



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0111482

(43) 공개일자 2015년10월06일

(51) 국제특허분류(Int. Cl.)

G06F 19/00 (2011.01)

(21) 출원번호 10-2014-0034526

(22) 출원일자 2014년03월25일

심사청구일자 2014년03월25일

(71) 출원인

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

정연돈

서울 중구 청계천로 400, 105동 812호 (황학동, 롯데캐슬베네치아)

임병남

경기 용인시 수지구 정평로 61, 505동 1702호 (풍덕천동, 신정마을성지아파트)

(74) 대리인

김동용

전체 청구항 수 : 총 8 항

(54) 발명의 명칭 그래프 극대 매칭 방법

(57) 요약

본 발명은 입력된 그래프를 병렬로 처리할 수 있는 프리quel 프레임워크 상에서 그래프의 특성을 고려하여 고속으로 극대 매칭을 찾을 수 있도록 하는 그래프 극대 매칭 방법에 관한 것이다.

본 발명을 이용함으로써 대용량의 그래프에 대한 매칭 처리를 고속으로 수행할 수 있도록 하는 효과가 있다.

대표도 - 도3

```

1: Initialize: .....
2:   Result matching set  $M \leftarrow \phi$ ; .....
3:   Set step as Degree Broadcast; .....
4: while this vertex is not Matched do
5:   if step is Degree Broadcast then .....
6:     Send degree messages to adjacent vertices; .....
7:     Set step as Match Trial; .....
8:   else if step is Match Trial then .....
9:     for all Received messages do .....
10:       Find a vertex which contains the lowest degree .....
11:     end for .....
12:     candidate mate  $\leftarrow$  sender of the lowest degree vertex; .....
13:     Send a match request message to candidate mate; .....
14:     Set step as Confirm; .....
15:   else if step is Confirm then .....
16:     for all Received messages do .....
17:       if Sender of the message == candidate mate then .....
18:          $e \leftarrow$  edge (this vertex, candidate mate); .....
19:          $M \leftarrow M \cup e$ ; .....
20:         Set this vertex as Matched; .....
21:       end for .....
22:     Set step as Degree Broadcast; .....
23:   Sync for superstep; .....
24: end while
  
```

초기화 스텝(S1)

차수 전송 스텝(S2)

매치 시도 스텝(S3)

확인 스텝(S4)

글로벌 동기화(S5)

이 발명을 지원한 국가연구개발사업

과제고유번호 2013022884

부처명 교육부

연구관리전문기관 한국연구재단

연구사업명 (이공)일반연구자-기본(모험연구)

연구과제명 소셜 네트워크 서비스에서 프라이버시 문제분석 및 보호 방안 연구

기 여 율 1/1

주관기관 고려대학교 산학협력단

연구기간 2013.05.01 ~ 2014.04.30

명세서

청구범위

청구항 1

- (a) 제1 정점(vertex)에 연결된 에지(edge)에 따른 차수(degree)를 포함하는 차수 메시지를 인접한 제2 정점(Vertex)들로 전송하는 단계;
- (b) 상기 인접한 제2 정점들로부터 차수 메시지를 수신하는 단계;
- (c) 수신된 상기 인접한 제2 정점들의 차수 메시지들의 차수들 간 비교로 후보 정점을 결정하는 단계; 및
- (d) 상기 후보 정점으로 매치 요구 메시지를 전송하는 단계;를 포함하는,
그래프 극대 매칭 방법.

청구항 2

- 제1항에 있어서,
상기 단계 (c)는 상기 인접한 제2 정점들 중 가장 작은 차수를 가지는 정점을 후보 정점으로 결정하는,
그래프 극대 매칭 방법.

청구항 3

- 제1항에 있어서,
(e) 상기 인접한 제2 정점들 중 하나 이상의 제2 정점들로부터 매치 요구 메시지를 수신하는 단계;
- (f) 상기 후보 정점으로부터 매치 요구 메시지를 수신하였는지를 결정하는 단계; 및
- (g) 상기 후보 정점으로부터 매치 요구 메시지의 수신시에 매칭 세트를 업데이트하는 단계;를 더 포함하는,
그래프 극대 매칭 방법.

청구항 4

- 제3항에 있어서,
(h) 상기 제1 정점 및 상기 제2 정점들을 포함하는 그래프로부터 상기 제1 정점과 상기 후보 정점을 연결하는 에지를 제거하여 상기 그래프를 변경하는 단계;를 더 포함하며,
상기 매칭 세트는 상기 업데이트에 따라 상기 제1 정점과 상기 후보 정점으로 구성된 에지를 포함하고, 상기 단계 (a) 내지 (g)는 변경된 상기 그래프에 대해서 반복적으로 수행되는,
그래프 극대 매칭 방법.

청구항 5

- 제4항에 있어서,
상기 그래프 극대 매칭 방법은 복수의 머신 상에서 수행되며,
상기 단계 (a) 이전에, 상기 그래프 극대 매칭 방법은 상기 제1 정점 및 상기 제2 정점들을 포함하는 그래프를 분할하는 단계; 및 분할된 그래프 각각을 복수의 머신에 각각 할당하는 단계;를 더 포함하며,
상기 복수의 머신 중 상기 제1 정점을 포함하는 분할된 그래프를 할당받은 머신이 상기 단계 (a) 내지 (g)를 수행하는,
그래프 극대 매칭 방법.

