(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0137796
(43) 공개일자 2015년12월09일

(51) 국제특허분류(Int. Cl.)
G06F 15/16 (2006.01) *G06F 17/00* (2006.01)
(21) 출원번호 10-2014-0066299
(22) 출원일자 2014년05월30일
심사청구일자 없음

(71) 출원인
경희대학교 산학협력단
경기도 용인시 기흥구 덕영대로 1732 (서천동, 경희대학교 국제캠퍼스내)
(72) 발명자
허의남
경기도 용인시 기흥구 사은로126번길 46, 316동 1802호 (보라동, 현대모닝사이드1차아파트)
팜 푸옥 홍
서울특별시 동대문구 경희대로 26, 전자정보대학 331호 (회기동)
손아영
서울특별시 동대문구 경희대로 26, 전자정보대학 331호 (회기동)
(74) 대리인
특허법인 신지

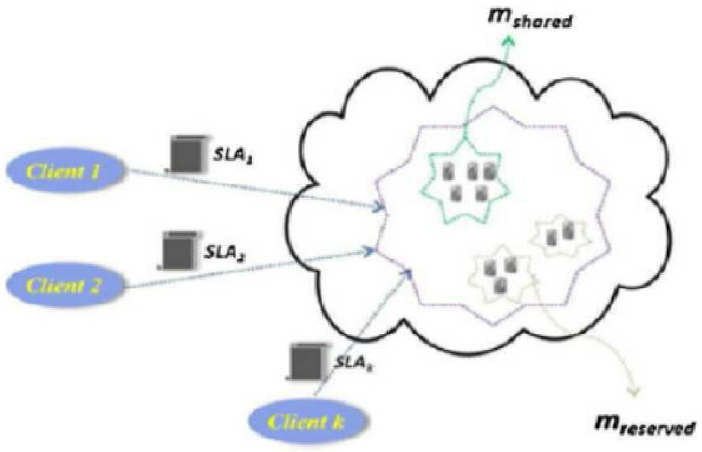
전체 청구항 수 : 총 3 항

(54) 발명의 명칭 **모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법**

(57) 요 약

본 발명에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법은 먼저, 최소 가상 머신을 결정하고, 리소스 할당 전략에 기초하여 가상 머신 리소스를 할당한다. 그리고, 데이터를 분류하고, 할당된 가상 머신의 용량에 따라 데이터를 할당한다. 다음으로, 씩 클라이언트(Thick Client)의 최적의 수를 결정하고, 씩 클라이언트에 서로 다른 밴드위스(Bandwidth)에 따라 청크(Chunk)를 할당한다. 그리고, 데이터를 결합하여 씬 클라이언트로 전송한다. 최소 가상 머신을 결정하는 단계는 응답시간을 최소화하고 서비스 품질(Quality of Service)을 만족시킨다. 그리고, 리소스 할당 전략에 기초하여 가상 머신 리소스를 할당하는 단계는 가용한 가상 머신을 공유 할당 그룹 및 예약 할당 그룹으로 분류하고, 하나의 큐를 통해 들어오는 큐를 상기 공유 할당 그룹에 할당하고, 상기 예약 할당 그룹은 도착하는 잡(Job)에 따라 분류하여 할당한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 1345201714
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 차세대정보컴퓨팅기술개발
연구과제명 클라우드 환경의 Thin-Client 모바일 단말 기술
기 여 율 1/1
주관기관 경희대학교
연구기간 2013.07.01 ~ 2014.06.30

명 세 서

청구범위

청구항 1

최소 가상 머신을 결정하는 단계;

리소스 할당 전략에 기초하여 상기 가상 머신 리소스를 할당하는 단계;

데이터를 분류하고, 상기 할당된 가상 머신의 용량에 따라 데이터를 할당하는 단계;

씩 클라이언트(Thick Client)의 최적의 수를 결정하는 단계;

상기 씩 클라이언트에 서로 다른 밴드위스(Bandwidth)에 따라 청크(Chunk)를 할당하는 단계; 및

데이터를 결합하여 씬 클라이언트로 전송하는 단계;

를 포함하는 것을 특징으로 하는 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법.

청구항 2

제1항에 있어서,

상기 최소 가상 머신을 결정하는 단계는 응답시간을 최소화하고 서비스 품질(Quality of Service)을 만족시키는 것을 특징으로 하는 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법.

