(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0041539
(43) 공개일자 2024년04월01일

(51) 국제특허분류(Int. Cl.)
G01W 1/10 (2006.01) *G01W 1/18* (2006.01)
G06Q 50/26 (2024.01)
(52) CPC특허분류
G01W 1/10 (2013.01)
G01W 1/18 (2013.01)
(21) 출원번호 10-2022-0120670
(22) 출원일자 2022년09월23일
심사청구일자 2022년09월23일

(71) 출원인
국립창원대학교 산학협력단
경상남도 창원시 의창구 창원대학로 20(퇴촌동)
(72) 발명자
정성욱
경상남도 창원시 의창구 북면 무동로267번길 10, 205동 902호(창원무동 STX KAN 2차아파트)
이창현
경상남도 창원시 의창구 하남천서길37번길 12, 302호(도계동, 남성빌라)
(뒷면에 계속)
(74) 대리인
양성보

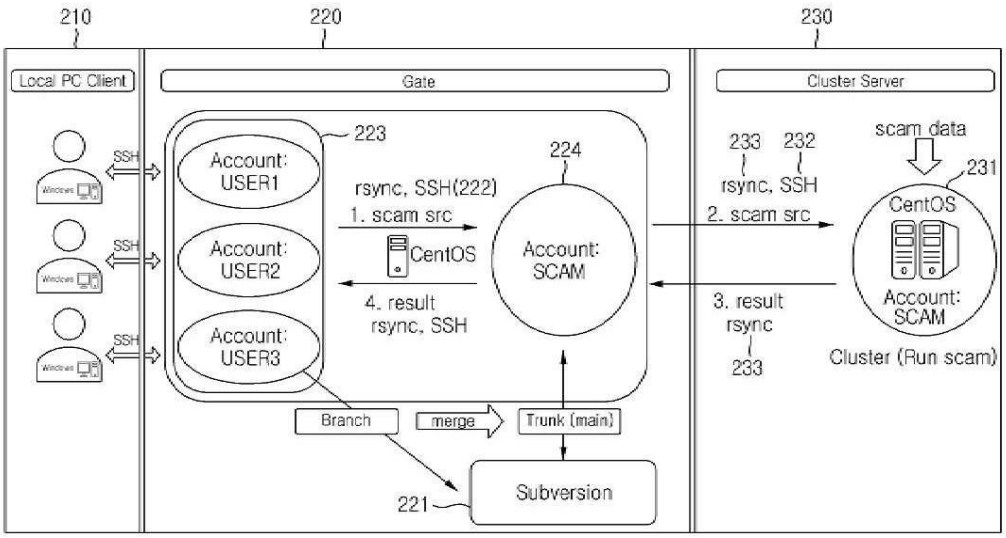
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 **적운 모수화 코드 향상을 위한 유니콘 최적화 방법 및 시스템**

(57) 요 약

적운 모수화 코드 향상을 위한 유니콘 최적화 방법 및 시스템이 제시된다. 본 발명에서 제안하는 SCAM 시스템은 원격 접속 프로그램을 이용하여 사용자가 게이트에 있는 자신의 사용자 계정에 접속하도록 인터페이스 역할을 수행하는 로컬 PC 클라이언트(Local PC Client), 사용자 계정 및 SCAM 계정을 통해 로컬 PC 클라이언트와 클러스터 서버를 연결하는 게이트(Gate) 및 복수의 노드를 포함하고, 게이트로부터 명령어를 입력 받아 서버의 배치 시스템(batch system)에 해당 명령어를 전달하여, SCAM 모델을 구동하고, 상기 구동 결과를 반환하여 게이트의 USER 계정에 전달하는 클러스터 서버(Cluster Server)를 포함한다.

대 표 도



(52) CPC특허분류
G06Q 50/26 (2024.01)
G01W 2201/00 (2013.01)
(72) 발명자
신대영
경상남도 창원시 의창구 읍성로 110(소답동)
조예린
경상남도 창원시 의창구 사림로16번길 35-26, 201호(사림동)
염기훈
경상남도 창원시 의창구 용동로57번안길 9-11, 502호(사림동)

이 발명을 지원한 국가연구개발사업

과제고유번호	1365003599
과제번호	KMI2021-01312
부처명	기상청
과제관리(전문)기관명	한국기상산업기술원
연구사업명	기후예측 및 위험 대응 강화 연구
연구과제명	차세대 전산과학 기술접합을 위한 기반 기술 개발
기 여 율	1/1
과제수행기관명	창원대학교 산학협력단
연구기간	2021.04.01 ~ 2023.12.31

명 세 서

청구범위

청구항 1

원격 접속 프로그램을 이용하여 사용자가 게이트에 있는 자신의 사용자 계정에 접속하도록 인터페이스 역할을 수행하는 로컬 PC 클라이언트(Local PC Client);

사용자 계정 및 SCAM 계정을 통해 로컬 PC 클라이언트와 클러스터 서버를 연결하는 게이트(Gate); 및

복수의 노드를 포함하고, 게이트로부터 명령어를 입력 받아 서버의 배치 시스템(batch system)에 해당 명령어를 전달하여, SCAM 모델을 구동하고, 상기 구동 결과를 반환하여 게이트의 USER 계정에 전달하는 클러스터 서버(Cluster Server)

를 포함하는 SCAM 시스템.

청구항 2

제1항에 있어서,

상기 로컬 PC 클라이언트는,

상기 게이트의 SCAM 계정의 지정된 디렉터리를 동기화시키기 위한 명령어를 통해 클러스터 서버로부터 결과 데이터를 받아 테스트와 검증을 수행하는

SCAM 시스템.

청구항 3

제1항에 있어서,

상기 게이트는,

사용자 계정의 경우 각각의 로컬 PC 클라이언트의 사용자가 자신의 사용자 계정에 접속하여 상기 로컬 PC 클라이언트에서의 활동들을 수행하기 위한 홈 환경을 제공하고, 각 사용자 계정은 자신의 해당 SVN 브랜치 branch)와 연동되는

SCAM 시스템.

청구항 4

제1항에 있어서,

상기 게이트는,

SCAM 계정의 경우 게이트의 사용자 계정과 클러스터 서버의 SCAM 계정 사이에서 데이터와 명령어를 전달하는 기능을 수행하며, SVN(Subversion)의 메인 스트림(main stream)과 연동되는

SCAM 시스템.

청구항 5

제1항에 있어서,

상기 게이트는,

각각의 사용자 계정이 각 소스 코드의 분기점마다 브랜치를 만들어 분기점에 대해서만 소스 코드를 테스트하고, 테스트가 완료된 후, SCAM 계정의 트런크와 병합하는

SCAM 시스템.

청구항 6

제1항에 있어서,

상기 게이트는,

사용자 계정 및 SCAM 계정에 소스 코드 정보를 저장하는 코드 디렉터리와 적운모수화 유니콘 소스 코드를 포함하는 모드 디렉터리가 존재하고,

상기 클러스터 서버는,

SCAM 계정에 SCAM의 입력 데이터에 해당하는 cesm, data_I3D, data_IOP, data_LES 디렉터리가 존재하는

SCAM 시스템.

청구항 7

로컬 PC 클라이언트가 원격 접속 프로그램을 이용하여 게이트의 사용자 계정에 접속하고, SCAM 모델이 구동되면 게이트의 사용자 계정으로부터 SCAM 계정으로 전환하는 단계;

게이트가 게이트의 SCAM 계정을 통해 게이트의 사용자 계정에 있는 SCAM 소스 코드를 가져오고, 클러스터 서버의 SCAM 계정으로 상기 SCAM 소스 코드를 전달하는 단계; 및

클러스터 서버가 클러스터 서버의 SCAM 계정에 SCAM 모델 수행을 요청하고, 상기 SCAM 모델이 구동되는 단계

를 포함하는 SCAM 시스템의 동작 방법.

청구항 8

제7항에 있어서,

상기 클러스터 서버가 클러스터 서버의 SCAM 계정에 SCAM 모델 수행을 요청하고, 상기 SCAM 모델이 구동되는 단계는,

상기 클러스터 서버가 전달 받은 SCAM 소스 코드를 컴파일 한 후 상기 SCAM 모델이 구동되면 각각 해당하는 컴파일 로그(compile log)와 러닝 로그(running log)를 남기는

SCAM 시스템의 동작 방법.

청구항 9

제8항에 있어서,

상기 SCAM 모델 구동이 모두 종료되면, 특정 확장자를 가진 모델의 결과가 저장되고, 게이트의 SCAM 계정을 거쳐 사용자 계정으로 결과를 전달하는

SCAM 시스템의 동작 방법.

청구항 10

제9항에 있어서,

상기 로컬 PC 클라이언트에서 원격 접속 프로그램을 통해 상기 상기 SCAM 모델 구동 결과를 실시간으로 모니터링하는

SCAM 시스템의 동작 방법.