청구항 6

제5항에 있어서,

상기 그래프 극대 매칭 방법은 분산 처리를 위한 BSP(Bulk-synchronous parallel) 모델을 이용하고 그래프 처리 아키텍처인 프리겔(Pregel) 상에서 수행되는,

그래프 극대 매칭 방법.

청구항 7

제5항에 있어서,

상기 복수의 머신 각각은 분할된 그래프의 복수의 정점들 각각을 제1 정점으로 하여 상기 단계 (a) 내지 (g)를 수행하는,

그래프 극대 매칭 방법.

청구항 8

제4항에 있어서,

상기 단계 (a) 내지 (g)의 반복 수행은 상기 매칭 세트의 업데이트로 상기 매칭 세트가 변경된 경우나 상기 그래프의 하나 이상의 정점이 액티브 상태인 경우에 수행하는,

그래프 극대 매칭 방법.

발명의 설명

기술 분야

[0001] 본 발명은 그래프 극대 매칭 방법에 관한 것으로서, 입력된 그래프를 병렬로 처리할 수 있는 프리겔 프레임워크 상에서 그래프의 특성을 고려하여 고속으로 극대 매칭을 찾을 수 있도록 하는, 그래프 극대 매칭 방법에 관한 것이다.

배경 기술

[0002] 그래프 처리 알고리즘은 다양한 분야에서 활용되고 있다. 그래프 처리 알고리즘은 최단 경로, 매칭 그래프, 클러스터링 그래프 등을 구성할 수 있고 다양한 응용 예에 적용될 수 있다. 이러한 그래프 처리 알고리즘은 프로그램 등에 의해서 인식될 수 있는 그래프를 입력받아 그 결과를 출력한다. 그래프는 일반적으로 정점(Vertex)과 정점과 정점을 연결하는 에지(edge)로 표현된다.

[0003] 한편 인터넷이 보편화되고 인터넷에 접속할 수 있는 전자 기기 등이 널리 보급됨에 따라 그래프 처리 알고리즘은 더 큰 사이즈의 그래프를 처리할 필요가 있다. 심지어 수백만 개 또는 수천만 개 이상의 정점으로 구성된 그래프가 입력으로 이용된다. 그래프 처리 알고리즘은 알고리즘의 복잡도가 적어도 N의 3승 이상이고 이에 따라 실질적으로 큰 그래프에 대해서는 처리가 힘들어진다. 큰 사이즈의 그래프는 예를 들어 웹 사이트나 소셜 네트워크의 그래프일 수 있다.

[0004] 대용량의 그래프 등과 같은 빅 데이터를 처리하기 위한 분산 시스템이 널리 알려져 있다. 소위 클라우드 시스템이 분산 시스템의 일 예이다. 이러한 클라우드 시스템 상에서 수행되는 프로그램 프레임워크가 또한 알려져 있는 데, 예를 들어 맵-리듀스(Map-Reduce), 프리겔(Pregel) 등이 있다.

[0005] 프리겔 프레임워크는 다수의 머신으로 구성된 분산 시스템(클라우드 시스템) 상에서 수행할 수 있는 프로그램 모델을 제공하고 특히 그래프의 병렬 처리에 효율적인 프로그램 모델을 제공한다. 프리겔은 정점 중심의(Vertex-centric) BSP(Bulk-Synchronous Parallel) 프로그램 모델을 이용한 그래프 처리 아키텍처를 제공한다.

[0006] 프리겔은 각각의 정점에서 독립적으로 수행되는 프로그램을 정점에 제공하고 각각의 정점에서의 프로그램 수행 완료 후에 재 반복하여 수행할 수 있도록 하는 프로그램 모델을 제공한다. 이 과정에서 각 정점들 사이의 글로벌 동기화를 위한 메커니즘을 프리겔이 또한 제공한다. 각 정점은 또한 자신의 정점의 상태를 변경하거나 에지

를 추가 또는 제거하여 그래프 처리 알고리즘의 결과를 출력하거나 이를 알릴 수 있다.