청구항 3

제1항에 있어서,

상기 리소스 할당 전략에 기초하여 상기 가상 머신 리소스를 할당하는 단계는 가용한 가상 머신을 공유 할당 그룹 및 예약 할당 그룹으로 분류하고, 하나의 큐를 통해 들어오는 큐를 상기 공유 할당 그룹에 할당하고, 상기 예약 할당 그룹은 도착하는 잡(Job)에 따라 분류하여 할당하는 것을 특징으로 하는 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 데이터 분배 및 자원 할당에 관한 기술로서, 보다 상세하게는 데이터 분배 및 컴퓨팅 리소스 이용을 최적화하는 방법에 관한 기술이다.

배 경 기 술

[0002] 클라우드 컴퓨팅 환경에서 하드웨어상의 제약이 사라지고 있다. 단말의 요소 기술이 발달하면서 더 강력한 씬 클라이언트 형태의 이동형 단말 , 이용자 편의 성과 이동성이 강화된 스마트 단말로 발전하고 있기 때문이다. 스토리지와 서버가 모두 클라우드에 존재하고 클라이언트에는 데이터가 저장될 필요가 없다. 그래서 가능해진 씬 클라이언트화는 PC하드웨어 시장에 근본적 변화를 일으키게 한다. 하지만, 모바일 디바이스는 크기가 작기 때문에 PC에 비해 컴퓨팅 자원이 부족하다는 문제점이 있으며 자원소비가 많고 높은 복잡도를 가진 애플리케이션은 이와 같은 모바일 디바이스에서 설치 및 운영되기 어렵다.

발명의 내용

해결하려는 과제

[0003] 본 발명이 해결하고자 하는 과제는 다양한 SLA(SErvice Level Agreement)를 만족시키기 위한 리소스 할당 전략을 선택하고, 서비스 품질(QoS)를 만족하면서 데이터 분배와 클라우드 컴퓨팅 리소스 이용률을 최적화 시키기 위한 자원할당 및 데이터 분배를 최적화하는 방법을 제공하는 것이다.

과제의 해결 수단

[0004] 본 발명에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법은 먼저, 최소 가상 머신을 결정하고, 리소스 할당 전략에 기초하여 가상 머신 리소스를 할당한다. 그리고, 데이터를 분류하고, 할당된 가상 머신의 용량에 따라 데이터를 할당한다. 다음으로, 씩 클라이언트(Thick Client)의 최적의 수를 결정하고, 씩 클라이언트에 서로 다른 밴드위스(Bandwidth)에 따라 청크(Chunk)를 할당한다. 그리고, 데이터를 결합하여 씬 클라이언트로 전송한다. 최소 가상 머신을 결정하는 단계는 응답시간을 최소화하고 서비스 품질(Quality of Service)을 만족시킨다. 그리고, 리소스 할당 전략에 기초하여 가상 머신 리소스를 할당하는 단계는 가용한 가상 머신을 공유 할당 그룹 및 예약 할당 그룹으로 분류하고, 하나의 큐를 통해 들어오는 큐를 상기 공유 할당 그룹에 할당하고, 상기 예약 할당 그룹은 도착하는 잡(Job)에 따라 분류하여 할당한다.

발명의 효과

[0005] 본 발명에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법은 협업을 통해 데이터 분배와 자원할당에 대한 분배를 고려하여 클라우드 컴퓨팅 자원 이용률을 최적화하고 서비스 품질을 만족하는 알고리즘 및 기술을 통하여 모바일 컴퓨팅 시장에서 제한적 자원을 효율적으로 사용할 수 있도록 할 수 있다. 또한, 클라우드 서버를 이용하여 외부의 장치의 자원을 취득할 수 있는 협업을 통해 자원 이용률의 최적화를 이룰 수 있다.

도면의 간단한 설명

[0006] 도 1은 본 발명에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 최소 가상 머신 결정의 일 실시예를 나타내는 도면이다.

도 2a는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 리소스 할당 전략을 나타내는 도면이고, 도 2b는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 리소스 할당 전략을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 우선순위에 따른 가상 머신 할당을 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 데이터 결합 과정을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 효율을 평가한 그래프를 나타낸다.