청구항 11

원격 접속 프로그램을 이용하여 사용자가 게이트에 있는 자신의 사용자 계정에 접속하도록 인터페이스 역할을 수행하는 로컬 PC 클라이언트(Local PC Client); 사용자 계정 및 SCAM 계정을 통해 로컬 PC 클라이언트와 클러스터 서버를 연결하는 게이트(Gate); 및 복수의 노드를 포함하고, 게이트로부터 명령어를 입력 받아 서버의 배치 시스템(batch system)에 해당 명령어를 전달하여, SCAM 모델을 구동하고, 상기 구동 결과를 반환하여 게이트의 USER 계정에 전달하는 클러스터 서버(Cluster Server)를 포함하는 SCAM 시스템의 적운 모수화 코드 최적화

방법에 있어서,

루프 벡터화(Loop Vectorization) 방법, 종속 벡터화(Dependency Vectorization) 방법 및 코드 모던화(Code Modernization) 방법을 모두 이용하여 스칼라(Scalar) 연산을 벡터화(Vectorization)는

적운 모수화 코드 최적화 방법.

청구항 12

제11항에 있어서,

상기 루프 벡터화 방법은,

루프 연산에서 루프가 중첩될 경우 두 루프 사이의 식을 재구성하여 중첩된 루프를 제거하는

적운 모수화 코드 최적화 방법.

청구항 13

제11항에 있어서,

상기 종속 벡터화 방법은,

하나의 반복문 안에 복수의 종속되지 않는 식이 존재하는 경우 각각의 식을 서로 다른 루프로 분리화하여 벡터화하는

적운 모수화 코드 최적화 방법.

청구항 14

제11항에 있어서,

상기 코드 모던화 방법은,

소스 코드에 있어서 두 변수가 서로 같음을 표시하는 방법을 변경하고, 예약된 약어를 모두 대문자로 변경하며, 한 라인에 작성된 코드가 미리 정해진 길이 이상일 경우 복수의 라인으로 분리하고, 변수 및 프로시저의 변수명은 소문자 알파벳으로 시작하고 복수의 의미가 중첩될 경우 중첩된 의미를 구분하는

적운 모수화 코드 최적화 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 적운 모수화 코드 향상을 위한 유니콘 최적화 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 하드웨어 기술이 발달하여 고성능의 슈퍼 컴퓨팅이 가능해지고 과학적으로 수치 모델 방식이 고도화됨에 따라 더욱 정밀한 기상예보를 진행할 수 있게 되었다. 그래서 각국의 기상청에서는 고성능의 슈퍼컴퓨터에 기상모델(Climate Model)을 이식하여 기상예보를 수행하고 있다. 그리고 한국 기상청(Korea Meteorological Administration; KMA)은 기상 예보를 위한 선진기술의 도입과 기상 전문인력 양상을 위해 미국의 국립대기 과학연구소(NCAR, National Center for Atmospheric Research)와 2004년에 기술협력(2004.03.22)을 체결하였고 전 지구 기상예보를 위해 2012년 영국 기상청(met office)과 기상 관련 공동운영 협약 (2012.06.26.)을 맺어 전 지구 기상모델(Global Climate Model; GCM)에 관한 자료를 공유하기로 하였으며, 더 나아가 2014년에 영국 기상청으로부터 Glosea5(Global Seasonal Forecast System)이라고 하는 전지구 기상모델을 슈퍼컴퓨터에 도입하여 현재(2022)까지 기상예보를 수행하고 있다.

[0003] 여기서, 한국 기상청은 실제 기상예보를 수행할 때, 한국이 자체적으로 개발한 기후 모델인 KIM(Korean Integrated Model) 모델과 영국에서 도입한 Glosea5를 병행하여 사용하고 있으며, 전 지구 모델인 Glosea5는 대기 모델(UM), 해양 모델(NEMO), 해 빙 모델(CICE) 그리고 지면 모델(JULES) 4가지로 구성된다. 그런데, Glosea5가 대기 모델을 수행할 때, 적운모수화(Cumulus Parameterizations)라고 불리는 기상현상에 대한 시뮬레

이션이 미흡하여 개선할 필요가 있다. 그래서 Glosea5의 새로운 버전인 Glosea6에서 해당 부분을 개선했으며, 더 나아가 미국의 전 지구 모델 CESM의 간소화 버전인 SCAM 모델에 포함된 적운모수화 코드(Unicon, Fortran)를 Glosea6에 적용할 필요가 있다. 따라서 이를 수행하기 전에 Unicon 소스 코드를 분석하고 그에 대한 테스트 및 검증할 필요가 있으며, 원활한 코드의 실행을 위해 소스 코드 최적화 기법을 통한 소스 코드 성능을 향상할 필요가 있다. CESM(Community Earth System Mode)은 대기, 해양, 해빙 그리고 지면 4가지의 모델과 기타 구성 요소로 구성된 전 지구 기상모델 시스템으로 미국의 국립대기 연구 센터(US National Center for Atomospheric Reserch; NCAR)에서 개발되었다. CESM은 전 지구 기상모델로, 과거, 현재 그리고 미래에 대한 기후 변화를 4개의 모델(대기, 해양, 해빙, 지면) 관점에서 전 지구를 일정한 격자(수평, 수직) 크기로 구분하여, 수치 모델링을 통해 격자들을 미분 및 적분하여 얻은 결과를 통해 기상을 시뮬레이션하는 기후 모델이다.

[0004] SCAM(Single-Columns Atmospheric Model)은 CESM의 간소화 버전으로 주로, 특정 기상 연구를 위해 사용된다. SCAM은 4가지 모델 모두 고려하는 CESM과 달리 오직 대기 모델(Atmosphric)에만 초점을 맞추어 수행되며, 오직 단일 격자(Single-Columns)의 적분을 통해 기상을 시뮬레이션하는 기후 모델이다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) [1] Kay, Jennifer E, et al., "The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability," Bulletin of the American Meteorological Society, p.2015, 1333-1349

발명의 내용

해결하려는 과제

[0006] 본 발명이 이루고자 하는 기술적 과제는 CESM(Community Earth System Mode)의 간소화 버전인 SCAM(Single-Columns Atmospheric Model)에 포함된 적운모수화 코드(Unicon, Fortran)를 최적화하고 유지보수성을 증가시키기 위해 루프 벡터화(Loop Vectorization), 종속 벡터화(Dependency Vectorization), 코드 모던화(Code Modernization) 3가지가 결합한 유니콘 최적화(Unicon Optimization) 기법을 제안하고 이를 테스트 하기 위한 SCAM 전체 실행 구조도를 제공하는데 있다.

과제의 해결 수단

[0007] 일 측면에 있어서, 본 발명에서 제안하는 SCAM 시스템은 원격 접속 프로그램을 이용하여 사용자가 게이트에 있는 자신의 사용자 계정에 접속하도록 인터페이스 역할을 수행하는 로컬 PC 클라이언트(Local PC Client), 사용자 계정 및 SCAM 계정을 통해 로컬 PC 클라이언트와 클러스터 서버를 연결하는 게이트(Gate) 및 복수의 노드를 포함하고, 게이트로부터 명령어를 입력 받아 서버의 배치 시스템(batch system)에 해당 명령어를 전달하여, SCAM 모델을 구동하고, 상기 구동 결과를 반환하여 게이트의 USER 계정에 전달하는 클러스터 서버(Cluster Server)를 포함한다.

[0008] 상기 로컬 PC 클라이언트는 상기 게이트의 SCAM 계정의 지정된 디렉터리를 동기화시키기 위한 명령어를 통해 클러스터 서버로부터 결과 데이터를 받아 테스트와 검증을 수행한다.

[0009] 상기 게이트는 사용자 계정의 경우 각각의 로컬 PC 클라이언트의 사용자가 자신의 사용자 계정에 접속하여 상기 로컬 PC 클라이언트에서의 활동들을 수행하기 위한 홈 환경을 제공하고, 각 사용자 계정은 자신의 해당 SVN 브랜치 branch)와 연동된다.

[0010] 상기 게이트는 SCAM 계정의 경우 게이트의 사용자 계정과 클러스터 서버의 SCAM 계정 사이에서 데이터와 명령어를 전달하는 기능을 수행하며, SVN(Subversion)의 메인 스트림(main stream)과 연동된다.

[0011] 상기 게이트는 각각의 사용자 계정이 각 소스 코드의 분기점마다 브랜치를 만들어 분기점에 대해서만 소스 코드를 테스트하고, 테스트가 완료된 후, SCAM 계정의 트런크와 병합한다.

[0012] 상기 게이트는 사용자 계정 및 SCAM 계정에 소스 코드 정보를 저장하는 코드 디렉터리와 적운모수화 유니콘 소스 코드를 포함하는 모드 디렉터리가 존재하고, 상기 클러스터 서버는 SCAM 계정에 SCAM의 입력 데이터에 해당

하는 cesm, data_I3D, data_IOP, data_LES 디렉터리가 존재한다.