- [0007] 한편 그래프 처리 알고리즘 중 하나인 매칭 알고리즘이 알려져 있다. 매칭 알고리즘은 일반적으로 최대 매칭을 찾거나 극대 매칭을 찾도록 하는 알고리즘이다. 매칭 알고리즘에서 그 결과로서 출력되는 매칭(M) 세트는 어떠한 두 개의 에지도 하나의 정점을 공유하지 않는 에지들의 집합이다. 하나의 그래프로부터 도출되는 매칭 세트는 널(Null) 집합으로부터 시작하여 다수가 존재한다. 하나의 에지가 매칭 세트에 삽입될 때, 이 하나의 에지를 구성하는 정점들은 (서로) 매칭 되었다고 이하 지칭될 수 있다.
- [0008] 주어진 그래프에서의 모든 매칭 세트 중 일부인 극대(maximal) 매칭 세트(이하 '극대 매칭'이라고도 함)는 그래프로부터의 모든 매칭 세트 중 다른 매칭 세트의 부분 집합이 아닌 집합이다. 따라서 극대 매칭은 다른 에지를 극대 매칭 내에 추가하면 매칭 세트가 구성되지 않는다. 주어진 그래프에서 모든 매칭 세트 중 일부인 최대(maximum) 매칭 세트(이하 '최대 매칭'이라고도 함)는 그래프에서 최대의 에지 수를 가지는 매칭 세트이다.
- [0009] 이러한 매칭(최대 매칭, 극대 매칭)은 다양한 분야에서 활용되고 있다. 예를 들어 매칭은 행렬에서 피벗(pivot)을 찾거나 생물, 화학 분야에 활용되거나 그래프의 축소(coarsen)에 이용되거나 웹 사이트나 SNS로부터 도출되는 그래프 처리에 이용되어질 수 있다. 심지어 매칭은 그래프의 정점으로 표현되는 다수의 사용자 간 1 대 1 연결에 이용될 수도 있다.
- [0010] 매칭 알고리즘을 포함하는 그래프 처리 알고리즘은 앞서 살펴본 바와 같이 그 처리 속도가 느려 대용량의 그래프를 처리하기에 부적당하다. 일 예로 에드몬드(Edmond)에 의해서 제안된 알고리즘(Edmonds, Jack. "Paths, trees, and flowers." Canadian Journal of mathematics 17.3, pp. 449-465, 1965)은 처리할 정점 개수의 4승에 해당하는 처리 속도가 걸리는 것으로 알려져 있다. 다른 예로 칼프(Karp) 및 사이프서(Sipser)에 의해서 제안된 알고리즘(Richard M. Karp and Michael Sipser, "Maximum matchings in sparse random graphs" Proceedings of the 22nd Annual IEEE Symposium of Foundations of Computing, 1981)은 높은 확률로 최대 매칭을 찾는 것으로 알려져 있으나, 반복 별 매칭 선정 과정이 차수(degree) 1에 국한되어 속도가 느린 문제가 존재한다.
- [0011] 특히 기존의 그래프 처리 알고리즘은 클라우드 시스템과 같은 분산 시스템을 고려하고 있지 않다.
- [0012] 이와 같이 프리퀀스 등과 같은 그래프 처리 아키텍처의 분산 시스템을 활용하고 그래프의 특성을 고려하여 극대 매칭과 같은 매칭을 찾기 위한 처리 속도를 향상시킬 수 있도록 하는 그래프 극대 매칭 방법이 필요하다.

발명의 내용

해결하려는 과제

- [0013] 본 발명은, 상술한 문제점을 해결하기 위해서 안출한 것으로서, 프로그램 모델을 활용하여 복수의 머신으로 구성된 분산 시스템에 적용할 수 있는 그래프 극대 매칭 방법을 제공하는 데 그 목적이 있다.
- [0014] 또한 본 발명은 대용량의 그래프에 대해 적용할 수 있는 그래프 극대 매칭 방법을 제공하는 데 그 목적이 있다.
- [0015] 또한 본 발명은 분산 시스템에 적용할 수 있는 그래프 처리 아키텍처이자 프로그램 모델인 프리퀀스를 이용하여 극대 매칭의 처리 속도를 향상시킬 수 있도록 하는 그래프 극대 매칭 방법을 제공하는 데 그 목적이 있다.
- [0016] 또한 본 발명은 그래프로부터 도출되는 그래프 특성을 이용하여 극대 매칭의 처리 속도를 향상시킬 수 있도록 하는 그래프 극대 매칭 방법을 제공하는 데 그 목적이 있다.
- [0017] 또한 본 발명은 그래프의 각각의 정점들을 머신에 할당하고 인접하는 정점들로부터 결정되는 차수(degree)를 활용하여 극대 매칭의 처리 속도를 향상시킬 수 있도록 하는 그래프 극대 매칭 방법을 제공하는 데 그 목적이 있다.
- [0018] 본 발명에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0019] 상기와 같은 목적을 달성하기 위한 그래프 극대 매칭 방법은 (a) 제1 정점(vertex)에 연결된 에지(edge)에 따른 차수(degree)를 포함하는 차수 메시지를 인접한 제2 정점(Vertex)들로 전송하는 단계, (b) 인접한 제2 정점들로

부터 차수 메시지를 수신하는 단계, (c) 수신된 인접한 제2 정점들의 차수 메시지들의 차수들 간 비교로 후보 정점을 결정하는 단계 및 (d) 후보 정점으로 매치 요구 메시지를 전송하는 단계를 포함한다.

[0020] 또한 단계 (c)는 인접한 제2 정점들 중 가장 작은 차수를 가지는 정점을 후보 정점으로 결정한다.

[0021] 또한 그래프 극대 매칭 방법은 (e) 인접한 제2 정점들 중 하나 이상의 제2 정점들로부터 매치 요구 메시지를 수신하는 단계, (f) 후보 정점으로부터 매치 요구 메시지를 수신하였는지를 결정하는 단계 및 (g) 후보 정점으로부터 매치 요구 메시지의 수신시에 매칭 세트를 업데이트하는 단계를 더 포함한다.

[0022] 또한 그래프 극대 매칭 방법은 (h) 제1 정점 및 제2 정점들을 포함하는 그래프로부터 제1 정점과 후보 정점을 연결하는 에지를 제거하여 그래프를 변경하는 단계를 더 포함하고, 매칭 세트는 업데이트에 따라 제1 정점과 후보 정점으로 구성된 에지를 포함하고, 단계 (a) 내지 (g)는 변경된 그래프에 대해서 반복적으로 수행된다.

[0023] 또한 그래프 극대 매칭 방법은 복수의 머신 상에서 수행되며, 그래프 극대 매칭 방법은 단계 (a) 이전에, 제1 정점 및 제2 정점들을 포함하는 그래프를 분할하는 단계 및 분할된 그래프 각각을 복수의 머신에 각각 할당하는 단계를 더 포함하고, 복수의 머신 중 제1 정점을 포함하는 분할된 그래프를 할당받은 머신이 단계 (a) 내지 (g)를 수행한다.

