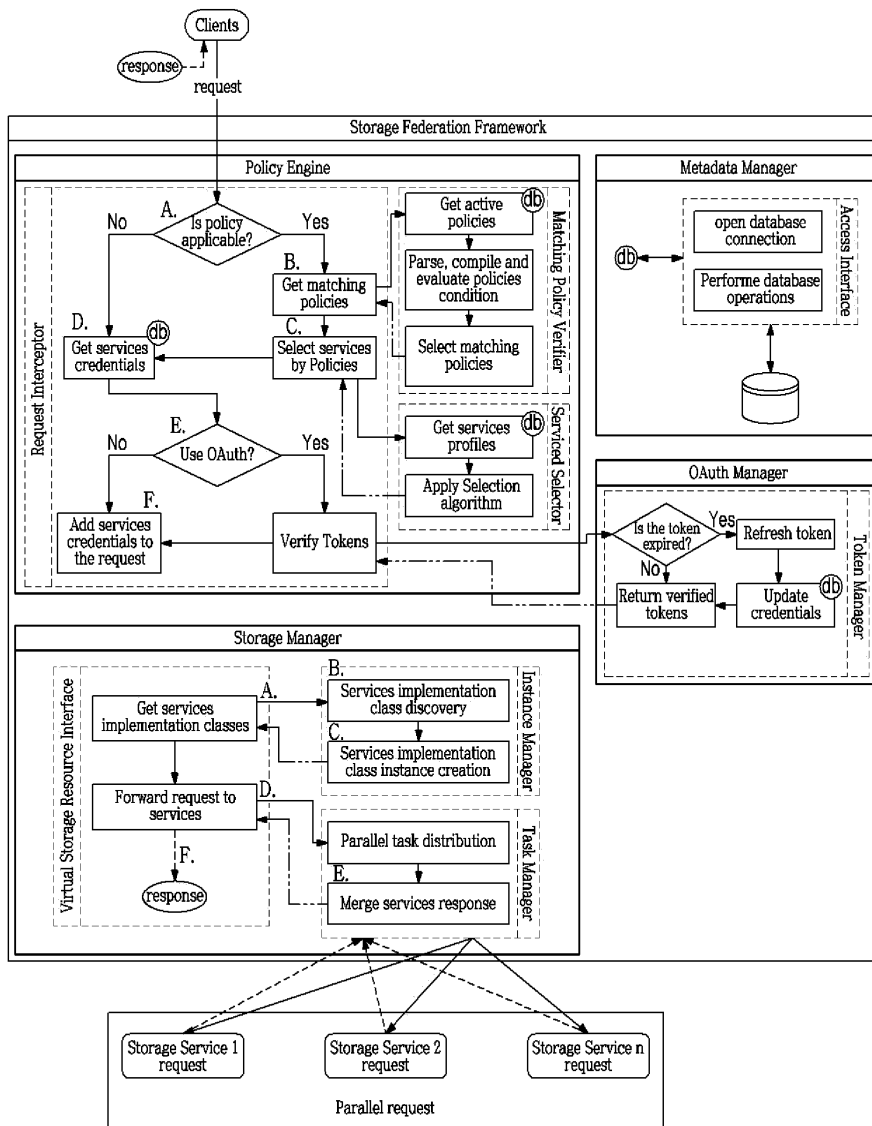
[0139] 1000: 스토리지 서비스 연합 수행 장치
 10: 클라이언트 100: REST API
 200: 인증 컨트롤러 300: 스토리지 페더레이션 프레임워크
 400: 정책 엔진 410: 요청 인터셉터
 420: 매칭 정책 검증자 430: 서비스 선택자
 500: 메타데이터 매니저 600: 인증 매니저
 610: 인증 URL 생성자 620: 토큰 매니저
 700: 스토리지 매니저 710: 가상 스토리지 자원 인터페이스
 720: 인스턴스 매니저 730: 작업 매니저
 800: 스토리지 서비스 인터페이스