도 6은 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 데이터 전송 시간 비교를 나타내는 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0007] 이하, 본 발명의 실시예를 첨부된 도면들을 참조하여 상세하게 설명한다. 본 명세서에서 사용되는 용어 및 단어들은 실시예에서의 기능을 고려하여 선택된 용어들로서, 그 용어의 의미는 발명의 의도 또는 관례 등에 따라 달라질 수 있다. 따라서 후술하는 실시예에서 사용된 용어는, 본 명세서에 구체적으로 정의된 경우에는 그 정의에 따르며, 구체적인 정의가 없는 경우는 당업자들이 일반적으로 인식하는 의미로 해석되어야 할 것이다.

[0008] 도 1은 본 발명에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 최소 가상 머신 결정의 일 실시예를 나타내는 도면이다.

[0009] 도 1을 참조하면, 본 발명에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법은 데이터를 분배하고 자원이용률을 최적화하기 위해 다음과 같은 과정을 거친다. 모바일 장치에 모바일 접속의 이익이나 외부의 장치의 자원을 취득할 수 있는 협업을 제공할 수 있도록 하는 클라이언트 협업(collaboration) 구조를 통해 QoS를 만족시키며 데이터 분배와 리소스 이용률을 최적화 시킬수 있다. 그 구조는 중간에 씩 클라이언트를 관리하는 브로커를 두고 이 브로커는 클라우드 제공자들에게 접근할 수 있다. 브로커를 이용하여 최적의 씩 클라이언트를 선택하고 다른 자원 프로세서들의 데이터를 배포하여 다시 브로커를 통하여 데이터를 결합하여

씬 클라이언트로 전송하게 된다. 구조상 두가지 레이어로 나누고 첫 번째 레이어(Layer)에서는 클라우드 구간으로 SLA만족시키고 응답시간을 최소화 시키시 위해 VM수를 최소화하며 VM용량에 따른 데이터를 분류하고, 분할 및 자원할당을 수행하게 된다. 두 번째 레이어에서는 씩 클라이언트 구간으로 최적은 씩 클라이언트 수를 결정하고 다른 자원의 프로세서들과 데이터를 배포하며 이를 결합하여 씬 클라이언트로 전송하게 된다.

[0010] 첫 번째 레이어에서는 최소 VM 결정-> 2)리소스 할당전략에 따른 VM 리소스 할당 ->3)데이터를 분류하고 VM용량에 따른 데이터 할당(그리디 알고리즘을 사용하여 용량에 따른 Best Vm들은 선택하게 된다.)

표 1

[0011] 응답시간을 최소화하고 QoS를 만족시키는 가상 머신 결정 알고리즘

알고리즘 1

```
Input : λ // 도착비율 , : 서비스 비율
        SLA(x,z) // x : 응답시간, z : 타겟가능성
Output :  m //요구사항에 만족하는 VM최소수

float σ=λ/μ
functiondetermineMin VM( μ , σ ,x,z){
      if ( σ ==(int)σ )       m= (int)σ  ;
      else   m =(int)Math.floor( σ ) +1;
      While F(x) <=z m++;
      return m;
```

[0012] CDF는 어떤 확률 분포에 대해서, 확률 변수가 특정 값보다 작거나 같은 확률을 나타낸다. 최소 VM를 m 이라고 선언하고 target 가능성에 도달할 때까지 증가시켜 구한다. F(x) <z 를 만족하는 하는 m 을 찾을때까지 증가셔켜서 최소 VM들의 수를 구한다. (응답시간에 따른 Qos를 평가하기 위해 target threshold x 보다 작아야한다.)

[0013] 도 2a는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 리소스 할당 전략을 나타내는 도면이고, 도 2b는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 리소스 할당 전략을 설명하기 위한 도면이다.

[0014] 도 2a 및 도 2b를 참조하면, 이용가능한 VM들을 두 그룹으로 분류하고 첫 번째 그룹을 SA(shared Allocation) , 두 번째 그룹을 RA(Reserved allocation)이라고 정의한다. SA는 하나의 큐를 통해 들어오며 RA는 각각 도착하는 job에 따라 큐를 분류하고 VM을 할당한다. 그리고, 자원할당 전략은 두 가지 제약조건을 만족해야한다.