[0024] 또한 그래프 극대 매칭 방법은 분산 처리를 위한 BSP(Bulk-synchronous parallel) 모델을 이용하고 그래프 처리 아키텍처인 프리겔(Pregel) 상에서 수행된다.

[0025] 또한 복수의 머신 각각은 분할된 그래프의 복수의 정점들 각각을 제1 정점으로 하여 단계 (a) 내지 (g)를 수행한다.

[0026] 또한 단계 (a) 내지 (g)의 반복 수행은 매칭 세트의 업데이트로 매칭 세트가 변경된 경우나 그래프의 하나 이상의 정점이 액티브 상태인 경우에 수행한다.

발명의 효과

[0027] 상기와 같은 본 발명에 따른 그래프 극대 매칭 방법은 프로그램 모델을 활용하여 복수의 머신으로 구성된 분산 시스템에 적용할 수 있는 효과가 있다.

[0028] 또한 상기와 같은 본 발명에 따른 그래프 극대 매칭 방법은 대용량의 그래프에 대해 적용할 수 있는 효과가 있다.

[0029] 또한 상기와 같은 본 발명에 따른 그래프 극대 매칭 방법은 분산 시스템에 적용할 수 있는 그래프 처리 아키텍처이자 프로그램 모델인 프리겔을 이용하여 극대 매칭의 처리 속도를 향상시킬 수 있도록 하는 효과가 있다.

[0030] 또한 상기와 같은 본 발명에 따른 그래프 극대 매칭 방법은 그래프로부터 도출되는 그래프 특성을 이용하여 극대 매칭의 처리 속도를 향상시킬 수 있도록 하는 효과가 있다.

[0031] 또한 상기와 같은 본 발명에 따른 그래프 극대 매칭 방법은 그래프의 각각의 정점들을 머신에 할당하고 인접하는 정점들로부터 결정되는 차수(degree)를 활용하여 극대 매칭의 처리 속도를 향상시킬 수 있도록 하는 효과가 있다.

[0032] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0033] 도 1은 분산 시스템의 예시적인 시스템 블록도를 도시한 도면이다.

도 2는 분산 시스템에 입력되는 예시적인 그래프를 도시한 도면이다.

도 3은 분산 시스템에서 수행되고 프리겔 프로그램 모델로 작성된 예시적인 극대 매칭 알고리즘을 도시한 도면이다.

도 4는 그래프를 이용하여 차수 전송 스텝을 설명하기 위한 예시적인 도면이다.

도 5는 그래프를 이용하여 매치 시도 스텝을 설명하기 위한 예시적인 도면이다.