[0013] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 SCAM 시스템의 동작 방법은 로컬 PC 클라이언트가 원격 접속 프로그램을 이용하여 게이트의 사용자 계정에 접속하고, SCAM 모델이 구동되면 게이트의 사용자 계정으로부터 SCAM 계정으로 전환하는 단계, 게이트가 게이트의 SCAM 계정을 통해 게이트의 사용자 계정에 있는 SCAM 소스 코드를 가져오고, 클러스터 서버의 SCAM 계정으로 상기 SCAM 소스 코드를 전달하는 단계 및 클러스터 서버가 클러스터 서버의 SCAM 계정에 SCAM 모델 수행을 요청하고, 상기 SCAM 모델이 구동되는 단계를 포함한다.

[0014] 본 발명의 실시예에 따르면, 클러스터 서버가 클러스터 서버의 SCAM 계정에 SCAM 모델 수행을 요청하고, 상기 SCAM 모델이 구동되는 단계는 상기 클러스터 서버가 전달 받은 SCAM 소스 코드를 컴파일 한 후 상기 SCAM 모델이 구동되면 각각 해당하는 컴파일 로그(compile log)와 러닝 로그(running log)를 남긴다.

[0015] 본 발명의 실시예에 따르면, SCAM 모델 구동이 모두 종료되면, 특정 확장자를 가진 모델의 결과가 저장되고, 게이트의 SCAM 계정을 거쳐 사용자 계정으로 결과를 전달한다.

[0016] 본 발명의 실시예에 따르면, 로컬 PC 클라이언트에서 원격 접속 프로그램을 통해 상기 상기 SCAM 모델 구동 결과를 실시간으로 모니터링한다.

[0017] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 SCAM 시스템의 적운 모수화 코드 최적화 방법은 루프 벡터화(Loop Vectorization) 방법, 종속 벡터화(Dependency Vectorization) 방법 및 코드 모던화(Code Modernization) 방법을 모두 이용하여 스칼라(Scalar) 연산을 벡터화(Vectorization)한다.

발명의 효과

[0018] 본 발명의 실시예들에 따르면 루프 벡터화(Loop Vectorization), 종속 벡터화(Dependency Vectorization), 코드 모던화(Code Modernization) 3가지가 결합한 유니콘 최적화(Unicon Optimization) 기법을 통해 CESM(Community Earth System Mode)의 간소화 버전인 SCAM(Single-Columns Atmospheric Model)에 포함된 적운 모수화 코드(Unicon, Fortran)를 최적화하고 유지보수성을 증가시킬 수 있다. 본 발명의 실시예에 따라 구축한 SCAM 실행 환경에서 제안한 유니콘 최적화 기법을 테스트 하였고 기존 소스 코드 대비 루프 벡터화는 3.086% 종속 벡터화는 0.4572% 성능이 향상됨을 확인하였다. 그리고 이를 모두 적용한 유니콘 최적화 기법의 경우 기존 소스 코드 대비 3.457%의 성능이 향상됨을 확인하였다.

도면의 간단한 설명

[0019] 도 1은 본 발명의 일 실시예에 따른 SCAM 시스템 디렉터리 구조도를 나타내는 도면이다.

도 2는 본 발명의 일 실시예에 따른 SCAM 전체 시스템 구조도를 나타내는 도면이다.

도 3은 본 발명의 일 실시예에 따른 SCAM 전체 실행 과정을 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 루프 벡터화 의사코드를 나타내는 도면이다.

도 5는 본 발명의 일 실시예에 따른 루프 벡터화 적용의 예시를 나타내는 도면이다.

도 6은 본 발명의 일 실시예에 따른 종속 벡터화 의사코드를 나타내는 도면이다.

도 7은 본 발명의 일 실시예에 따른 종속 벡터화 적용의 예시를 나타내는 도면이다.

도 8은 본 발명의 일 실시예에 따른 코드 모던화 적용의 예시를 나타내는 도면이다.

도 9는 본 발명의 일 실시예에 따른 유니콘 최적화 성능 평가를 나타내는 그래프이다.

도 10은 본 발명의 일 실시예에 따른 루프 벡터화 성능 평가를 나타내는 그래프이다.

도 11은 본 발명의 일 실시예에 따른 종속 벡터화 성능 평가를 나타내는 그래프이다.

도 12는 본 발명의 일 실시예에 따른 유니콘 벡터화 성능 평가를 나타내는 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0020] 하드웨어 기술이 발달하고 수치 모델 방식이 고도화됨에 따라 더욱 정밀한 기상예보를 진행할 수 있게 되었다. 본 발명에서는 CESM(Community Earth System Mode)의 간소화 버전인 SCAM(Single-Columns Atmospheric Model)

에 포함된 적운모수화 코드(Unicon, Fortran)를 최적화하고 유지보수성을 증가시키기 위해 루프 벡터화(Loop Vectorization), 종속 벡터화(Dependency Vectorization), 코드 모던화(Code Modernization) 3가지가 결합한 유니콘 최적화(Unicon Optimization) 기법을 제안하고 이를 테스트 하기 위한 SCAM 전체 실행 구조도를 제시하였다. 본 발명의 실시예에 따라 구축한 SCAM 실행 환경에서 제안한 유니콘 최적화 기법을 테스트 하였고 기존 소스 코드 대비 루프 벡터화는 3.086% 종속 벡터화는 0.4572% 성능이 향상됨을 확인하였다. 그리고 이를 모두 적용한 유니콘 최적화 기법의 경우 기존 소스 코드 대비 3.457%의 성능이 향상됨을 확인하였다. 이는 본 발명에서 제안한 유니콘 최적화 기법이 우수한 성능을 제공하고 있음을 입증한다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0022] 유니콘(Unicon) 적운모수화(Cumulus Parameterization)는 CESM, SCAM의 대기 모델을 구성하는 전체 코드의 일부이며, 대기의 기상 현상 중 하나인 적운모수화를 측정하기 위해 사용된다. 본 발명에서는 현재 CESM에서 사용하고 있는 유니콘 적운모수화 소스 코드를 한국 기상청에서 전 지구 기상예보를 위해 사용하고 있는 Glosea6에 적용하기 전에 유니콘 소스 코드의 이식성을 테스트하기 위해서 소스 코드를 실행할 수 있는 SCAM의 실행환경을 구축하고 이를 분석한다. 그 후, 본 발명에서 제안하는 최적화 기법을 통해 적운모수화 소스 코드를 최적화하여 이에 대한 이식성 검증을 진행하는 것이 주요 목적이다.

[0024] 도 1은 본 발명의 일 실시예에 따른 SCAM 시스템 디렉터리 구조도를 나타내는 도면이다.

[0025] SCAM 디렉터리 구조는 도 1과 같이 SCAM의 입력 데이터(110)와 관련된 cesm(111), data_I3D(112), data_IOP(113), data_LES 디렉터리(114), 실제 SCAM의 전체 소스 코드가 들어있는 codes 디렉터리(120), codes 디렉터리에서 일부 수정이 필요한 코드를 가져와 자신의 상황에 맡게 변경한 mods 디렉터리(130)(적운모수화 Unicon이 포함되어 있음), SCAM의 실행 스크립트가 들어있는 csh(c-shell) 디렉터리(140), 모델이 수행된 모든 결과(목적 코드, 실행파일, 실행로그, 결과파일)가 저장되는 runs 디렉터리(150)가 있으며 runs 디렉터리의 로그들(compile, running, both time)을 링크시켜 각각의 로그들에 대한 참조를 쉽게 만든 log 디렉터리(160)와 SCAM 실행에 대한 결과물을 참조하는 output 디렉터리(170)로 구성된다.

[0027] 도 2는 본 발명의 일 실시예에 따른 SCAM 전체 시스템 구조도를 나타내는 도면이다.

[0028] 적운모수화 코드인 유니콘을 테스트하기 위해 두 가지 방법으로 SCAM 디렉터리 구조를 재구축하였다. 첫 번째는 소스 코드를 원활히 관리하기 위해 소프트웨어 형상 관리 도구 중 하나인 SVN(Subversion)(221)을 활용 하여 구축하였고 두 번째는 실제로 모델이 수행되는 클러스터 서버(Cluster Server)와 모델 수행을 요청하는 사용자(User)를 rsync 와 SSH(222)를 이용해 분리하였다. 따라서 위의 두 가지 방법을 이용하여 구축된 SCAM 테스트 환경은 도 2와 같이 로컬 PC 클라이언트(Local PC Client)(210), 게이트(Gate)(220), 클러스터 서버(Cluster Server)(230)의 형태로 구성된다.