[0015] 첫 번째 제약조건은 QoS를 만족하기 위한 조건으로 수학식 1(Little law 법칙)을 따른다.

수학식 1

$$E[N]=\frac{\rho}{1-\rho}\quad\left(\rho=\frac{\lambda}{\mu}\right)$$

$$E[T]=\frac{E[N]}{\lambda}=\frac{\rho}{\lambda(1-\rho)}=\frac{1}{\mu(1-\rho)}=\frac{1}{\mu-\lambda}$$

$$\mu\geq\frac{1}{E[T]}+\lambda$$

[0016]

[0017] 그리고, 두 번째 제약조건은 수학식 2를 따른다.

수학식 2

[0018] $$\sin\alpha \leq \alpha$$

[0019] α는 D와 관계과 있으며 다음 표를 따른다. D는 수학식 3과 같이 정의해둘 수 있다.

수학식 3

[0020] $$E[SLA_1\text{-}SLA_2]$$

표 2

[0021] α 및 D의 관계

D	α
[0,20]	0
[20,40]	20
[40,66]	50
[66,88]	70

[0022] 수학식 2, 수학식 3 및 표 2에 기초하여 수학식 4를 산출한다.

수학식 4

[0023] $$\sin\alpha = \frac{\lambda_2}{\sqrt{\lambda^{2}{}_1 + \lambda_2^2}}$$

[0024] 첫 번째 제약조건에 우선적으로 만족하여서 두 번째 제약조건에 만족한다면 RA에 할당하고 아닐경우에 자원은 SA에 할당하게 된다. 표 3은 자원할당 전략을 나타낸다.

표 3

[0025] 자원할당 전략

알고리즘 2 자원할당 전략

$Input: \lambda_1, \quad \lambda_2$ //도착비율, μ : 서비스비율
SLA_1, SLA_2
E // 서비스 실행 기대시간
$output: SA, RA$ // $Allocation\ Strategy$

$Function\ Determine\ AllocStrategy(\lambda_1, \lambda_2, SLA_1 SLA_2, E, \mu)$
$Calulate\ SLA \quad Difference\ D$
Get the corresponding angle α from the SLA difference table
$If(\mu \geq (1/E[T] + \lambda_1)$ && $If(\mu \geq (1/E[T] + \lambda_2)$
return RA
else
return SA
else
return false
}

[0026] 도 3은 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 우선순위에 따른 가상 머신 할당을 설명하기 위한 도면이다.

[0027] 도 3을 참조하면, 데이터를 분류하고 VM용량에 따른 데이터 할당하고, 알고리즘을 사용하여 용량에 따른 Best VM들을 선택하게 된다. 두 번째 레이어에서, 씩 클라이언트에 데이터를 배포하고 다시 데이터를 결합하여 씬 클라이언트로 전송하게 되는데 데이터 배포단계에서는 서로 다른 bandwidth 에 따라 layer1에서 할당된 블록을 분할받고 프로세서 또한 내림 차순으로 정렬하여 놓은 용량을 가진 씩 클라이언트가 큰 청크(chunk)를 할당받게 된다. 씬 클라이언트로 전송 시에는 방화벽에 의한 복잡성을 줄이기 위해 마스터 슬레이브 방식으로 통신을 하게 된다.

[0028] 1)씩 클라이언트의 최적의 수 결정

[0029] 다음의 식은 씩 클라이언트의 응답시간과 워크로드에 기반한 문제를 공식화 하여 수학식 5 를 산출한다.

수학식 5

$$Minimize \quad M$$

$$\sum_{n \in S_1^m} W_n < \sum_{m=1}^{M} a_m W_m,$$

$$Min\left(T_n^R, \forall n \in S_1^m\right) > T_m^R, \forall m$$

[0030]

[0031] 씩 클라이언트의 수를 최소화하고 다양한 씬클라이언트의 수를 만족시켜야한다. 브로커의 연결을 통하여 첫 번째 제약조건은 씬 클라이언트의 전체 워크로드는 씩 클라이언트의 전체 워크로드를 능가해서는 안된다. 두 번째

제약조건은 각 씩 클라이언트의 응답시간이 씬클라이언트가 요구되는 응답시간 보다 작아야한다.다양한 씬 클라이언트들이 직접연산을 처리할때보다 브로커를 통하여 연산을 할 수 있도록하여 컴퓨팅 효율을 높으면서도, 응답시간이 짧아야한다는 조건을 만족시켜야한다. 다른 용량의 데이터 분배는 수학식 6에 의해 이루어진다.

