



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년03월25일
(11) 등록번호 10-1378182
(24) 등록일자 2014년03월19일

(51) 국제특허분류(Int. Cl.)
G06F 11/36 (2006.01) G06F 11/28 (2006.01)
(21) 출원번호 10-2012-0130737
(22) 출원일자 2012년11월19일
심사청구일자 2012년11월19일
(56) 선행기술조사문헌
KR1020070058943 A*
KR1020120070215 A*
KR1020120051933 A
KR1020110134582 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
최병주
서울 강남구 압구정로 201, 74동 1305호 (압구정동, 현대아파트)
(74) 대리인
남정길

전체 청구항 수 : 총 5 항

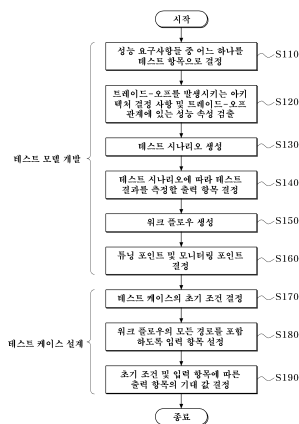
심사관 : 김상택

(54) 발명의 명칭 트레이드-오프 기반의 테스트 케이스 생성 방법

(57) 요약

개시된 기술은 소프트웨어 성능 평가 시 사용될 수 있는 테스트 케이스를 생성하는 방법에 관한 것으로, 보다 상세하지만 제한된 없이는, 소프트웨어 아키텍처 상의 트레이드-오프 특성을 기반으로 하는 테스트 케이스 생성 방법에 관한 것이다. 실시예들 중에서, 테스트 케이스 생성 방법은 소프트웨어의 성능 속성들 중 어느 하나를 테스트 항목으로 선택하는 단계; 상기 소프트웨어의 아키텍처 결정 사항들 중 상기 테스트 항목에 트레이드-오프를 발생시키는 아키텍처 결정 사항 및 상기 아키텍처 결정 사항에 따라 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 검출하는 단계; 및 상기 테스트 항목 및 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 기반으로 테스트 결과를 측정할 출력 항목들을 결정하는 단계를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1415123332
부처명	지식경제부
연구사업명	대학IT연구센터 육성지원
연구과제명	융합 소프트웨어를 위한 요구 및 품질검증 기술 개발
기 여 율	1/1
주관기관	서강대학교 산학협력단
연구기간	2012.01.01 ~ 2012.12.31

특허청구의 범위

청구항 1

소프트웨어의 성능 속성들 중 어느 하나를 테스트 항목으로 선택하는 단계;

상기 소프트웨어의 아키텍처 결정 사항들 중 상기 테스트 항목에 트레이드-오프를 발생시키는 아키텍처 결정 사항 및 상기 아키텍처 결정 사항에 따라 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 검출하는 단계;

상기 테스트 항목 및 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 기반으로 테스트 결과를 측정할 출력 항목들을 결정하는 단계; 및

상기 테스트 항목, 상기 아키텍처 결정 사항 및 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 기반으로 생성된 테스트 시나리오의 흐름을 나타내는 워크 플로우를 생성하는 단계를 포함하는 테스트 케이스 생성 방법.

청구항 2

삭제

청구항 3

제1항에 있어서, 상기 워크 플로우를 생성하는 단계는,

상기 워크 플로우 상에서 상기 트레이드-오프가 발생하는 위치를 검출하는 단계;

상기 트레이드-오프가 발생하는 위치에서, 상기 트레이드-오프와 관련된 함수를 확장하여 표현하는 단계; 및

상기 확장된 워크 플로우에서 상기 트레이드-오프가 발생하는 경로와 발생하지 않는 경로의 분기 조건을 검출하는 단계를 포함하는 테스트 케이스 생성 방법.

청구항 4

제1항에 있어서,

상기 트레이드-오프가 발생하는 경우와 발생하지 않는 경우가 모두 테스트될 수 있도록 테스트 케이스의 초기 조건을 결정하는 단계를 더 포함하는 테스트 케이스 생성 방법.

청구항 5

제1항에 있어서,

상기 트레이드-오프가 발생하는 경우와 발생하지 않는 경우 값이 변경되는 변수를, 테스트 결과 분석을 위하여 측정할 모니터링 포인트로 결정하는 단계를 더 포함하는 테스트 케이스 생성 방법.

청구항 6

제1항에 있어서,

상기 트레이드-오프가 발생하는 경우와 발생하지 않는 경우의 분기 조건을, 테스트 수행 시 값이 조정될 수 있는 튜닝 포인트로 결정하는 단계를 더 포함하는 테스트 케이스 생성 방법.

명세서

기술 분야

개시된 기술은 소프트웨어 성능 평가 시 사용될 수 있는 테스트 케이스를 생성하는 방법에 관한 것으로, 보다 상세하지만 제한됨 없이는, 소프트웨어 아키텍처 상의 트레이드-오프 특성을 기반으로 하는 테스트 케이스 생성 방법에 관한 것이다.

[0001]

배경 기술

[0002] 개발되는 소프트웨어의 성능을 평가하기 위한 방법으로, 개발 이전에 성능을 예측 및 평가하는 연구와, 개발이 끝난 이후에 성능 요구사항이 만족되었는지를 판단하기 위한 성능 테스트가 있다. 일반적으로 이루어지는 성능 테스트 방법으로 시나리오 기반의 블랙 박스 기법의 테스트가 있다. Experience with performance testing of software systems(IEEE Transactions on, vol.26, no.12, pp.1147-1156) 및 Deriving Performance Requirements and Test Cases with the Performance Refinement and Evolution Model(Department of Computer Science, North Carolina State University Technical Report No. TR-2006-30)를 참조하면, 성능 요구사항을 기반으로 테스트 시나리오가 도출될 수 있다. 하지만, 시스템 레벨에서의 명세 기반 성능 테스트는 특정 부하 조건에서의 성능을 측정하는데 초점을 맞추고 있기 때문에 성능 문제의 원인을 식별하기 힘들고, 시스템의 개발이 완료된 상태라 성능 문제를 해결하기 힘들다는 한계점이 있다.