[0029] 로컬 PC 클라이언트(210)는 도 2의 SCAM 전체 시스템 구성에서 사용자 단말에 속하는 부분이다. 그리고 로컬 PC 클라이언트(210)의 주요 기능은 SSH 프로토콜을 사용하는 원격 접속 프로그램(예를 들어, putty, xshell, MobaXterm 등)을 이용해 사용자가 게이트(220)에 있는 자신의 USER 계정에 접속하기 위한 인터페이스 역할을 한다. 이를 통해 사용자는 SCAM의 소스 코드를 분석 및 수정하여 SCAM의 컴파일 및 실행을 요청하고 그에 대한 모니터링을 수행한다. 그리고 지정된 디렉터리를 동기화시키는 linux의 rysnc라는 명령어를 통해 클러스터 서버(230)로부터 결과 데이터를 받아 이에 대한 테스트와 검증을 할 수 있다. 로컬 PC 클라이언트(210)를 구축하기 위해서는 SSH를 지원하는 원격 접속 프로그램을 다운로드(본 발명의 실시예에서는 MobaXterm을 사용하여 진행하였음)하여 게이트(220)의 USER & Host 정보를 가져와 SSH 설정을 수행하기만 하면 된다.

[0030] 게이트(220)는 도 2의 SCAM 전체 시스템 구성에서 사용자와 클러스터 서버(230) 사이를 연결해주는 부분이다. 도 2를 참조하면, 게이트(220)의 계정은 USER 계정(223)과 SCAM 계정(224)으로 구분되며, 게이트(220)의 기능은 계정별로 별도의 다른 기능을 제공한다. 게이트(220)의 USER 계정(223)의 경우 각각의 로컬 PC 클라이언트(210)의 사용자가 자신의 USER 계정에 맞게 접속하여 앞서 로컬 PC 클라이언트(210)에서 언급한 다양한 활동들을 할 수 있는 홈(Home) 환경을 제공하며, 각자 자신의 SVN 브랜치(branch)와 연동되어 있다. 게이트(220)의 SCAM 계정(224)의 경우 게이트(220)의 USER 계정(223)과 클러스터 서버(230)의 SCAM 계정(231) 사이에서 데이터와 명령어를 전달하는 기능을 수행하며, SVN의 트런크(trunk(다시 말해, 메인 스트림(main stream)))와 연동되어 있

다. 이를 통해 각각의 사용자는 각 소스 코드의 분기점마다 브랜치를 만들어 분기점에 대해서만 소스 코드를 테스트할 수 있으며, 테스트가 완료되고 나면 SCAM 계정의 트런크와 병합할 수 있다. 게이트(220)를 구축하기 위해서는 게이트(220)의 USER 계정(223)과 SCAM 계정(224)을 양방향으로 SSH와 rsync(222)를 연결하고 게이트(220)의 SCAM 계정(224)과 클러스터 서버(230)의 SCAM 계정(231)과 단방향으로 SSH(232)를 연결하고 양방향으로 rsync(233)를 연결한다.

[0031] 클러스터 서버(230)는 도 2의 SCAM 전체 시스템 구성에서 실제로 SCAM을 실행하는 부분이다. 클러스터 서버(230)는 여러 노드로 구성되어 있으며, 게이트(220)로부터 명령을 지시 받아 서버의 배치 시스템(batch system)에 해당 명령을 전달하여, SCAM 모델을 구동하고 그에 관한 결과를 반환하여 게이트(220)의 USER 계정에 넘겨주는 역할을 가진다. 클러스터 서버(230)를 구축하기 위해서는 실제로 모델을 구동하기 위한 SCAM의 다양한 종속 소프트웨어(Dependency Software)를 설치해야 하며, 해당하는 소프트웨어는 표 1과 같다.

[0032] <표 1>

Name	Version
Linux(CentOS7)	7.6.1810
Intell oneAPI HPC (ifort)	2021.2.0
GNU compiler (gcc, g++)	2019.01.5
cmake	2.8.12.2
make	3.8.2
gmake	3.8.2
mpich	3.1.4
hdf5	1.10.3
netcdf-c	4.3.3.1
netcdf-fortran	4.4.2
nco	4.7.5
ncl-ncar	6.6.2

[0033]

[0034] 그리고 도 1의 디렉터리 구조에서 게이트의 USER 및 SCAM 계정에는 소스 코드 정보를 저장하는 codes 디렉터리와 적운모수화 Unicon 소스 코드가 있는 mod 디렉터리만이 존재하며, 클러스터 서버(230)의 SCAM 계정에는 SCAM의 입력 데이터에 해당하는 cesm, data_I3D, data_IOP, data_LES 디렉터리가 존재한다.

[0036] 도 3은 본 발명의 일 실시예에 따른 SCAM 전체 실행 과정을 설명하기 위한 도면이다.

[0037] 본 발명의 실시예에 따른 SCAM 시스템의 동작 방법은 SCAM 로컬 PC 클라이언트가 원격 접속 프로그램을 이용하여 게이트의 사용자 계정에 접속하고, SCAM 모델이 구동되면 게이트의 사용자 계정으로부터 SCAM 계정으로 전환하는 단계(341), 게이트가 게이트의 SCAM 계정을 통해 게이트의 사용자 계정에 있는 SCAM 소스 코드를 가져오고, 클러스터 서버의 SCAM 계정으로 상기 SCAM 소스 코드를 전달하는 단계(342) 및 클러스터 서버가 클러스터 서버의 SCAM 계정에 SCAM 모델 수행을 요청하고, 상기 SCAM 모델이 구동되는 단계(343)를 포함한다.

[0038] SCAM의 모델 전체 수행 과정은 도 3과 같이 먼저 단계(341)에서, 로컬 PC 클라이언트(310)에서 SSH 원격 접속 프로그램(311)을 이용하여 게이트(320)의 USER 계정(321)에 접속한다.

[0039] 그리고 단계(342)에서, 사용자가 SCAM 모델을 실행하게 되면 게이트(320)의 USER 계정(321)으로부터 SCAM계정(322)으로 전환하게 된다. 이후, 게이트(320)의 SCAM 계정(322)은 게이트(320)의 USER 계정(321)에 있는 SCAM 소스 코드(323)를 가져가 클러스터 서버(330)의 SCAM 계정(331)으로 소스 코드(332)를 전달하게 된다.

[0040] 소스 코드 전달이 완료 되고 나면, 단계(343)에서 클러스터 서버(330)의 SCAM 계정(331)에 SCAM 모델 수행을 요청(333)하게 되고 클러스터 서버(330)의 SCAM 계정(331)은 서버의 배치 시스템(Batch System)(pbs)에 SCAM Running Job을 전달하게 된다. 그 후, 배치 시스템의 Job scheduler(torque)를 거쳐 모델이 실행(334)된다. 여기서, SCAM 모델의 실행(334)은 전달받은 SCAM의 소스 코드(332)를 컴파일 후(모델을 실행하면 새로 컴파일이 진행한다) 모델이 실행(run)하게 되면 각각 해당하는 컴파일 로그(compile log)와 러닝 로그(running log)를 남기게 된다. 그리고 모델이 모두 정상적으로 종료가 되고 나면 nc 확장자를 가진 모델의 결과(335)가 저장되며 rynch를 이용하여 게이트(320)의 SCAM 계정(322)을 거쳐 USER 계정(321)으로 결과를 전달(324)하게 된다. 또한, 이러한 과정들은 로컬 PC 클라이언트(310)에서 원격 접속 프로그램을 통해 실시간으로 모니터링을 할 수 있다.

[0041] 본 발명의 실시예에 따른 벡터화 최적화 방법을 구체적으로 제안하고 실제로 앞서 소개한 SCAM의 대기 모델 중 하나인 적운모수화 유니콘 소스 코드에 적용한다.

[0042] 본 발명의 실시예에 따른 SCAM 시스템의 적운 모수화 코드 최적화 방법인 유니콘 최적화 기법은 총 3가지의 방법이 적용되며 소스 코드의 성능 향상을 위해 스칼라(Scalar) 연산을 벡터화(Vectorization)하여 병렬성을 증진시키고 유지보수성을 향상하기 위하여 작명법과 표기법을 최신화한다. 본 발명에서는 유니콘 적운모수화코드를 최적화 할 때, 성능 향상을 위하여 루프 벡터화(Loop Vectorization) 방법과 종속 벡터화(Dependency Vectorization) 방법을 제안하며, 소스 코드의 원활한 유지 보수를 위해 코드 모던화(Code Modernization) 방법을 제안한다. 그리고 본 발명에서 제안한 유니콘 최적화 기법은 상기 3가지의 방법이 모두 함께 적용되어 동작하게 된다.

[0044] 도 4는 본 발명의 일 실시예에 따른 루프 벡터화 의사코드를 나타내는 도면이다.

[0045] 루프 벡터화 방법은 루프 연산에 있어서 프로세서 수준의 기술인 벡터화(Vectorization)를 사용하여 소스 코드의 성능 향상을 끌어내는 기법이다.