도 6은 그래프를 이용하여 확인 스텝 이후의 과정을 설명하기 위한 예시적인 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0034] 상술한 목적, 특징 및 장점은 첨부된 도면을 참조하여 상세하게 후술 되어 있는 상세한 설명을 통하여 더욱 명
확해 질 것이며, 그에 따라 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 본 발명의 기술적 사상을
용이하게 실시할 수 있을 것이다. 또한, 본 발명을 설명함에 있어서 본 발명과 관련된 공지 기술에 대한 구체적
인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에 그 상세한 설명을 생략하기로 한다.
이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시 예를 상세히 설명하기로 한다.
- [0035] 도 1은 분산 시스템의 예시적인 시스템 블록도를 도시한 도면이다.
- [0036] 도 1에 따르면 분산 시스템은 복수의 머신(100)과 네트워크(200)를 포함한다. 이 분산 시스템은 예를 들어 클라
우드 시스템일 수 있다.
- [0037] 분산 시스템의 각 구성 요소를 간단히 살펴보면, 머신(100)은 내부에 통신 인터페이스, 메모리 및 프로세서를
적어도 구비한다. 머신(100)은 통신 인터페이스를 통해 수신된 프로그램과 데이터를 메모리에 로딩하고 데이터
를 이용하여 프로그램을 수행할 수 있다. 본 발명에 따라 데이터는 바람직하게 그래프를 나타내는 데이터이다.
- [0038] 그래프(G)는 {V,E}로 표현되고 V는 그래프에 포함된 정점(Vertex)들의 집합을 의미하고 E는 정점과 정점 사이에
연결된 에지의 집합을 의미한다. 머신(100)은 예를 들어 분산 시스템에 이용될 수 있는 PC, 서버, 그 외 별도로
제작된 셋탑 박스 등일 수 있다.
- [0039] 분산 시스템상에서 프로그램을 수행하기 위해서 복수의 머신(100) 중 하나의 머신(100)은 마스터(Master) 머신
(100)으로 설정될 수 있고, 나머지 머신(100)들은 워크 머신(100)들로 설정될 수 있다.
- [0040] 마스터 머신(100)은 프로그램과 전체 데이터(즉 전체 그래프)를 분산 시스템의 사용자로부터 수신하고, 프로그
램과 전체 데이터 중 일부를 분할하여 워크 머신(100)들로 할당하고 분배할 수 있다. 또한 마스터 머신(100)은
워크 머신(100)들 간의 글로벌 동기화(S5)를 수행할 수 있다. 따라서 마스터 머신(100)은 각각의 워크 머신
(100)들에서 한 스텝(프리겔의 슈퍼스텝(superstep))의 완료로 모두 확인한 후에 다음 스텝(슈퍼스텝)을 수행하
도록 워크 머신(100)을 제어할 수 있다.
- [0041] 또한 마스터 머신(100)은 입력된 그래프를 분할하고 분할된 그래프를 병렬적으로 수행되도록 워크 머신(100) 각
각에 분할된 그래프 각각을 할당할 수 있다. 이러한 분할과 할당은 워크 머신(100)에서 수행되는 극대 매칭 알
고리즘의 수행 이전에 이루어진다. 이에 따라 각각의 워크 머신(100)은 마스터 머신(100)으로부터 수행할 프
로그램과 프로그램에서 이용될 데이터(전체 그래프의 일부)를 수신하고 해당 프로그램을 분할된 일부 그래프에 대
해서 수행할 수 있다.
- [0042] 본 발명에 따른 분산 시스템은 바람직하게 프리겔(Pregel)로 된 프로그램 (모델)을 수행할 수 있다. 프리겔은
분산 또는 병렬 처리를 위한 BSP(Bulk-synchronous parallel) 모델을 이용하는 그래프 처리 아키텍처이다. 프리
겔은 그래프의 각 정점에서 수행되는 프로그램을 정의하고 각 프로그램이 정점이 할당된 머신(100)(워크 머신
(100))에서 수행된다. 프리겔은 반복문을 이용하여 각 정점에서 한 스텝(superstep)의 수행이 모두 완료되면 다
음 반복문을 수행하도록 각 정점을 제어할 수 있다. 또한 프리겔은 정점에 의해 수행되는 프로그램에 의해서 또
는 정점과 연동하는 마스터 머신(100)에 의해서 입력된 전체 그래프를 변경할 수 있다. 이러한 그래프의 변경은
적어도 에지의 제거를 포함한다.
- [0043] 프리겔 프로그램 모델에 따라, 복수의 머신(100) 중 워크 머신(100) 각각은 하나 또는 복수의 정점을 할당받아
해당 정점(들)에 대응하는 프로그램을 수행한다. 따라서 복수의 머신(100)은 다른 머신(100) 또는 다른 정점과
는 별개로 독립적으로 프로그램을 수행할 수 있다. 이러한 분산 처리 시스템과 분산 처리 시스템에서 이용할 수
있는 프리겔 프로그램 모델(또는 프레임워크)은 대용량의 그래프를 고속으로 처리할 수 있도록 하는 환경을 제
공한다.
- [0044] 워크 머신(100)에서 수행되는 극대 매칭 알고리즘은 도 3 이하를 통해 살펴보도록 한다.
- [0045] 네트워크(200)는 메시지를 송수신할 수 있도록 한다. 네트워크(200)는 예를 들어 유선 네트워크, 무선 네트워크
또는 유선 및 무선 네트워크의 조합으로 구성된다. 네트워크(200)는 예를 들어 IP(Internet Protocol)에 따른

광대역 네트워크이거나 근거리 네트워크일 수 있다.