[0003] 소프트웨어 아키텍처(Software Architecture: SA)는 상위 추상화 단계에서 소프트웨어의 구조를 표현한 것으로, 소프트웨어의 구조에 대한 중요한 의사 결정의 집합이다. 소프트웨어 아키텍처를 기반으로 소프트웨어의 구현이 이루어지므로 소프트웨어 아키텍처는 소프트웨어의 성능에 많은 영향을 미친다. 따라서, 소프트웨어의 성능은 소프트웨어 아키텍처 설계 단계에서 거의 결정될 수 있으며, 소프트웨어 아키텍처를 통하여 성능 예측이 가능하다. 아키텍처 평가 방법 중 ATAM(Architecture Tradeoff Analysis Method)은 아키텍처가 초기에 의도했던 품질 목표를 만족시키고 있는지 분석하고, 아키텍처 결정사항의 트레이드-오프를 분석하여 위험요소를 식별함으로써 아키텍처의 적합성을 판단할 수 있도록 한다.

발명의 내용

해결하려는 과제

[0004] 개시된 기술이 이루고자 하는 기술적 과제는 트레이드-오프 특성을 이용하여 테스트 케이스를 생성하는 방법을 제공하는 데 있다. 비 기능 테스트의 경우 기능 테스트와 달리 체계적인 테스트 케이스 도출 방안이 구축되어 있지 않으며, 테스트의 당위성을 판단하기 위한 기준이 충분하지 않다. 본 명세서에서는 소프트웨어의 아키텍처 결정 사항에 따라 발생하는 성능 속성 간의 트레이드-오프 특성을 기반으로 테스트 케이스를 체계적으로 설계하는 방안을 제안한다.

과제의 해결 수단

[0005] 상기의 기술적 과제를 이루기 위하여 개시된 기술의 제1 측면은 소프트웨어의 성능 속성들 중 어느 하나를 테스트 항목으로 선택하는 단계; 상기 소프트웨어의 아키텍처 결정 사항들 중 상기 테스트 항목에 트레이드-오프를 발생시키는 아키텍처 결정 사항 및 상기 아키텍처 결정 사항에 따라 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 검출하는 단계; 및 상기 테스트 항목 및 상기 테스트 항목과 트레이드-오프 관계에 있는 성능 속성을 기반으로 테스트 결과를 측정할 출력 항목들을 결정하는 단계를 포함하는 테스트 케이스 생성 방법을 제공한다.

발명의 효과

[0006] 개시된 기술의 실시 예들은 다음의 장점들을 포함하는 효과를 가질 수 있다. 다만, 개시된 기술의 실시 예들이 이를 전부 포함하여야 한다는 의미는 아니므로, 개시된 기술의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.

[0007] 개시된 기술에 따르면, 테스트 항목의 side-effect 분석이 용이한 테스트 케이스를 체계적으로 생성할 수 있다는 장점이 있다. 개시된 기술에서는 테스트 항목과 트레이드-오프 관계에 있는 성능 속성도 성능 지표로 포함함으로써, 상호간에 미치는 side-effect 를 평가할 수 있으며, 이러한 영향이 테스트 결과에 반영될 수 있도록 테스트 케이스의 초기 조건을 체계적으로 설정할 수 있다. 따라서, 성능 테스트를 통하여 성능 요구사항을 만족시키는 지 확인하는 것과 더불어 Side-effect도 확인하여 잠재적 위험에 대한 대비를 할 수 있다는 효과가 있다. 또한, 개시된 기술에서 제안한 모니터링 포인트는 테스트 결과를 문제의 원인과 연관하여 용이하게 분석할 수 있도록 한다는 장점이 있다.

[0008] 개시된 기술에서는 튜닝 포인트를 새로이 정의하여 사용함으로써, 소프트웨어의 성능을 향상시킬 수 있다는 효과가 있다. 개시된 기술에 따르면 튜닝 포인트의 값을 조절함으로써, 소프트웨어의 성능을 튜닝 할 수 있으며,

이러한 튜닝 과정에 대한 테스트 결과를 기반으로 소프트웨어의 성능을 향상시킬 수 있다.

도면의 간단한 설명

- [0009] 도 1은 개시된 기술의 일 실시예에 따른 테스트 케이스 생성 방법을 설명하기 위한 순서도이다.
- 도 2는 YAFFS2의 성능 속성 카테고리를 나타낸 표이다.
- 도 3은 YAFFS2와 관련된 성능 요구사항들 및 아키텍처 결정사항들의 일부를 나타낸 표이다.
- 도 4는 개시된 기술의 일 실시예에 따른 방법을 YAFFS2에 적용하여 생성된 테스트 모델을 나타낸다.
- 도 5는 개시된 기술의 일 실시예에 따라 생성된 테스트 케이스의 일부를 나타낸다.
- 도 6은 개시된 기술의 일 실시예에 따라 워크 플로우를 생성하는 과정을 설명하기 위한 순서도이다.
- 도 7은 테스트 항목이 '파일 쓰기 속도'이고 제1 아키텍처 결정사항이 '가비지 컬렉션'인 경우 생성된 워크 플로우이다.
- 도 8은 도 5의 테스트 케이스에 따른 테스트 결과를 그래프로 나타낸 도면이다.
- 도 9는 튜닝 전과 튜닝 후의 성능 변화를 나타낸 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0010] 개시된 기술에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 개시된 기술의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 개시된 기술의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다.
- [0011] 한편, 본 출원에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.
- [0012] “제1”, “제2” 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.
- [0013] 어떤 구성요소가 다른 구성요소에 “연결되어” 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 “직접 연결되어” 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 “~사이에”와 “바로 ~사이에” 또는 “~에 이웃하는”과 “~에 직접 이웃하는” 등도 마찬가지로 해석되어야 한다.
- [0014] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, “포함하다” 또는 “가지다” 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0015] 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0016] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 개시된 기술이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.
- [0017] 도 1은 개시된 기술의 일 실시예에 따른 테스트 케이스 생성 방법을 설명하기 위한 순서도이다. 개발된 소프트웨어의 성능을 테스트하는 테스트 장치는 우선, 테스트 케이스를 생성하여야 한다. 개시된 기술에서는 아키텍처 결정사항에 따라 성능 속성들 간에 발생하는 트레이드-오프를 고려하여 테스트 장치가 체계적으로 테스트 케이스를 생성하는 방법을 제안한다. 본 명세서에서, 트레이드-오프(trade-off)란 아키텍처적 의사 결정이 하나 이상의 성능 속성에 영향을 미치는 것을 의미한다. 예를 들어, 가상 사설 네트워크에서 암호화 비트 수를 증가시