[0046] 도 4에는 오리지널 소스 코드(610)와 제안하는 루프 벡터화 방법에 따른 따른 소스 코드(620)를 도시하였다. 도 4와 같이 루프의 벡터화는 반드시 가장 안에 감 싸여져 있는 루프만을 벡터화할 수 있으므로 루프가 중첩될 경우, 두 루프 사이의 식을 재구성하여 중첩된 루프를 제거한다.

[0047] 도 5는 본 발명의 일 실시예에 따른 루프 벡터화 적용의 예시를 나타내는 도면이다.

[0048] 도 5는 도 4의 루프 벡터화가 실제 적용된 유니콘 소스 코드의 일부를 나타내는 도면이다.

[0050] 도 6은 본 발명의 일 실시예에 따른 종속 벡터화 의사코드를 나타내는 도면이다.

[0051] 도 6에는 오리지널 소스 코드(610)와 제안하는 루프 벡터화 방법에 따른 따른 소스 코드(620)를 도시하였다.

[0052] 종속 벡터화는 도 6과 같이 하나의 반복문 안에 여러 개의 종속되지 않는 식이 존재하면 각각의 식을 서로 다른 루프로 분리화하여 벡터화를 한다.

[0053] 도 7은 본 발명의 일 실시예에 따른 종속 벡터화 적용의 예시를 나타내는 도면이다.

[0054] 도 7은 도 5의 종속 벡터화가 실제 적용된 유니콘 소스 코드의 일부를 나타내는 도면이다.

[0056] 도 8은 본 발명의 일 실시예에 따른 코드 모던화 적용의 예시를 나타내는 도면이다.

[0057] 코드 모던화는 성능 향상을 목적으로 하는 루프 벡터화 및 종속 벡터화와는 달리 원활한 유지 보수를 위하여 포트란(Fortran) 소스 코드의 작성방식을 균일화시켜 유지보수성을 증가시키는 방법이다. 소스 코드의 원활한 유지 보수를 위해 제안하는 작성방식은 두 변수가 서로 같음을 표시하는 방법을 '.eq.'에서 '==로 변경하고 예약된 약어(IF, ELSE, DO) 등은 모두 대문자로 변경하며, 한 라인에 너무 길게 작성된 코드는 여러 라인으로 분리한다. 그리고 변수 및 프로시저의 변수명일 때 가능한 소문자 알파벳으로 시작하고 여러 의미가 중첩되었으면 '_'로 구분시킨다. 도 8은 코드 모던화가 실제 적용된 유니콘 소스 코드의 일부를 나타내는 도면이다.

[0059] 도 9는 본 발명의 일 실시예에 따른 유니콘 최적화 성능 평가를 나타내는 그래프이다.

[0060] 제안하는 적운 모수화 코드 향상을 위한 유니콘 최적화 방법을 도 2의 SCAM 실행환경에서 수행하였다.

[0061] 본 발명의 실시예에 따라 구축한 SCAM 실행 환경에서 제안한 유니콘 최적화 기법을 테스트 하였고, 오리지널 소스 코드, 종속 벡터화, 루프 벡터화 및 유니콘 최적화 방법을 비교한 결과 기존 소스 코드 대비 루프 벡터화는 3.086% 종속 벡터화는 0.4572% 성능이 향상됨을 확인하였다. 그리고 이를 모두 적용한 유니콘 최적화 기법의 경우 기존 소스 코드 대비 3.457%의 성능이 향상됨을 확인하였다. 이는 본 발명에서 제안한 유니콘 최적화 기법이 우수한 성능을 제공하고 있음을 입증한다.

[0063] 도 10은 본 발명의 일 실시예에 따른 루프 벡터화 성능 평가를 나타내는 그래프이다.

[0064] 루프 벡터화 방법은 평균 4.5805초, 기존 소스 코드는 4.7263초의 성능을 보여주며, 이는 기존 대비 3.086% 성능 향상을 보였다.

[0066] 도 11은 본 발명의 일 실시예에 따른 종속 벡터화 성능 평가를 나타내는 그래프이다.

[0067] 종속 벡터 방법은 평균 4.7047초, 기존 소스 코드는 4.7263초의 성능을 보여주며, 이는 기존 대비 0.457% 성능 향상을 보였다.

[0069] 도 12는 본 발명의 일 실시예에 따른 유니콘 벡터화 성능 평가를 나타내는 그래프이다.

[0070] 유니콘 벡터화 성능 평가는 적운 모수화 코드 향상을 위한 유니콘 최적화 방법의 성능 향상을 위해 루프 벡터화 방법 및 속 벡터화 방법 제안하였고 오래된 포트란(Fortran) 소스 코드를 유지 보수를 위하여 코드 모던화 방법을 제안하였다. 도 2의 SCAM의 실행환경에서 SCAM 모델이 수행되었다. 유니콘 최적화 기법은 평균 4.5599초, 기존 소스 코드는 4.7263초의 성능을 보여 준다. 이는 기존 대비 3.521%의 성능 향상을 보이며, 유니콘 최적화 기법의 성능 우수성을 의미한 다.

[0072] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0073] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0074] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

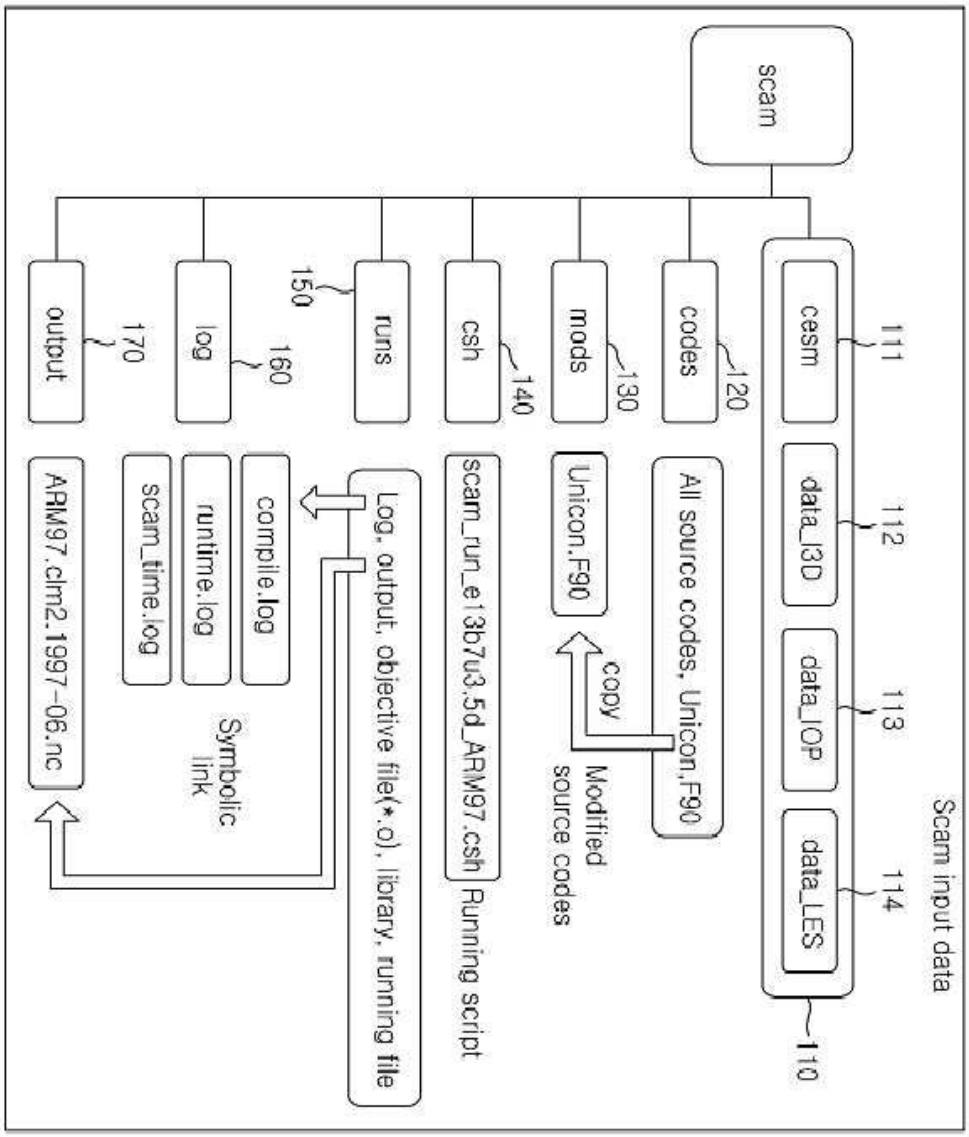
[0075] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과