- [0046] 네트워크(200)는 마스터 머신(100)과 워크 머신(100) 사이에서 송수신되는 각종 메시지를 주소를 이용하여 송신하고 수신할 수 있도록 구성된다.
- [0047] 도 2는 분산 시스템에 입력되는 예시적인 그래프를 도시한 도면이다.
- [0048] 도 2에 따른 예시적인 그래프는 사용자에게 의해 마스터 머신(100)에 입력될 수 있다. 마스터 머신(100)은 입력된 전체 그래프를 분할하고 분할된 그래프를 나타내는 데이터와 사용자로부터 수신된 프로그램을 또는 사용자 입력으로부터 추출된 프로그램을 워크 머신(100)들에 할당한다.
- [0049] 예를 들어 마스터 머신(100)은 도 2에 도시된 바와 같이 부분 그래프 $\{V_2, V_3\}$, $\{V_1, V_2\}$, (V_2, V_6) , (V_2, V_3) 를 머신(100) 1에 할당하고 머신(100) 2 내지 머신(100) 4에 각각의 정점들과 에지들을 할당할 수 있다. 이러한 할당은 예를 들어 부분 그래프에 대한 데이터의 해당 머신(100)으로의 전송과 각 정점에서 수행될 프로그램의 전송을 수반한다. 부분 그래프의 정점과 에지는 정점의 식별자(Id)로 식별될 수 있다. 이 식별자를 이용하여 인접하는 정점으로 각종 메시지를 머신(100)에 의해서 또는 각 정점들에 의해서 송수신할 수 있다.
- [0050] 도 2에서는 설명의 간단한 이해를 위해서 8개의 정점으로 구성된 그래프를 도시하였으나, 일반적으로 이 그래프는 수십만에서 수백만 이상의 정점들을 포함할 수 있다. 수백만 이상의 정점들의 처리를 위해 또한 머신(100)들은 4개나 8개 뿐 아니라 수십 개 또는 수백 개 이상으로 구성될 수도 있다.
- [0051] 이러한 할당을 통해 각각의 머신(100)은 자신들에게 할당된 정점들에 대하여 프로그램을 병렬로 그리고 독립적으로 수행할 수 있다. 할당된 머신(100)들에서 모든 정점들의 프로그램 수행으로 입력된 그래프의 모든 정점들에 대해서 극대 매칭 알고리즘이 각 스텝별로 수행될 수 있다.
- [0052] 도 3은 분산 시스템에서 수행되고 프리겔 프로그램 모델로 작성된 예시적인 극대 매칭 알고리즘을 도시한 도면이다. 도 3에서는 이 극대 매칭 알고리즘이 슈도(Pseudo) 코드로 표현되어 있다.
- [0053] 극대 매칭 알고리즘은 마스터 머신(100)과 워크 머신(100) 상에서 수행되고 각 정점들에 대해서 수행된다. 예를 들어 워크 머신(100)에 2개의 정점들이 할당된 경우에 극대 매칭 알고리즘은 2개의 각 정점들에 대해 각각 독립적으로 수행된다. 워크 머신(100)에 두 개 이상의 프로세서가 내장된 경우에는 각 프로세서에 각 정점을 할당하여 수행하도록 구성되고 하나의 프로세서만이 내장된 경우에는 각 정점에 대한 수행은 순차적으로 이루어질 수 있다.
- [0054] 극대 매칭 알고리즘을 살펴보면, 극대 매칭 알고리즘은 초기화 스텝(S1), 차수(degree) 전송 스텝(S2), 매치 시도 스텝(S3), 및 확인 스텝(S4)을 적어도 포함하고 마스터 머신(100)과 또는 다른 워크 머신(100)과 글로벌 동기화(S5)를 수행할 수 있다.
- [0055] 각각의 스텝을 간단히 살펴보면, 초기화 스텝(S1)은 매칭 세트(M)를 빈 세트로 초기화하고 각각의 정점들의 동작 상태를 차수 전송 스텝(S2)으로 설정한다. 초기화 스텝(S1)의 매칭 세트 초기화는 바람직하게 마스터 머신(100)에 의해서 수행된다. 매칭 세트는 소위 글로벌 변수일 수 있다.
- [0056] 마스터 머신(100)은 이러한 초기화 스텝(S1)에 앞서 또는 병행하여 극대 매칭을 수행하기 위한 그래프를 사용자로부터 수신하고 수신된 그래프를 복수의 부분(Partial) 그래프로 분할한다. 마스터 머신(100)은 분할된 그래프(부분 그래프)를 워크 머신(100)에 할당한다. 이 과정에서 부분 그래프를 나타내기 위한 데이터(V, E)와 프로그램이 각각의 워크 머신(100)으로 분배될 수 있다. 이후 각각의 워크 머신(100)은 할당된 그래프와 프로그램을 이용하여 각 정점에 대한 매칭을 수행한다.
- [0057] 차수 전송 스텝(S2), 매치 시도 스텝(S3) 및 확인 스텝(S4)은 워크 머신들(100)에서 수행되고 워크 머신들(100)에 할당된 그래프의 각 정점에 대해서 수행된다. 그리고 다수의 워크 머신(100)을 이용하여 입력된 그래프의 모든 정점들에 대해서 매칭이 수행된다. 차수 전송 스텝(S2), 매치 시도 스텝(S3), 확인 스텝(S4)은 로컬 스텝(도 3의 'step')에 의해서 구분된다. 만일 워크 머신(100)에서 수행되는 각각의 정점이 다른 정점과 매칭된 경우에는 각각의 정점의 수행 프로그램은 정지(halt) 상태로 전이하여 더 이상 반복문(iteration, 도 3의 while 루프 내)을 수행하지 않게 구성된다. 차수 전송 스텝(S2), 매치 시도 스텝(S3) 및 확인 스텝(S4) 등으로 반복문이 수행되는 상태를 이하 액티브(Active) 상태라고 하고 더 이상 반복문 내부를 수행하지 않는 상태를 정지 상

태라고 한다.