키면 암호화 수준이 높아져 신뢰도가 향상되지만, 암호화를 위한 처리시간이 더 필요해 지므로 시간 반응성을 저하시킬 수 있다. 즉, 암호화 비트 수를 증가시키는 것(아키텍처 의사결정에 대응됨)은 성능 속성인 신뢰성과 시간 반응성 간에 트레이드-오프를 발생시키며, 신뢰성과 시간 반응성은 서로 트레이드-오프 관계에 있다고 표현할 수 있다. 또한, 아키텍처 결정사항이란, 품질 속성을 달성하는데 직접적인 영향을 주는 아키텍처의 측면이다. 일반적으로 컴포넌트 간의 관계 또는 컴포넌트의 특성이 아키텍처 결정사항에 해당될 수 있다. 이하, 도 1을 참조하여 아키텍처 결정사항을 분석하고 트레이드-오프를 파악하여 성능 테스트의 테스트 케이스를 도출하는 방법을 설명한다. 이해를 돕기 위하여 본 명세서에서는 NAND 플래시 메모리 파일 시스템 YAFFS2의 성능을 테스트하는 경우를 가정하여 테스트 케이스를 생성하는 방법을 예를 들어 설명한다. 도 2는 YAFFS2의 성능 속성 카테고리를 나타낸 표이다. YAFFS2의 품질 속성은 도 2와 같이 분류할 수 있으며, 예컨대 시간 반응성에 해당하는 성능 속성은 '파일 읽기 속도', '파일 쓰기 속도', '마운트 속도'로 분류된다. 도 3은 YAFFS2와 관련된 성능 요구사항들 및 아키텍처 결정사항들의 일부를 나타낸 표이다. 도 2에서와 같이 분류된 성능 속성은 소프트웨어를 평가하기 위한 평가 항목으로 성능 요구사항과 매핑될 수 있다. 성능 요구사항들은 일반적으로 소프트웨어 개발 단계에서 미리 정해지며, 아키텍처 결정사항 및 그에 따라 발생하는 성능 속성 간 트레이드-오프는 소프트웨어 아키텍처를 분석함으로써 도출될 수 있다.

[0018] 도 1을 참조하면, 실시예에 따라, 테스트 케이스를 생성하는 과정은 다시 테스트 모델을 개발하는 과정(S110 내지 S160)과 테스트 케이스를 설계하는 과정(S170 내지 S190)으로 나뉘어 질 수 있다. 도 4는 개시된 기술의 일 실시예에 따른 방법을 YAFFS2에 적용하여 생성된 테스트 모델을 나타낸다. 도 4를 참조하면, 본 실시예에 따른 테스트 모델은 크게 테스트 항목, 테스트 시나리오, 기대 결과, 워크플로우 모델, 튜닝 포인트 그리고 모니터링 포인트 6가지로 구성된다. 테스트 장치는 성능 요구사항들과 아키텍처 결정사항들을 기초로 S110 단계 내지 S160 단계를 통하여 테스트 모델을 개발한다. 도 5는 개시된 기술의 일 실시예에 따라 생성된 테스트 케이스의 일부를 나타낸다. 도 5를 참조하면, 테스트 케이스는 초기 조건, 입력, 예상 출력으로 구성된다. 테스트 장치는 상기 개발된 테스트 모델을 기반으로 S170 단계 내지 S190 단계를 통하여 테스트 케이스의 초기조건, 입출력 항목 등을 결정함으로써 테스트 케이스를 설계한다.

[0019] 다시, 도 1을 참조하면 S110 단계에서, 테스트 장치는 소프트웨어의 성능 속성들 중 어느 하나를 테스트 항목으로 선택한다. 테스트 항목은 테스트 하고자 하는 성능 속성이다. 예를 들어, 테스트 장치는 도 2의 성능 속성들 중 시간 반응성에 해당하는 파일 쓰기 속도를 테스트 항목으로 선택할 수 있다.

[0020] S120 단계에서, 테스트 장치는 소프트웨어의 아키텍처 결정 사항들 중 S110 단계에서 결정된 테스트 항목에 트레이드-오프를 발생시키는 아키텍처 결정 사항을 검출한다. 이를 위하여, 아키텍처 결정 사항이 테스트 항목에 어떻게 영향을 미치는지, 다른 성능 속성간의 상충되는 요소가 있는지, 상충되는 부분이 어떠한 위험을 유발시킬 수 있는지에 대해 분석한다. 도 3의 예를 들면, 테스트 장치는 테스트 항목인 파일 쓰기 속도를 트레이드-오프로 가지는 아키텍처 결정사항으로 캐시와 가비지 콜렉션을 검출할 수 있다. 테스트 장치는 검출된 아키텍처 결정사항이 복수개인 경우, 그 중 어느 하나를 선택해 가면서 테스트 케이스를 생성하는 과정을 반복할 수 있다. 본 실시예에서는 가비지 콜렉션(이하, 제1 아키텍처 결정사항이라 함)을 검출, 선택한 경우를 설명한다. 또한, 테스트 장치는 제1 아키텍처 결정사항과 관련하여 테스트 항목과 트레이드-오프 관계에 있는 성능 속성(이하, 제1 성능 속성이라 함)을 검출한다. 도 3의 예를 들면, 테스트 장치는 가비지 콜렉션과 관련하여 테스트 항목인 파일 쓰기 속도(시간 반응성)와 트레이드-오프 관계에 있는 NAND의 여유 공간 확보(자원 효율성)를 검출할 수 있다.

[0021] S130 단계에서, 테스트 장치는 테스트 항목, 제1 아키텍처 결정사항 및 제1 성능 속성을 기초로 테스트 시나리오를 생성한다. 본 실시예에 따르면, 테스트 장치는 아키텍처의 결정사항이 테스트 항목에 직접적으로 영향을 미치는 경우, 또는 다른 품질 속성에 영향을 미치는 상황에 대해서 검증할 수 있는 테스트 시나리오를 개발할 수 있다. 이때, 아키텍처 결정사항이 지닌 특성에 따라 한 개의 아키텍처 결정사항에 대해 한 개 이상의 테스트 시나리오가 도출될 수 있다. 예컨대, 테스트 장치는 테스트 항목과 관련된 성능 요구사항과, S120 단계에서 검출된 제1 아키텍처 결정사항의 테스트 항목과 제1 성능 속성 간 트레이드-오프를 활용하여 테스트 시나리오를 생성할 수 있다. 도 3을 참조하여 예를 들어 설명하면, 테스트 항목인 '파일 쓰기 속도'는 구체적으로, “파일 시스템이 정상적으로 마운트된 환경에서 파일 쓰기 연산이 들어올 때, 1.8MB/sec을 보장해야 한다”는 성능 요구사항이 정해져 있다. 제1 아키텍처 결정사항인 '가비지 콜렉션'은 이를 사용할 경우, 쓰기 속도가 저하되는 부작용(side-effect)을 포함하고 있지만 NAND의 가용 공간을 확보하여 자원 효율성을 향상시키는 장점이 있다. 즉, 가비지 콜렉션과 관련하여 쓰기 속도와 여유 공간 확보 간에는 트레이드-오프 관계가 있다. 이러한 특성을 기반으로, '파일 쓰기를 수행할 때, 가비지 콜렉션을 수행하여 가용공간을 확보하며, 파일 쓰기 속도 평균

1.8MB/sec를 보장해야 한다.'와 같은 테스트 시나리오가 생성될 수 있다.