다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

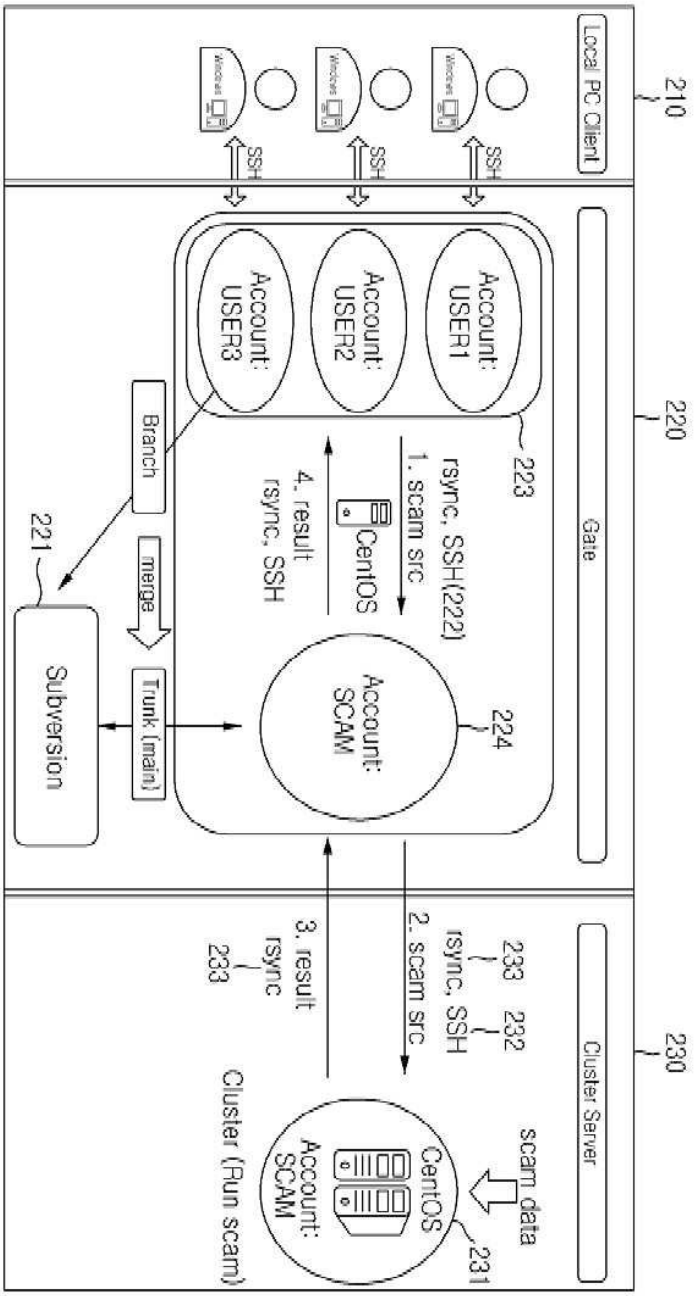
[0076] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

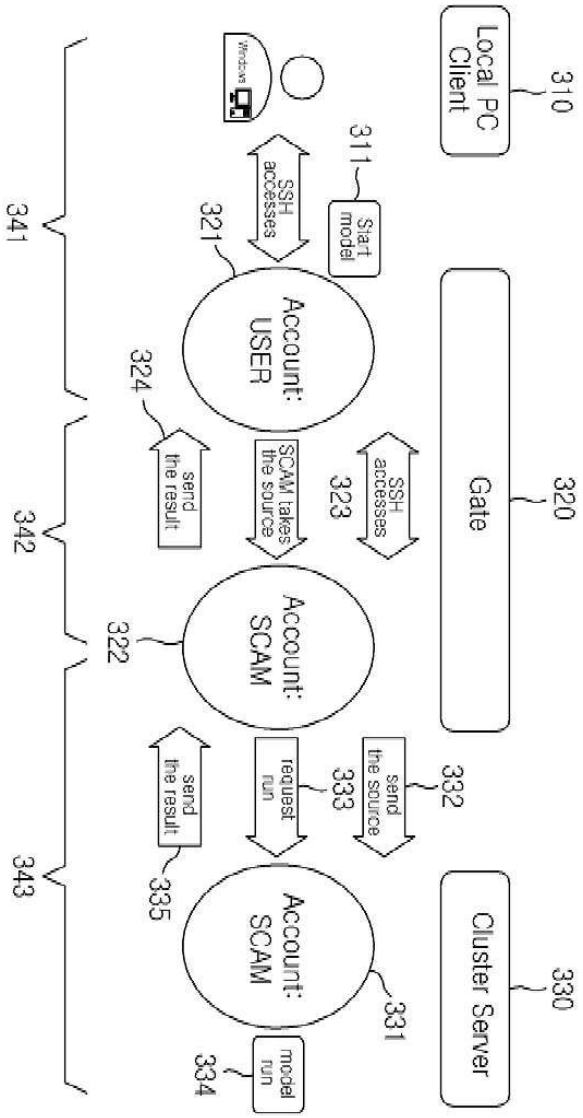
도면1



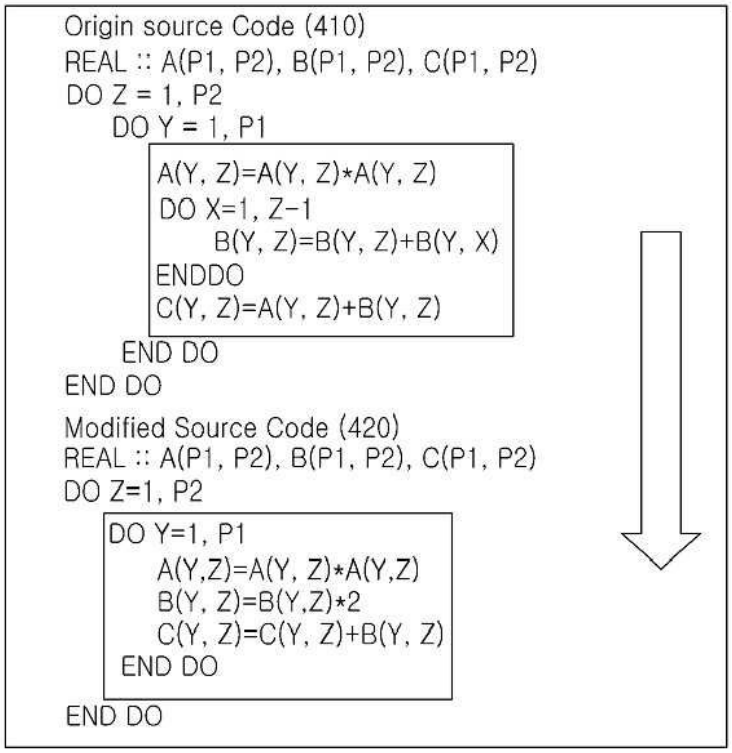
도면2


Local PC Client
210
SSH
SSH
SSH
Account: USER1
Account: USER2
Account: USER3
220
Gate
223
rsync, SSH(222)
1. scam src
CentOS
4. result
rsync, SSH
Branch
merge
221
Subversion
Trunk (main)
Account: SCAM
224
Cluster Server
230
233
232
rsync, SSH
2. scam src
scam data
CentOS
Account: SCAM
231
3. result
rsync
233
Cluster (Run scam)