표 4

[0032] 수학식 변수 정의

Symbol	Meaning
S_1^m	씬 클라이언트의 집합으로 씩 클라이언트에서 처리될 데이터 사이즈
T_n^R	응답시간/씬클라이언트 n의 워크로드 단위
T_m^R	응답시간/씩클라이언트 m의 워크로드 단위
M	씩 클라이언트의 수
W_n	씬 클라이언트의 워크로드
W_m	씩 클라이언트의 워크로드

표 5

[0033] 씩 클라이언트의 최소수 결정

알고리즘 씩 클라이언트의 최소수 결정

$Input : \left(W_n, T_n^R, S_1^m \right)$

$Output :$ 씩 클라이언트의 최소수

$Function$ determine$Number\ Thick Client$

```
Insert thick clients into a list L1
Insert thin clients into a list L2
Sort L1 according to the decreasing order of capacity
Sort L2 according to the decreasing order of capacity
for(int i=0 . i<=nuber of thin clients; i++
   {
        FirstITem = poplist(L1);
        WorkloadThick = FirstITem.workload;
        WorkloadThin += L2[i].workload
      if! (L2[i].responseTime>FistITem.responsetime &&Workload Thick>workload Thin)
                {
                     m++;
                     workloadThin = L2[i].workload
                }
     }
      return m
}
```

수학식 6

$$\frac{w(ch_1)}{b_1} = \frac{w(ch_2)}{b_2} = \ldots\ldots = \frac{w(ch_i)}{b_i} = t$$

$$set\ S = w(data) = \sum_{i=0}^{n} w(ch_1) = t\sum_{i=0}^{n} b_i$$

$$Thus, w(ch_1) = t * b_i = \frac{S}{\sum_{i=o}^{n} b_i} * b_i$$

[0034]

[0035] 도 4는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 데이터 결합 과정을 설명하기 위한 도면이다.

[0036] 도 4를 참조하면, 서로 다른 밴드위스에 따라 레이어1에서 할당된 블록을 분할받고, 프로세서 또한 내림 차순으로 정렬하여 놓은 용량을 가진 씩 클라이언트가 큰 청크를 할당받게 된다. 다음으로, 데이터를 결하하고 씬 클라이언트로 전송한다. 씬 클라이언트로 전송시에는 방화벽에 의한 복잡성을 줄이기 위해 마스터 슬레이브 방식으로 통신을 하게 된다.

[0037] 도 5는 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 효율을 평가한 그래프를 나타낸다.

[0038] 도 5를 참조하면, 응답시간에 따른 가상 머신수(501)로부터 응답시간이 증가할수록 최소 가상 머신 수가 감소하는 것을 확인할 수 있다. 반면에 타겟 가능성에 따른 가상 머신수(502)로부터 SLA에 만족하는 최소 가상 머신은 타겟 가능성에 비례하는 것을 확인할 수 있다.

[0039] 도 6은 본 발명의 일 실시예에 따른 모바일 컴퓨팅 환경에서의 자원할당 및 데이터 분배를 최적화하는 방법의 데이터 전송 시간 비교를 나타내는 그래프이다.

[0040] 도 6을 참조하면, 데이터 전송시간에 대한 비교를 확인하면, 다른 하나의 프로세서에 대한 시간 비교를 통해 본 발명의 데이터 전송시간이 더 빠름을 확인할 수 있다.

[0041] 상술한 내용을 포함하는 본 발명은 컴퓨터 프로그램으로 작성이 가능하다. 그리고 상기 프로그램을 구성하는 코드 및 코드 세그먼트는 당분야의 컴퓨터 프로그래머에 의하여 용이하게 추론될 수 있다. 또한, 상기 작성된 프로그램은 컴퓨터가 읽을 수 있는 기록매체 또는 정보저장매체에 저장되고, 컴퓨터에 의하여 판독되고 실행함으로써 본 발명의 방법을 구현할 수 있다. 그리고 상기 기록매체는 컴퓨터가 판독할 수 있는 모든 형태의 기록매체를 포함한다.