[0022]

S140 단계에서 테스트 장치는, 생성된 테스트 시나리오에 따라 테스트 결과를 측정할 출력 항목들을 결정한다. 출력 항목은 기능과 성능으로 나뉠 수 있다. 기능은 시나리오가 정상적으로 완료되었을 경우, 기능적인 측면에서의 어떻게 출력되는지를 의미하며, 성능은 성능 요구사항 및 트레이드-오프 특성으로부터 도출될 수 있다. 테스트 항목에 트레이드-오프가 발생하면 제1 성능 속성에 어떠한 부작용(side-effect)이 있는지를 분석할 수 있도록, 성능은 트레이드-오프(+) 항목과 트레이드-오프(-) 항목으로 구성된다. S130 단계에서 예를 들어 설명한 테스트 시나리오(파일 쓰기를 수행할 때, 가비지 콜렉션을 수행하여 가용공간을 확보하며, 파일 쓰기 속도 평균 1.8MB/sec를 보장해야 한다)가 정상적으로 완료될 경우, 기능적인 측면에서 '파일 쓰기가 정상적으로 완료'된다고 예상 출력을 도출할 수 있다. 성능의 예상출력은 가비지 콜렉션의 트레이드-오프를 활용하여 '+'항목으로는 자원 효율성에 해당하는 '가용 공간 확보'로 도출할 수 있으며, '-'항목으로 시간 반응성에 해당하는 '파일 쓰기 속도 1.8MB/sec 보장'을 도출할 수 있다. 이를 기초로 테스트 장치는 테스트 케이스의 트레이드-오프(+) 출력 항목으로 '새로 확보한 여유 공간(Free Chunk 개수)', 트레이드-오프 (-) 출력 항목으로 '파일 쓰기 속도'를 결정할 수 있다. 한편, 실시예에 따라, 테스트 장치는 테스트 시나리오 없이 테스트 항목인 “파일 쓰기 속도(시간 반응성)”로부터 트레이드-오프(-) 출력 항목을, 제1 성능 속성인 “NAND의 여유 공간 확보(자원 효율성)”로부터 트레이드-오프(+) 출력 항목을 도출할 수도 있다. 즉, 테스트 장치는 테스트 시나리오, 또는 테스트 항목과 제1 성능 속성을 기반으로 테스트 결과를 측정할 출력 항목들을 결정한다.

[0023]

S150 단계에서 테스트 장치는 앞서 생성된 테스트 시나리오에 따라 워크 플로우를 생성한다. 테스트 시나리오는 S140 단계에서 설명하였듯이 테스트 항목, 제1 아키텍처 결정 사항 및 제1 성능 속성을 기반으로 생성될 수 있다. 워크 플로우는 테스트 케이스 도출 기준에 해당되는 테스트 커버리지에 해당된다. 워크 플로우는 테스트 시나리오의 흐름을 나타내며, 트레이드-오프가 어떻게 유발되는지 표현되도록 설계된다. 워크플로우는 트레이드-오프 발생위치, 트레이드-오프 관련 함수, 트레이드-오프 발생 경로, 트레이드-오프 미발생 경로, 및 트레이드-오프 발생 분기 조건이 표현될 수 있도록 생성한다. 일 실시예에 따라 워크 플로우를 생성하는 구체적인 방법을 도 6 및 도 7을 참조하여 설명한다.

[0024]

도 6은 개시된 기술의 일 실시예에 따라 워크 플로우를 생성하는 과정을 설명하기 위한 순서도이다. 도 7은 테스트 항목이 '파일 쓰기 속도'이고 제1 아키텍처 결정사항이 '가비지 콜렉션'인 경우 생성된 워크플로우이다. S610 단계에서 테스트 장치는, 사용자 입력 레벨의 워크 플로우를 생성한다. 도 7에서 흰 네모 상자들은 '파일 쓰기'를 위해 사용자가 입력할 수 있는 시스템 콜들을 나타낸다. 파일 쓰기 동작을 수행하는 경우 요청되는 사용자 입력 레벨의 작업인 흰 네모 상자들이 순차적으로 화살표로 연결되어, 가능한 모든 경로를 포함하는 워크 플로우가 생성된다. S620 단계에서 테스트 장치는, 워크 플로우 상에서 트레이드 오프가 발생하는 위치를 검출한다. 도 7의 실시예에서, 흰 네모 상자들 중 write() 시스템 콜이 가비지 콜렉션을 호출 하므로 트레이드-오프 발생 위치에 해당한다. S630 단계에서 테스트 장치는, 트레이드 오프가 발생하는 위치에서, 트레이드 오프와 관련된 함수를 확장하여 표현한다. 본 실시예에서는, 트레이드-오프로 인해 발생하는 이벤트의 흐름이 나타날 수 있도록, 워크 플로우를 확장하여 설계한다. 본 실시예에 따르면, 확장된 워크 플로우를 기반으로, 트레이드-오프 발생 경로와 미발생 경로를 모두 커버할 수 있도록 테스트 케이스가 도출되어, 테스트 항목의 부작용(side-effect)의 분석이 용이하다. 도 7의 예에서는, 트레이드-오프로 인한 이벤트의 흐름을 표현하기 위하여, YAFFS2의 가비지 콜렉션과 관련된 함수를 파란 네모상자로 표현한다. 여기서, 여유공간 n이 1280KB보다 작은 경우, aggressive 변수가 1로 선언된다. 이 경우의 경로는 가비지 콜렉션을 하기 위해 NAND 메모리의 모든 블록을 스캔하여 파일 쓰기 속도가 저하될 수 있으므로 트레이드-오프 발생 경로가 된다. 그렇지 않은 경우, aggressive 변수가 0으로 선언되고, NAND 메모리의 일부만 스캔하여 가비지 콜렉션을 수행하므로, 해당 경로는 트레이드-오프 미발생 경로가 된다. S640 단계에서 테스트 장치는, 확장된 워크 플로우에서 트레이드-오프가 발생하는 경로와 발생하지 않는 경로의 분기 조건을 검출한다. 도 7에서 확장된 워크 플로우 부분을 살펴보면, 트레이드-오프가 발생하는 분기 조건을 기준으로 트레이드-오프가 발생하는 경로와 트레이드-오프가 발생하는 않는 경로가 나누어 표현됨을 알 수 있다. 도 7의 경우, aggressive 변수 값이 다르게 설정되는 조건인 여유공간 n이 1280KB보다 작은지 여부가 분기 조건이 된다.