- [0058] 차수 전송 스텝(S2)(도 3의 5 ~ 7 라인 참조)에서 워크 머신(100)에 할당되어 각 스텝을 수행하는 정점(이하 '제1 정점'이라고도 함)은 자신의 차수(degree)를 포함하는 차수 메시지(degree message)를 에지(edge)로 인접한 정점들(이하 '제2 정점'이라고도 함)로 전송한다. 차수 전송 스텝(S2)에 전송되는 차수 메시지는 바람직하게는 자신의 식별자인 제1 정점의 식별자와 자신의 차수를 포함한다. 인접하는 제2 정점들은 제1 정점과 동일한 워크 머신(100) 상에 할당된 정점이거나 다른 워크 머신(100)상에 할당된 정점일 수 있다. 다른 워크 머신(100)상에 할당된 정점인 경우, 차수 메시지는 네트워크(200)를 통해 전송된다. 차수 메시지를 포함하는 메시지의 전송과 수신은 바람직하게 프리켈에 의한 제어로 또는 프리켈로부터 수신된 데이터(예를 들어 인접하는 제2 정점들의 식별자)로 이루어진다.
- [0059] 차수 메시지의 전송 이후에 내부 로컬 스텝을 매치 시도 스텝(S3)(도 3의 7 라인 참조)으로 변경하고 마스터 머신(100)과 글로벌 동기화(S5)(도 3의 23 라인 참조)한다. 이후 제1 정점은 마스터 머신(100)으로부터의 시그널에 따라 다시 반복문을 수행할 수 있다.
- [0060] 도 4는 그래프를 이용하여 차수 전송 스텝(S2)을 설명하기 위한 예시적인 도면이다. 도 4의 예시적인 도면은 도 2의 그래프를 이용하여 설명된다.
- [0061] 도 4에서 알 수 있는 바와 같이, 각각의 정점들은 반복문에 의해서 그리고 내부의 로컬 스텝에 의해서 서로 동기화되어 서로 자신의 식별자와 자신의 차수를 포함하는 차수 메시지를 인접하는 정점들로 전송한다.
- [0062] 예를 들어 정점 V1은 3개의 에지가 있어 차수가 3이다. 정점 V1은 자신의 식별자와 차수 3을 포함하는 차수 메시지를 V2, V4, V5 등으로 전송하고 정점 V2는 자신의 식별자와 차수 3을 포함하는 차수 메시지를 V1, V3, V6로 전송한다. 다른 정점들 역시 동일하게 차수 메시지를 전송한다. 모든 정점들이 차수 메시지의 전송을 완료하면 글로벌 동기화(S5)에 따른 마스터 머신(100)의 제어에 따라 다음 반복문(도 3의 라인 4 참조)이 수행된다.
- [0063] 다음 반복문에서 모든 정점들은 로컬 스텝(도 3의 'step')의 상태 값에 따라 매치 시도 스텝(S3)으로 전이하고, 차수 메시지를 수신할 수 있다. 이러한 메시지의 수신은 프리켈 프레임워크에 의해서 제어된다. 모든 제1 정점들이 인접하는 제2 정점들로 차수 메시지를 차수 전송 스텝(S2)에서 전송하였으므로 모든 제1 정점들은 매치 시도 스텝(S3)에서 인접하는 제2 정점들로부터 차수 메시지를 수신한다.
- [0064] 매치 시도 스텝(S3)(도 3의 8 ~ 14 라인 참조)을 수행하는 제1 정점은 인접하는 제2 정점들로부터 수신된 차수 메시지들의 차수들의 비교로 매칭을 위한 후보 정점을 결정(도 3의 9 ~ 12 라인 참조)한다. 제1 정점은 예를 들어 차수들 중 가장 작은 차수를 가진 제2 정점을 후보 정점으로 결정할 수 있다.
- [0065] 그래프의 극대 매칭을 위해서는 최대한 가장 작은 차수를 가진 에지부터 고려할 필요가 있다. 극단적으로는 차수 1부터 고려할 수 있으나 가장 낮은 차수를 가진 에지를 고려하는 것이 처리 속도 면에서 유리하고 반복의 회수를 줄일 수 있도록 한다. 이에 따라 대용량의 그래프의 극대 매칭의 그래프를 빠른 속도로 찾을 수 있다.
- [0066] 이후 매치 시도 스텝(S3)에서 제1 정점은 인접하는 제2 정점들 중 차수에 따라 선택된 후보 정점으로 매치 요구 메시지(match request message)를 전송하고 다음번 로컬 스텝을 확인 스텝(S4)으로 설정(도 3의 13, 14 라인 참조)한다. 매치 요구 메시지는 제1 정점 자신의 식별자를 포함할 수 있다.
- [0067] 도 5는 그래프를 이용하여 매치 시도 스텝(S3)을 설명하기 위한 예시적인 도면이다. 도 5의 예시적인 도면은 도 2의 그래프를 이용하여 설명된다.
- [0068] 도 5에서 알 수 있는 바와 같이 각각의 제1 정점들은 가장 낮은 차수를 가진 인접하는 정점으로 매치 요구 메시지를 전송한다. 예를 들어 정점 V1은 인접하는 정점의 차수 중 가장 낮은 차수를 가진 V2(여기서는 동일한 차수로 인해 랜덤 또는 식별자가 낮은 순으로 선택)로 매치 요구 메시지를 전송한다. V2는 가장 낮은 차수를 가진 V3로 매치 요구 메시지를 전송하고 V3는 V2로 매치 요구 메시지를 전송한다. 다른 정점들 역시 동일하게 매치 요구 메시지를 전송한다.
- [0069] 매치 요구 메시지는 제1 정점에서 인접 제2 정점의 차수에 따라서 선택된 후보 정점으로 전송되는 데, 이는 그래프에서 에지의 연결 토폴로지를 고려한 선택이다. 낮은 차수의 에지(즉 제2 정점의 에지)를 가진 정점을 먼저 선택함으로써 극대 또는 최대 매칭이 이루어지도록 한다. 나아가 모든 제1 정점에서 낮은 차수의 에지로 후보 정점을 결정함으로써 매칭을 위한 병렬성을 높이고 이에 따라 극대 매칭의 처리 속도를 향상시킬 수 있다.
- [0070] 이러한 매치 시도 스텝(S3)은 그래프의 모든 정점들에 대해서 할당된 워크 머신(100)상에서 수행되고 글로벌 동

기화(S5)에 따라 각각의 정점들은 다음 반복문에서 확인 스텝(S4)을 수행할 수 있다. 글로벌 동기화(S5) 이후에 각각의 정점들(제1 정점들)은 인접하는 제2 정점들 중 하나 이상의 제2 정점으로부터 매치 요구 메시지를 수신한다. 이러한 매치 요구 메시지는 제2 정점에 의해서 결정된 후보 정점에 전송된 매치 요구 메시지이다.

[0071] 확인 스텝(S4)(도 3의 15 ~ 22 라인 참조)을 수행하는 제1 정점은 인접하는 제2 정점들 중 매치 시도 스텝(S3)에서 결정된 후보 정점으로부터 매치 요구 메시지를 수신하였는지를 결정한다. 이러한 결정은 매치 시도 스텝(S3)에서 저장된 후보 정점의 식별자와 매치 요구 메시지의 식별자의 비교로 이루어질 수 있다.