[0042] 이상 바람직한 실시예를 들어 본 발명을 상세하게 설명하였으나, 본 발명은 전술한 실시예에 한정되지 않고, 본 발명의 기술적 사상의 범위 내에서 당분야에서 통상의 지식을 가진자에 의하여 여러 가지 변형이 가능하다.

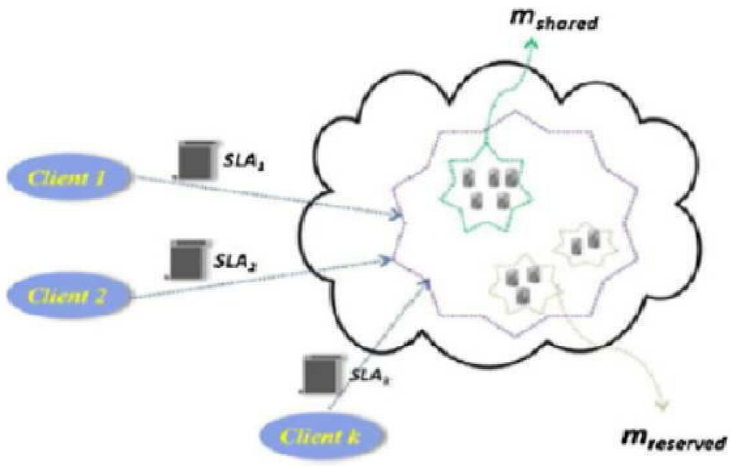
부호의 설명

[0043] 501: 응답시간에 따른 가상 머신수

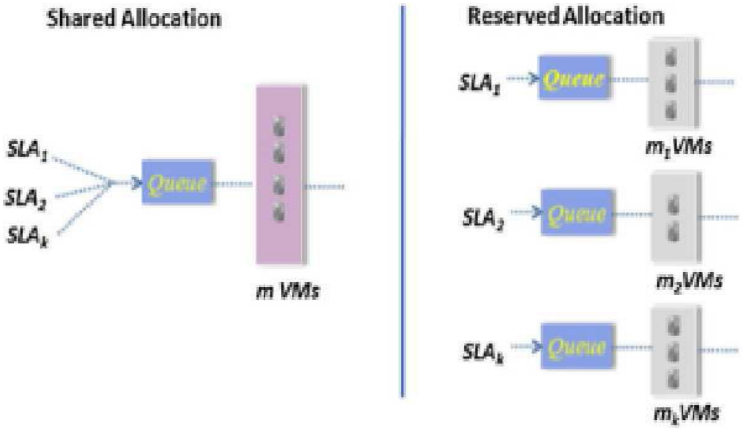
502: 타겟 가능성에 따른 가상 머신수

도면

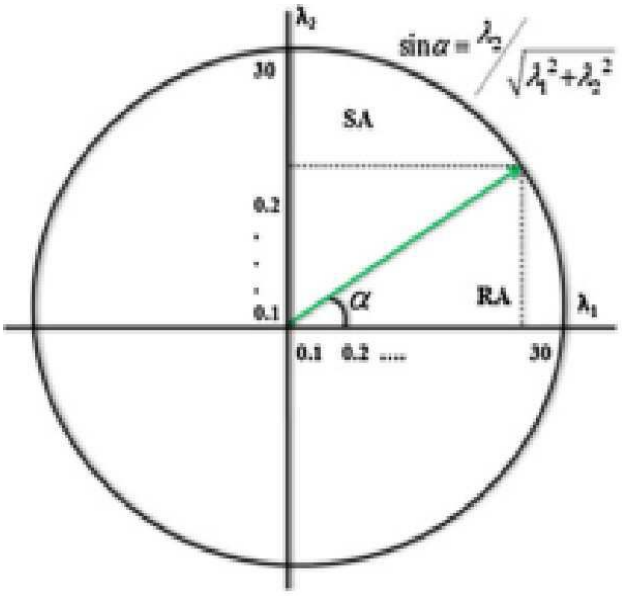
도면1



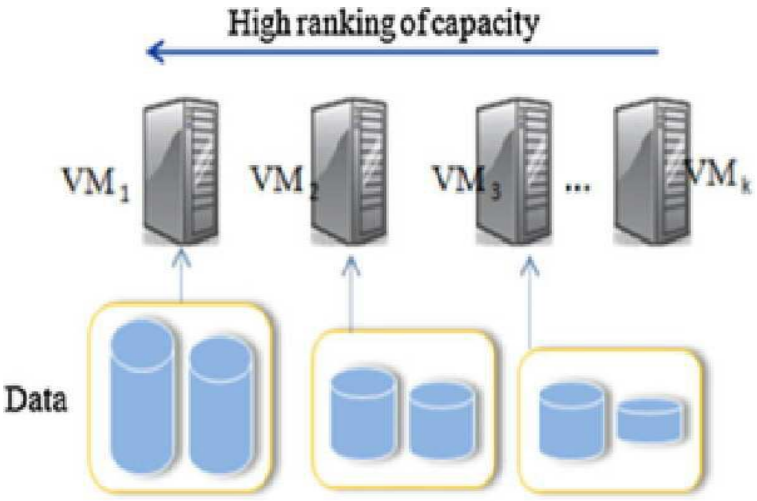
도면2a



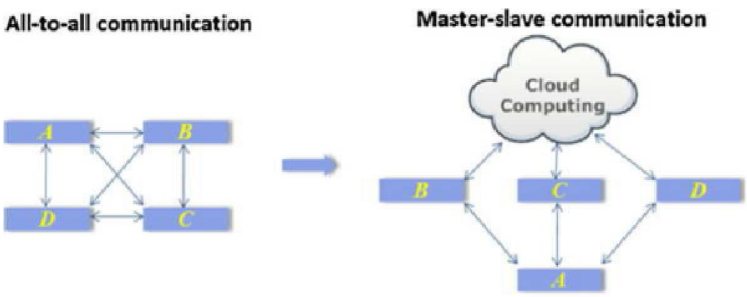
도면2b


λ2
sinα = λ2 / √(λ1² + λ2²)
30
SA
0.2
0.1