[0025]

다시 도 1을 참조하면, S160 단계에서 테스트 장치는 트레이드-오프가 발생하는 경우와 발생하지 않는 경우의 분기 조건을, 테스트 수행 시 값이 조정될 수 있는 튜닝 포인트로 결정한다. 튜닝 포인트란, 성능을 향상시키기 위하여 튜닝이 필요한 변수 조건이다. 일 실시예에 따라, 튜닝 포인트는 도 7과 같은 워크플로우 모델에서 트레이드-오프 발생 경로와 미 발생 경로로 나뉘는 분기의 조건으로 도출될 수 있다. 이는 트레이드-오프가 발생하는 조건을 튜닝 함으로써 성능이 튜닝 될 수 있기 때문이다. 도 7을 참조하면, 트레이드-오프 발생 경로와 미발

생 경로로 나뉘는 분기 조건은 ' $n < 1280$ '임을 알 수 있다. 그러므로, 본 실시예에서의 튜닝 포인트는 aggressive 변수를 1로 선언하는 초기 여유 공간의 양으로 결정될 수 있다.

[0026] 또한, S160 단계에서 테스트 장치는, 트레이드-오프가 발생하는 경우와 발생하지 않는 경우 값이 변경되는 변수를, 테스트 결과 분석을 위하여 측정할 모니터링 포인트로 결정한다. 모니터링 포인트란 테스트 결과에 영향을 미치는 값으로, 트레이드-오프가 테스트 결과에 어떻게 영향을 미치고 있는지를 분석하기 위해 사용될 수 있다. 일 실시예에 따라, 모니터링 포인트는 워크 플로우를 통해 도출될 수 있다. 예컨대, 모니터링 포인트는 트레이드-오프 발생 경로와 미발생 경로에 따라 변화되는 변수 및 위치가 될 수 있다. 도 7을 참조하면, 트레이드-오프 발생 경로에서 aggressive 변수가 1로 선언되고, 트레이드-오프 미 발생 경로에서 aggressive 변수가 0으로 선언되는 것을 알 수 있다. 그러므로 aggressive 변수가 1로 선언된 횟수를 모니터링 포인트로 결정하여 그 값을 모니터링 하면, 테스트 결과를 분석할 때 aggressive 의 변수가 1로 선언된 횟수에 따라 성능에 어떤 변화가 있는지를 분석할 수 있다. 즉, 모니터링 포인트를 이용하여 트레이드-오프 발생이 테스트 결과에 어떤 영향을 미치는지를 분석할 수 있다.

[0027] 이상으로 S110 단계 내지 S160 단계를 통하여 테스트 모델을 개발하는 과정을 설명하였다. 이후, S170 단계 내지 S190 단계에서는 개발된 테스트 모델을 기반으로 테스트 케이스를 설계하는 과정을 설명한다. 그러나, 테스트 케이스를 생성하기 위하여 반드시 테스트 모델을 개발해야 하는 것은 아니며, 실시예에 따라, 테스트 항목, 제1 아키텍처 결정사항 및 제1 성능 속성으로부터, 테스트 모델 없이 테스트 케이스가 생성될 수도 있다.

[0028] S170 단계에서 테스트 장치는, 트레이드-오프가 발생하는 경우와 발생하지 않는 경우가 모두 테스트에 포함될 수 있도록 테스트 케이스의 초기 조건을 결정한다. 초기 조건은 테스트 케이스의 입력이 들어오기 전의 시스템의 상태이다. 일 실시예에 따라 초기 조건은 워크 플로우의 분기 조건을 참조하여 도출될 수 있다. 이때, 테스트 모델의 워크 플로우 모델은 테스트 커버리지로 활용되어 테스트 케이스의 초기 조건을 결정하는데 기준이 된다. 예컨대, 도 5의 모든 경로를 커버하기 위하여 초기 조건은 '초기 여유 공간'으로 도출될 수 있다. 초기 여유 공간에 다양한 값이 선택되면, aggressive 변수가 1로 되는 경우와 0으로 되는 경우가 커버될 수 있기 때문이다. 초기 여유 공간이 될 수 있는 값의 범위는 NAND메모리의 사용률이 0%인 경우부터 100%까지 있을 수 있다. 본 실시예에서는, NAND메모리의 사용률을 10%씩 나누어 0%~90%를 초기 조건으로 도출하고, 트레이드-오프의 발생 경로인 여유 공간이 1280KB미만인 경우를 커버하기 위하여 NAND 메모리 사용률이 99%인 경우를 추가적으로 도출하였다.

[0029] S180 단계에서 테스트 장치는, 워크 플로우의 모든 경로를 포함하도록 입력 항목을 설정한다. 테스트 케이스의 입력은 초기 조건과 마찬가지로 워크 플로우의 모든 경로를 커버할 수 있도록 도출된다. 트레이드-오프 발생 경로와 미 발생 경로는 초기 조건으로 커버할 수 있도록 하였으므로, 테스트 케이스의 입력은 도 7의 나머지 경로를 모두 커버하도록 도출된다. 도 7의 워크 플로우에는 총 6가지 경로가 존재하며 이는 다음과 같다. Path1: open->truncate->write->fsync->close, Path2: open->truncate->write->close, Path3: open->lseek->write->fsync->close, Path4: open->lseek->write->close, Path5: create->write->fsync->close, Path6: create->write->close. 이때, 도 7의 경로 Path 1 내지 Path 6는, 파일을 오픈하기 위한 시스템 콜의 종류, truncate 시스템 콜의 호출 여부, lseek 시스템 콜의 호출 여부, lseek 함수를 호출 할 경우 파일 포인터의 위치, write 시스템 콜에 파라미터로 넘기기 위한 파일의 크기, fsync 시스템 콜의 호출 여부를 입력 항목으로 설정하면 모두 커버될 수 있다.