도면

도면1



도면2



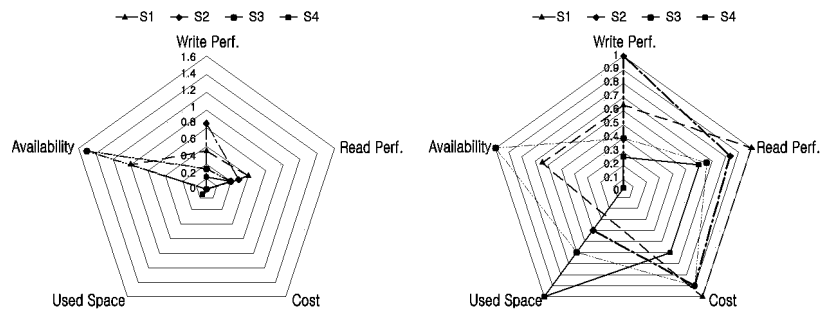
도면3

Service	Write Perf.	Read Perf.	Cost	Used Space	Availability
S1	0.530	0.570	0.011	0.022	0.992
S2	0.833	0.466	0.010	0.037	0.025
S3	0.313	0.370	0.010	0.054	1.511
S4	0.211	0.348	0.007	0.090	0.007

도면4

Service	Write Perf.	Read Perf.	Cost	Used Space	Availability
S1	0.636	1.000	1.000	0.244	0.656
S2	1.000	0.818	0.876	0.411	0.017
S3	0.376	0.650	0.864	0.600	1.000
S4	0.253	0.611	0.604	1.000	0.005