α
RA
λ1
0.1 0.2 30

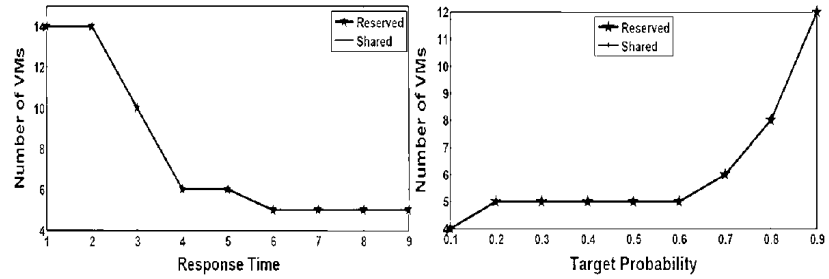
도면3


High ranking of capacity
VM1
VM2
VM3
...
VMk
Data

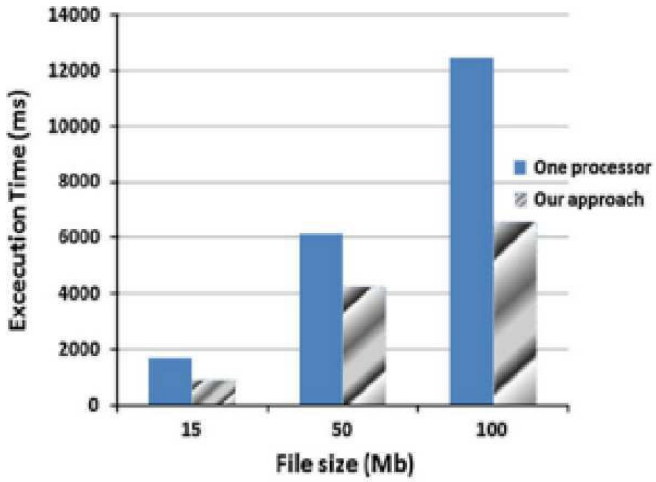
도면4


All-to-all communication
Master-slave communication
Cloud Computing
A
B
C
D

도면5


Reserved
Shared
Number of VMs
Response Time
Reserved
Shared
Number of VMs
Target Probability

도면6


14000
12000
10000
8000
6000
4000
2000
0
Execution Time (ms)
15
50
100
File size (Mb)
One processor
Our approach