[0030] S190 단계에서 테스트 장치는, 초기 조건 및 입력 항목의 값에 따른 출력 항목의 기대 값을 결정한다. 테스트 케이스의 출력 항목은 S140 단계에서 결정된 항목을 사용할 수 있다. S140 단계에서 설명한 바와 같이, 기대 결과는 기능과 성능 두 가지 측면으로 나누어 도출되며, 이때 테스트 모델에서 도출된 기대 결과가 이용될 수 있다. 이와 같이 S110 단계 내지 S190 단계를 통하여 생성된 테스트 케이스가 도 5에 개시된다.

[0031] 도 5는 본 실시예에 따른 테스트 케이스 중 'path1'을 커버하기 위한 부분이다. 도 5에서는, 초기 조건인 '여유 공간'의 값을 변화시켜 테스트 커버리지인 워크 플로우의 트레이드-오프 발생 경로와 미 발생 경로를 모두 커버하도록 하였다. 또한, 'Path1'을 커버하기 위하여, 파일 open방법으로 open()시스템 콜을 이용하고 truncate()시스템 콜을 호출 한 뒤에 write() 시스템 콜을 호출하며, 파일 쓰기 이후 fsync()시스템 콜을 호출한다. 이와 같이 'Path1'을 커버할 수 있도록, 도 5의 테스트 케이스의 입력 항목의 값이 설정되었다. 한편, 도 5의 출력 항목의 기대 값을 살펴보면, 초기 조건인 초기 여유 공간의 값이 1%인 경우, 트레이드-오프가 발생할 것으로 예상되므로 파일 쓰기 속도가 1.8MB/sec 미만으로 설정되었음을 알 수 있다.

[0032] 도 8은 도 5의 테스트 케이스에 따른 테스트 결과를 그래프로 나타낸 도면이다. 개시된 기술에 따르면, 테스트

항목이 트레이드-오프를 지니고 있을 경우, 트레이드-오프 관계에 있는 또 다른 성능 속성도 성능 지표로 포함된다. 도 8의 왼쪽 그래프를 참조하면, 파일 쓰기 속도와 더불어 NAND 메모리에 새로 확보된 가용공간의 양을 측정함으로써, 가비지 콜렉션이 갖는 '시간 반응성'과 '자원 효율성' 간의 트레이드-오프를 확인할 수 있다. 보다 구체적으로, NAND 메모리 사용량이 90% 이하인 경우에는 파일 쓰기 속도가 1.8MB/sec 이상으로 측정되어 성능 요구사항을 만족시켰지만, NAND 메모리 사용량이 99%인 경우에 파일 쓰기 속도가 예상 파일 쓰기 속도보다 매우 낮게 측정되어 성능 요구사항을 만족시키지 않음을 볼 수 있다. 반면에, '자원 효율성' 측면에서 '새로 확보한 가용 공간'이 매우 증가한 것을 볼 수 있다. 기존의 성능 테스트는 테스트의 판단 기준으로 성능 요구사항을 만족하는 여부만을 사용하므로 이러한 경우, Fail이라고 결론 내릴 수 있다. 그러나, 도 8에서 확인할 수 있듯이 가비지 콜렉션이 갖는 트레이드-오프로 인하여 파일 쓰기 속도가 저하되더라도 자원효율성이 높게 측정되기 때문에, '파일 쓰기 속도'가 성능 요구사항을 만족시키지 못하더라도 테스트 결과가 Fail이라고 단정지을 수 없다. 즉, 한 가지의 성능을 평가하기 위하여 해당 성능 지표만으로 평가하면 안되며, 트레이드-오프가 있는 지를 분석하여 트레이드-오프에 해당하는 성능 속성도 성능 지표로 같이 활용하는 것이 정확한 성능 평가를 위하여 바람직하다.

[0033] 한편, 개시된 기술에서 새로이 제안한 모니터링 포인트는 테스트 결과의 분석에 이용될 수 있다. 테스트 결과와 함께 측정된 모니터링 포인트의 값은 도 8의 오른쪽 그래프와 같다. 모니터링 포인트의 그래프는 파일 쓰기 속도 저하가 자원 효율성 향상으로 인한 트레이드-오프 현상임을 뒷받침한다. 파일 쓰기 속도가 1.8MB/sec 이상으로 측정된 테스트 케이스들의 Aggressive Mode 진입횟수가 0인 것에 비해, 파일 쓰기 속도가 저하된 테스트 케이스만이 Aggressive Mode에 빈번하게 진입한 것을 볼 수 있다.