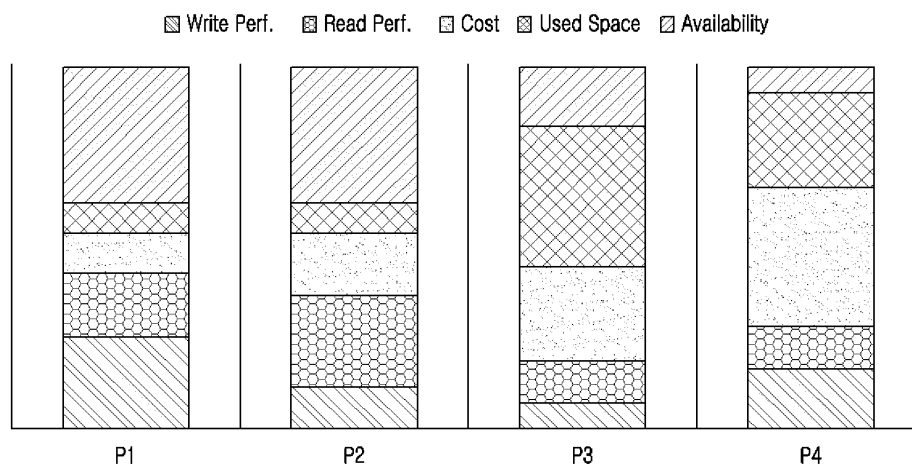
도면5



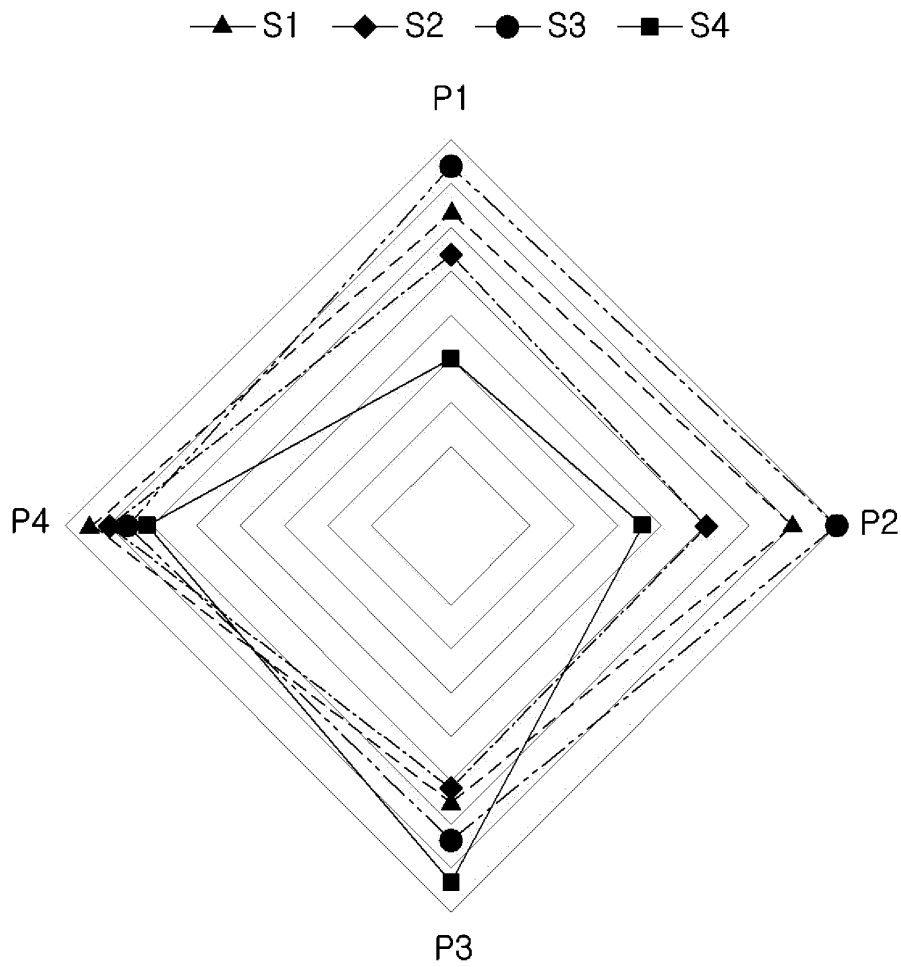
도면6

Policy	Write Perf.	Read Perf.	Cost	Used Space	Availability
P1	2	3	4	5	1
P2	4	2	3	5	1
P3	5	4	2	1	3
P4	3	4	1	2	5