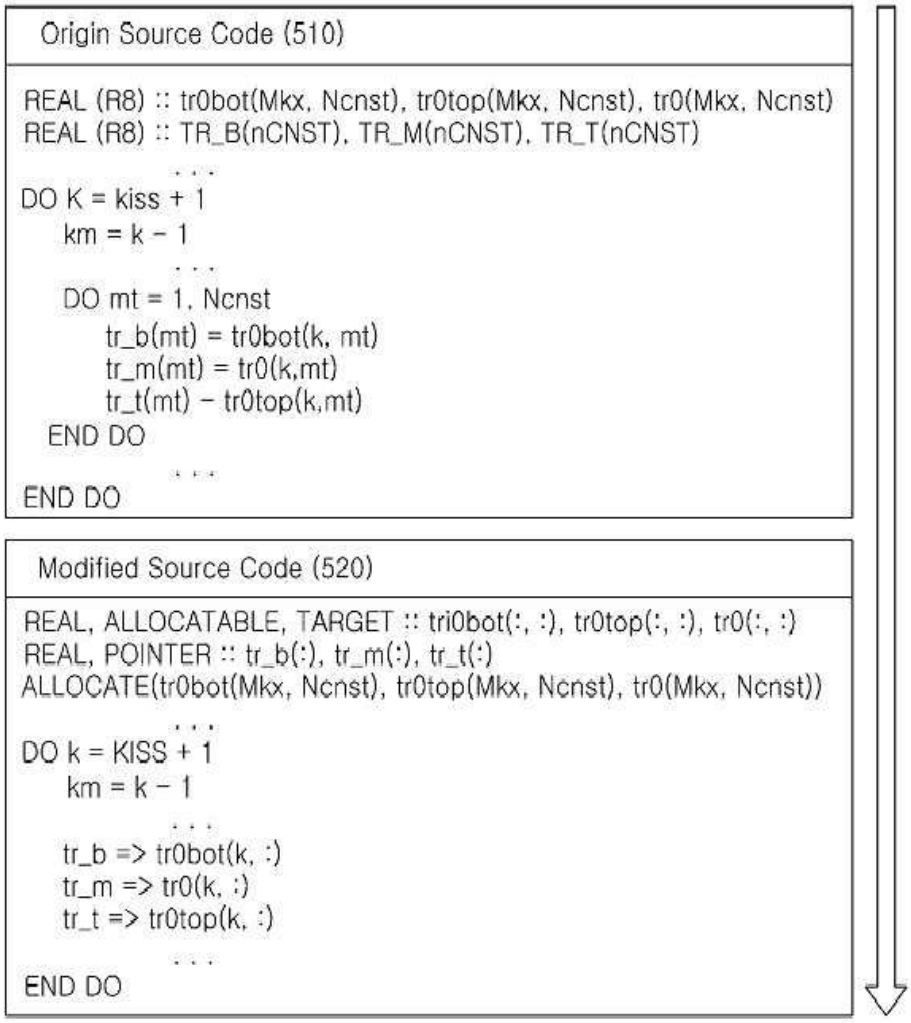
도면3


Local PC Client
310
Gate
320
Cluster Server
330
311
Start model
SSH accesses
321
Account: USER
324
send the result
SCAM takes the source
SSH accesses
323
Account: SCAM
322
send the result
335
request run
333
send the source
332
Account: SCAM
331
model run
334
341
342
343

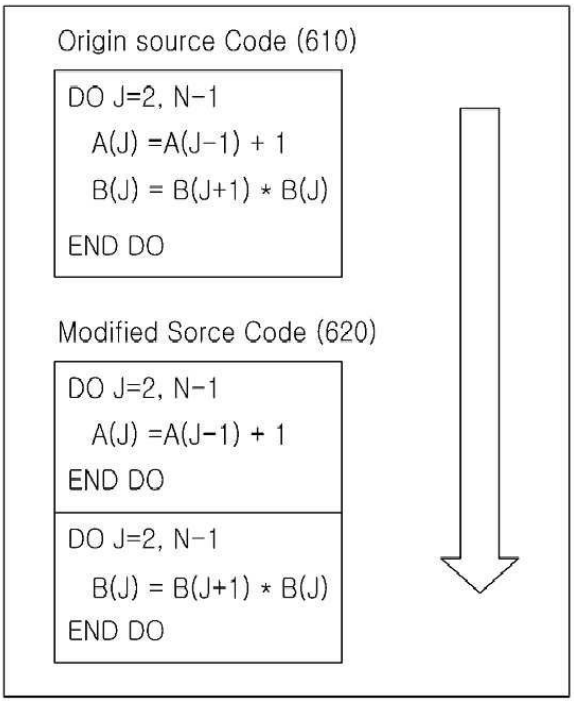
도면4


Origin source Code (410)
REAL :: A(P1, P2), B(P1, P2), C(P1, P2)
DO Z = 1, P2
DO Y = 1, P1
A(Y, Z)=A(Y, Z)*A(Y, Z)
DO X=1, Z-1
B(Y, Z)=B(Y, Z)+B(Y, X)
ENDDO
C(Y, Z)=A(Y, Z)+B(Y, Z)
END DO
END DO
Modified Source Code (420)
REAL :: A(P1, P2), B(P1, P2), C(P1, P2)
DO Z=1, P2
DO Y=1, P1
A(Y,Z)=A(Y, Z)*A(Y,Z)
B(Y, Z)=B(Y,Z)*2
C(Y, Z)=C(Y, Z)+B(Y, Z)
END DO
END DO

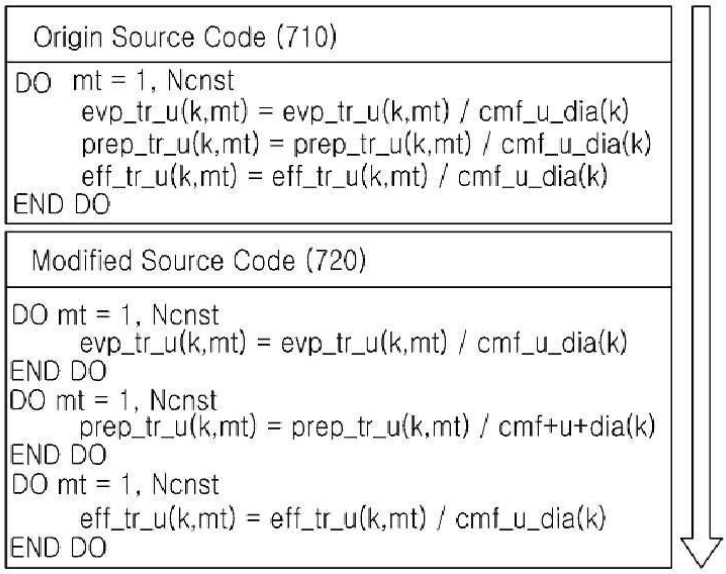
도면5


Origin Source Code (510)
REAL (R8) :: tr0bot(Mkx, Ncnst), tr0top(Mkx, Ncnst), tr0(Mkx, Ncnst)
REAL (R8) :: TR_B(nCNST), TR_M(nCNST), TR_T(nCNST)
. . .
DO K = kiss + 1
km = k - 1
. . .
DO mt = 1, Ncnst
tr_b(mt) = tr0bot(k, mt)
tr_m(mt) = tr0(k,mt)
tr_t(mt) - tr0top(k,mt)
END DO
. . .
END DO
Modified Source Code (520)
REAL, ALLOCATABLE, TARGET :: tri0bot(:, :), tr0top(:, :), tr0(:, :)
REAL, POINTER :: tr_b(:), tr_m(:), tr_t(:)
ALLOCATE(tr0bot(Mkx, Ncnst), tr0top(Mkx, Ncnst), tr0(Mkx, Ncnst))
. . .
DO k = KISS + 1
km = k - 1
. . .
tr_b => tr0bot(k, :)
tr_m => tr0(k, :)
tr_t => tr0top(k, :)
. . .
END DO

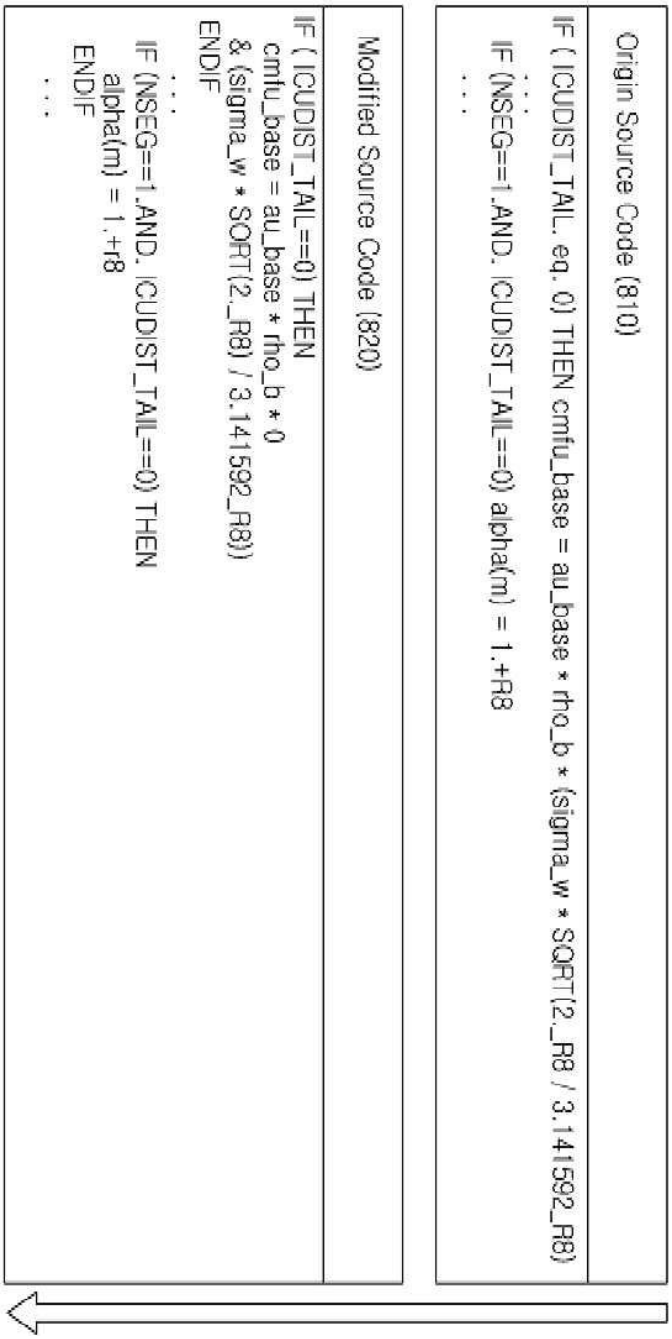
도면6


Origin source Code (610)
DO J=2, N-1
A(J) =A(J-1) + 1
B(J) = B(J+1) * B(J)
END DO
Modified Sorce Code (620)
DO J=2, N-1
A(J) =A(J-1) + 1
END DO
DO J=2, N-1
B(J) = B(J+1) * B(J)
END DO