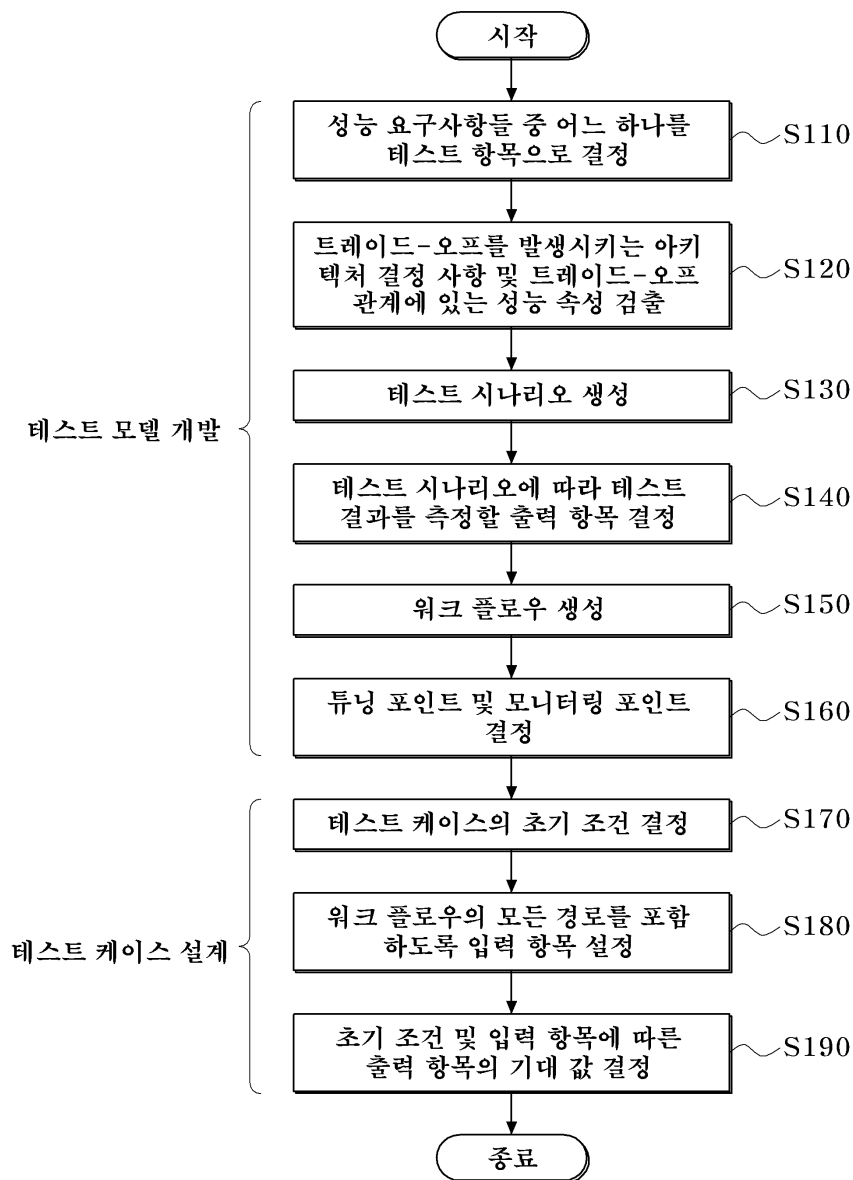
[0034] 도 9는 튜닝 전과 튜닝 후의 성능 변화를 나타낸 그래프이다. 도 4의 테스트 모델을 참조하면, 튜닝 포인트는 Aggressive Mode에 진입하는 시점이고, 여유 공간이 1280KB 미만으로 설정되어 있다. 도 9의 그래프에서는 튜닝 포인트의 효과를 분석하기 위하여, 튜닝 포인트를 여유 공간이 전체 NAND 용량의 25%보다 작을 경우로 튜닝하여 튜닝 전과의 성능을 비교하였다. Aggressive Mode에 진입하는 조건을 여유 공간이 25% 미만으로 튜닝 한 결과, NAND 메모리 사용량이 80% 이상인 경우에 파일 쓰기를 수행했을 때 파일 쓰기 속도가 저하되었다. 또한, NAND 메모리 사용량이 80%이상인 경우에 Aggressive Mode에 빈번하게 진입하지만 새로 확보한 가용 공간은 상승하지 않았다. 이를 통해, 튜닝 포인트의 변화가 성능을 변화시킬 수 있으며, 튜닝 후의 성능은 저하되었음을 확인할 수 있다. 따라서 튜닝 포인트의 값을 변경하면서 테스트를 반복하면, 소프트웨어의 성능을 향상시킬 수 있는 최적의 튜닝 포인트를 결정할 수 있다.

[0035] 한편, 트레이드-오프를 나타낸 확장 경로 없이 '파일 쓰기 속도'의 테스트 케이스를 도출한다면, 초기 조건으로 NAND 메모리 사용률을 간과하거나, 여유공간이 1280KB 미만이 되는 트레이드-오프의 상황을 커버하기 위한 테스트 케이스를 작성하기 어려울 것이다. 실제로도, 리눅스 파일 시스템의 성능 평가하기 위한 벤치마킹 툴들은 파일 쓰기 속도를 측정할 때, NAND 메모리의 사용률에 대해선 간과하고 있다. 그러므로, 트레이드-오프가 발생하는 경우와 발생하지 않는 경우 모두를 테스트 커버리지로 활용하여 테스트 케이스를 도출하면, 보다 다양한 상황에서의 성능을 평가할 수 있으므로 테스트의 완성도가 높아질 것으로 기대한다.

[0036] 이러한 개시된 기술인 시스템 및 장치는 이해를 돕기 위하여 도면에 도시된 실시 예를 참고로 설명되었으나, 이는 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 개시된 기술의 진정한 기술적 보호범위는 첨부된 특허청구범위에 의해 정해져야 할 것이다.

도면

도면1



도면2

품질속성	세부속성	평가항목
효율성	시간반응성	파일 읽기 속도
		파일 쓰기 속도
		마운트 속도
	자원효율성	블록 관리
		메모리 관리
신뢰성	오류허용성	에러 복구
		데이터 무결성
		동작 수명 (Wearleveling)
	회복성	에러 감지
		에러 점검
이식성	설치성	설치 용이
	적응성	이식 용이

도면3

성능 요구사항 : '파일 쓰기 속도'			
파일 시스템이 정상적으로 마운트된 환경에서 파일 쓰기 연산이 들어올 때, 1.8MB/sec을 보장해야 한다.			
아키텍처 결정사항	Trade off		
캐시	+	시간 반응성	파일 입출력 속도 향상
	-	신뢰성	전원 결함시 데이터 손실
가비지 콜렉션	+	자원 효율성	NAND의 여유 공간 확보
	-	시간 반응성	파일 쓰기 속도 저하

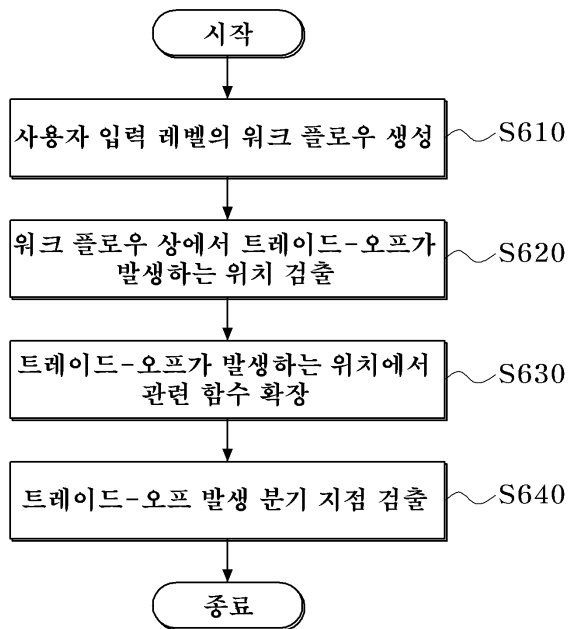
도면4

Test Modle										
품질 속성	세부 속성	테스트 항목	테스트 슈트	테스트 시나리오	기 능	Trade off (+)	Trade off (+)	Workflow Model	특징 포인트	모니터링 포인트
효율성	시간 반응성	파일 쓰기 속도	가비지 컬렉션	파일 쓰기를 수행할 때, 가비지 컬렉션을 수행하 여 여유공간을 확보하며, 1.8MB/sec를 보장해야 한다.	파 일 쓰 기 성 공	여 유 공 간 확 보	파 일 쓰 기 속 도 1.8MB /sec	그림2 참조	Aggressive Mode 진입 시점 :여유공간 1280KB 미만	Aggressive Mode 진입 횟수