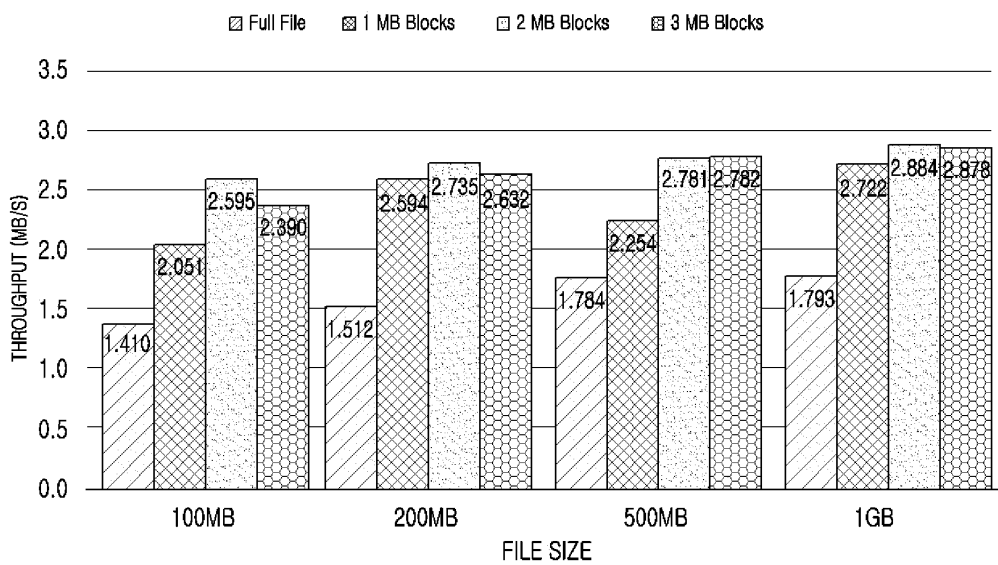
도면7



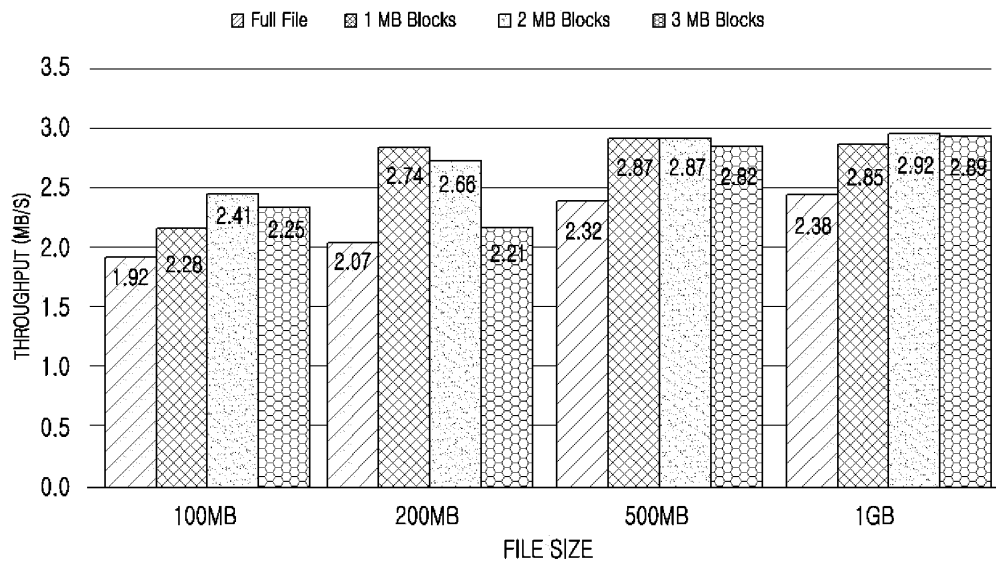
도면8



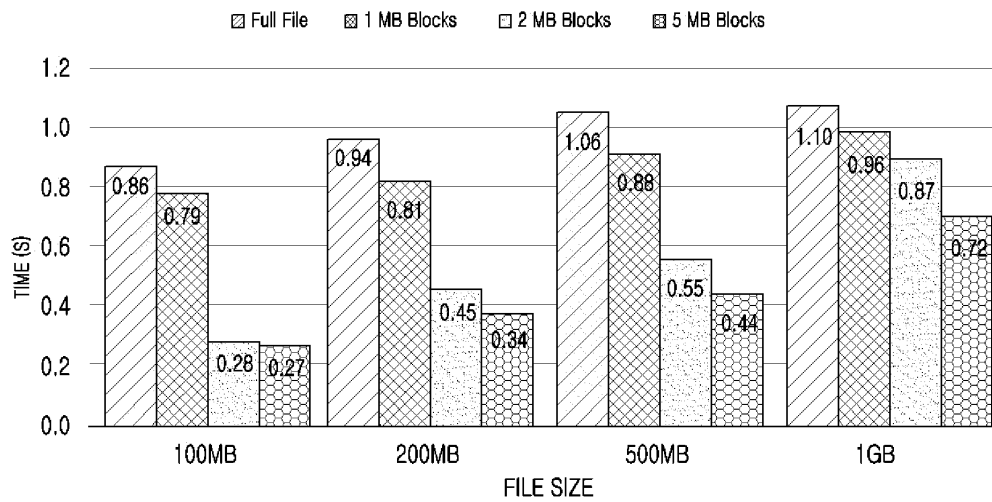
도면9a



도면9b



도면9c



도면10

Algorithm1: Policy-based Storage Service Selection

input : Storage services set S , Policies set P , Number of metrics k , Replication factor r
output : Selected storage services set

```

if size( $S$ )  $\leq r$  then
    return  $S$ ;
end
else
     $fw[k]$ ;
    for  $j=1 \rightarrow k$  do
         $fw[j] \leftarrow 0$ ;
        foreach  $p_i \in P$  do
            /*  $j$ -th metric weight by policy  $p_i$ 
            */
             $fw[j] \leftarrow fw[j] + p_i[j]$ ;
        end
    end
    foreach  $s_i \in S$  do
         $dist \leftarrow 0$ ;
        for  $j=1 \rightarrow k$  do
             $s_i[j] \leftarrow s_i[j] * fw[j]$ 
             $dist \leftarrow dist + s_i[j] * s_i[j]$ ;
        end
         $s_i.d \leftarrow \text{sqrt}(dist)$ ;
    end
     $S \leftarrow \text{orderByDistance}(S)$ ;
    return  $S[1:r]$ ;
end

```