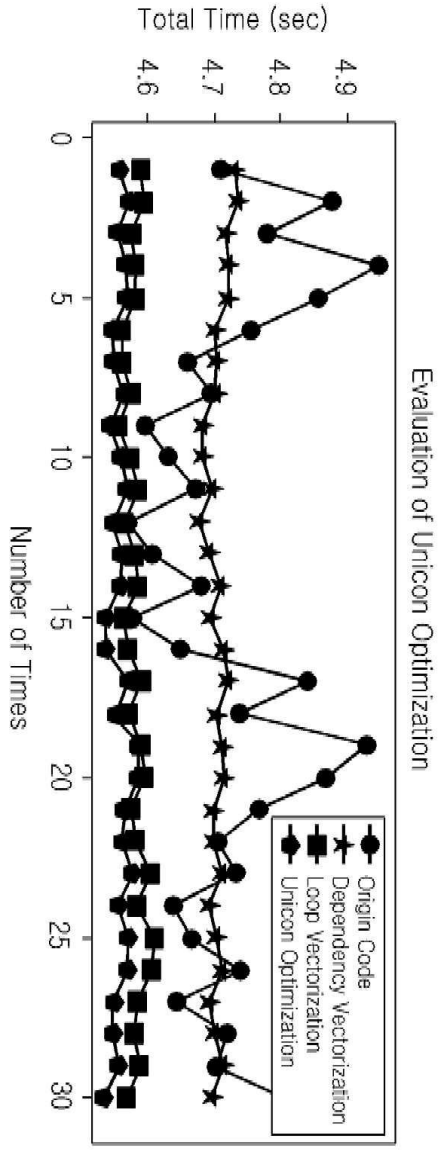
도면7


Origin Source Code (710)
DO mt = 1, Ncnst
evp_tr_u(k,mt) = evp_tr_u(k,mt) / cmf_u_dia(k)
prep_tr_u(k,mt) = prep_tr_u(k,mt) / cmf_u_dia(k)
eff_tr_u(k,mt) = eff_tr_u(k,mt) / cmf_u_dia(k)
END DO
Modified Source Code (720)
DO mt = 1, Ncnst
evp_tr_u(k,mt) = evp_tr_u(k,mt) / cmf_u_dia(k)
END DO
DO mt = 1, Ncnst
prep_tr_u(k,mt) = prep_tr_u(k,mt) / cmf+u+dia(k)
END DO
DO mt = 1, Ncnst
eff_tr_u(k,mt) = eff_tr_u(k,mt) / cmf_u_dia(k)
END DO

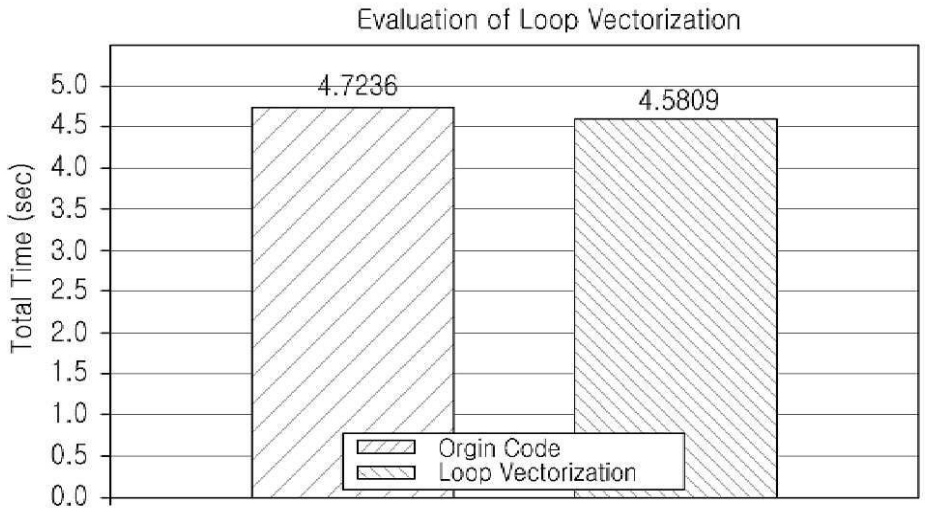
도면8


Origin Source Code (810)
IF (ICUDIST_TAIL. eq. 0) THEN cmfu_base = au_base * rho_b * (sigma_w * SQRT(2._R8 / 3.141592_R8)
. . .
IF (NSEG==1.AND. ICUDIST_TAIL==0) alpha(m) = 1.+R8
. . .
Modified Source Code (820)
IF (ICUDIST_TAIL==0) THEN
cmfu_base = au_base * rho_b * 0
& (sigma_w * SQRT(2._R8) / 3.141592_R8))
ENDIF
. . .
IF (NSEG==1.AND. ICUDIST_TAIL==0) THEN
alpha(m) = 1.+r8
ENDIF
. . .

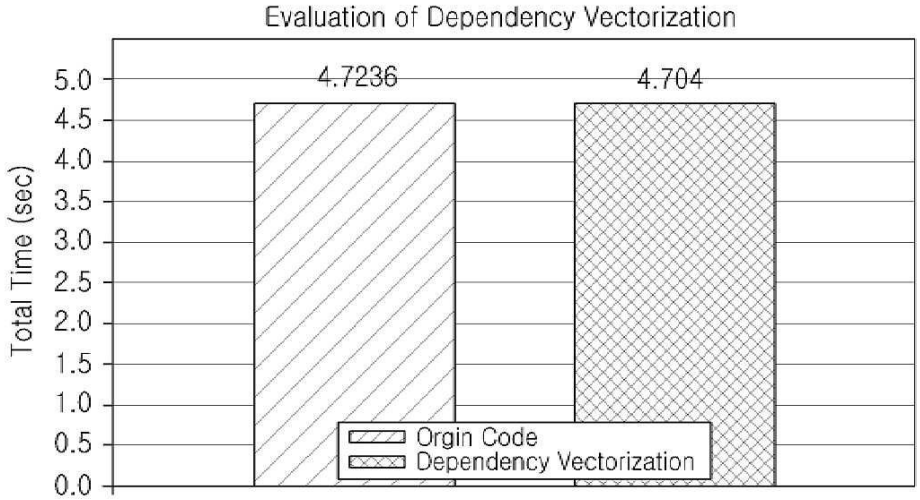
도면9


Total Time (sec)
4.6
4.7
4.8
4.9
0
5
10
15
20
25
30
Number of Times
Evaluation of Unicon Optimization
Origin Code
Dependency Vectorization
Loop Vectorization
Unicon Optimization

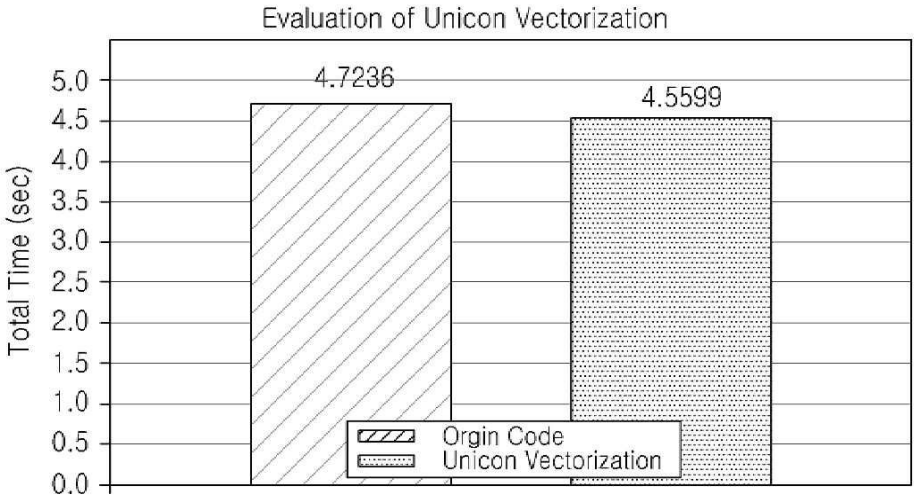
도면10


Evaluation of Loop Vectorization
Total Time (sec)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
4.7236
4.5809
Orgin Code
Loop Vectorization

도면11


Evaluation of Dependency Vectorization
Total Time (sec)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
4.7236
4.704
Orgin Code
Dependency Vectorization

도면12


Evaluation of Unicon Vectorization
Total Time (sec)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
4.7236
4.5599
Orgin Code
Unicon Vectorization