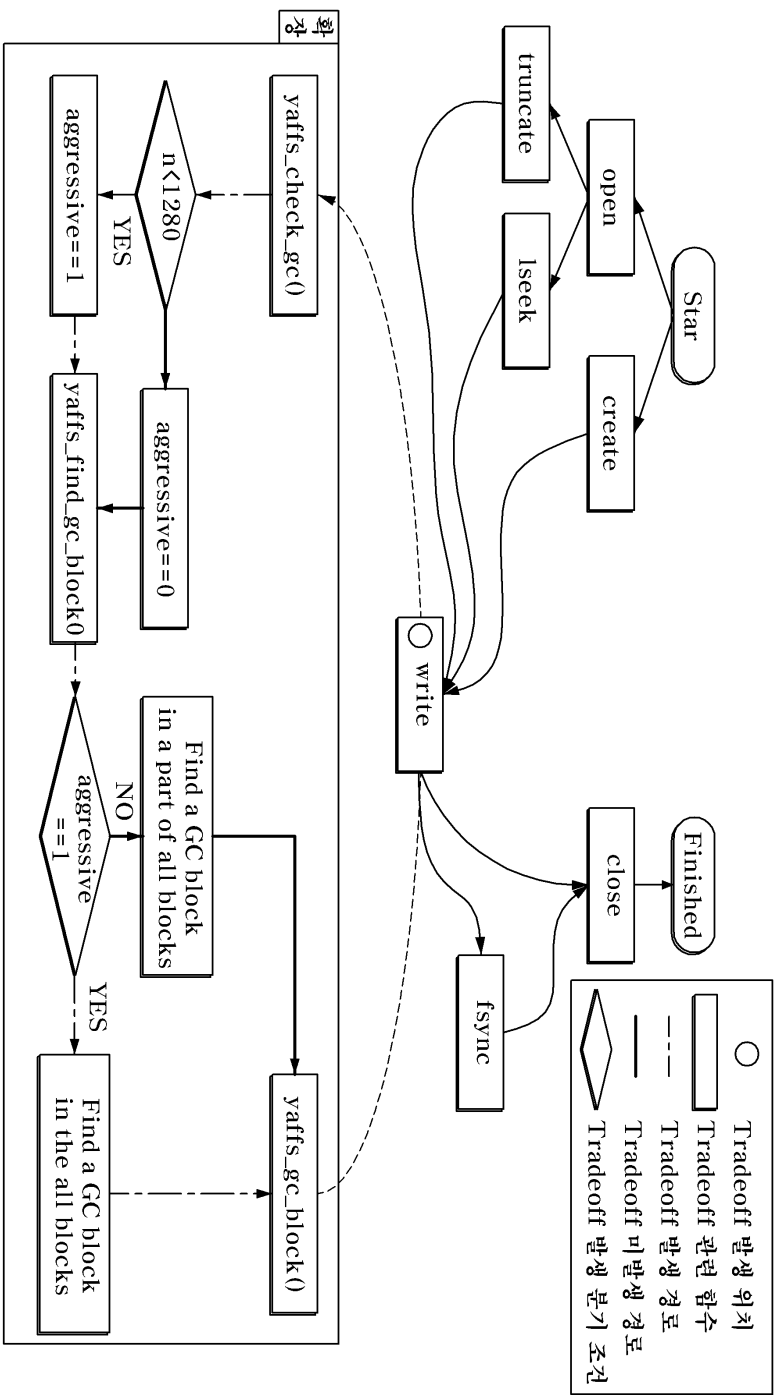
도면5

Test Case									
ID	초기조건		Input				Expected Output		
	여유공간	파일 open 방법	truncate	lseek 위치	파일 크기	동기화 여부	기능	Tradeoff(+) 새로 확보한 Free Chunk 개수	Tradeoff(-) 파일 쓰기 속도
DWE1	100%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE2	90%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE3	80%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE4	70%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE5	60%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE6	50%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE7	40%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE8	30%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE9	20%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE10	10%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 이상
DWE11	1%	open()	Y	N/A	1M	Y	파일 쓰기 성공	0개 이상	1.8MB/sec 미만

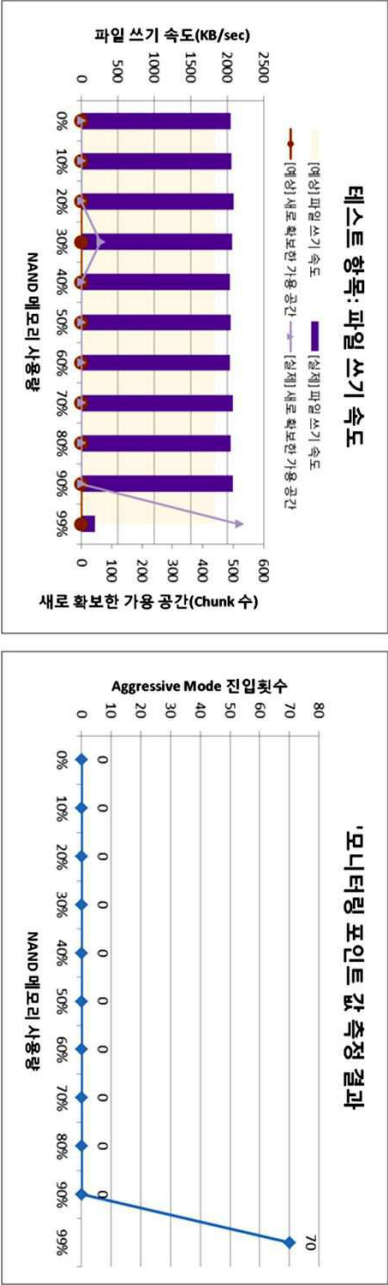
도면6



도면7



도면8



도면9