[0072] 만일 후보 정점의 식별자와 매치 요구 메시지의 특정 식별자가 동일한 경우에, 제1 정점과 후보 정점인 제2 정점은 서로 매칭을 요구한 것으로 판단되고 이 두 정점을 매칭 세트에 삽입할 수 있다.

[0073] 후보 정점으로부터 매치 요구 메시지가 수신된 경우, 확인 스텝(S4)의 제1 정점은 제1 정점과 후보 정점인 제2 정점을 포함하는 에지를 생성하고 에지를 매칭 세트(M)에 삽입하여 매칭 세트를 업데이트(도 3의 17 ~ 19 라인 참조)한다. 또한 제1 정점은 이 제1 정점이 매칭된 것으로 설정하여 액티브 상태에서 정지 상태로 전이하고 이후의 반복문의 수행을 정지(도 3의 20 라인 및 4 라인 참조)한다.

[0074] 물론 매칭이 된 경우에, 적어도 두 개의 정점(제1 정점)에 의해서 매칭 세트의 업데이트가 이루어지나 둘 중 하나의 에지는 생략하도록 구성될 수 있다.

[0075] 만일 후보 정점으로부터 매치 요구 메시지가 수신되지 않은 경우에는 로컬 스텝의 상태를 차수 전송 스텝(S2)(도 3의 22 라인 참조)으로 설정하여 제1 정점이 재차 매칭을 시도한다.

[0076] 확인 스텝(S4) 이후에 제1 정점들은 글로벌 동기화(S5)되어 확인 스텝(S4)의 수행을 완료한다. 확인 스텝(S4)의 수행 완료 후에, 마스터 머신(100)은 매칭 세트의 변경을 확인하고 매칭 세트 내에 포함된 에지를 또는 추가된 에지를 그래프로부터 제거하여 그래프를 변경한다.

[0077] 도 6은 그래프를 이용하여 확인 스텝(S4) 이후의 과정을 설명하기 위한 예시적인 도면이다. 도 6의 예시적인 도면은 도 2의 그래프를 이용하여 설명된다.

[0078] 정점 V1은 V2로 매치 요구 메시지를 전송하였으나 V2로부터 매치 요구 메시지를 수신하지 못하여 매칭되지 못하였다. V2와 V3는 서로 매치 요구 메시지를 송수신하여 매칭되었고, V4와 V7는 서로 매치 요구 메시지를 송수신하여 매칭되었다. 또한 V6와 V8 역시 매칭되었다.

[0079] 이에 따라 매칭 세트는 {(V2, V3), (V4, V7), (V6, V8)}으로 변경되고, 매칭 세트에 포함된 또는 추가된 에지는 그래프로부터 제거된다. 또한 매칭 세트의 정점에 연결된 다른 에지가 또한 그래프로부터 제거된다. 이에 따라 매칭되지 못한 V1과 V5는 에지의 차수가 1로 변경된다. 이러한 그래프의 변경은 모든 정점들에서의 확인 스텝(S4) 완료 이후에 마스터 머신(100)에 의해서 수행될 수 있다.

[0080] 그래프의 변경 이후에, 마스터 머신(100)은 변경된 그래프를 나타내는 데이터(V,E)를 워크 머신(100)으로 전달하고 워크 머신(100)으로 하여금 변경된 그래프에 대한 매칭을 반복적으로 수행하도록 한다. 각각의 정점들에 대해서 할당받은 워크 머신(100)은 수신된 분할 그래프의 데이터를 이용하여 각각의 정점들에 대해서 매칭을 수행한다.

[0081] 물론 이미 매칭된 정점들은 도 3의 반복문 내로 진입하지 못하는 정지 상태에 있으므로 더 이상의 매칭 시도가 이루어지지 않는다. 그러나 도 6에서 알 수 있는 바와 같이 정점 V1(차수 1로 변경)과 V6(차수 1로 변경)는 차수 전송 스텝(S2), 매치 시도 스텝(S3), 확인 스텝(S4)을 재차 수행하여 서로 매칭된다.

[0082] 마스터 머신(100)은 도 3의 극대 매칭 알고리즘의 종료를 결정할 수 있다. 예를 들어 마스터 머신(100)은 그래프의 변경으로 그래프에 에지가 존재하지 않는 경우, 모든 정점들이 정지 상태에 있는 경우, 차수 전송 스텝(S2), 매치 시도 스텝(S3), 확인 스텝(S4)을 통한 반복에서 이전 매칭 세트와 이후 매칭 세트가 동일한 경우, 극대 매칭 알고리즘의 수행을 종료할 수 있다.

[0083] 즉 다른 말로, 마스터 머신(100)은 그래프의 변경으로도 아직 에지가 존재하거나 하나 이상의 정점들이 아직 액티브 상태이거나 3개의 스텝의 수행 전과 후에 매칭 세트가 다른 경우에 3 스텝으로 구성된 극대 매칭 알고리즘을 반복적으로 수행한다.

[0084] 극대 매칭 알고리즘의 전체 수행 완료 후에, 마스터 머신(100)은 그 결과를 출력할 수 있다. 출력되는 결과는 예를 들어 모든 정점들과 정점들 사이에서 선택된 에지를 포함하는 극대 매칭을 나타내는 결과 그래프일 수 있다. 이러한 결과 그래프는 이미지로 표현되거나 텍스트로 출력된다.

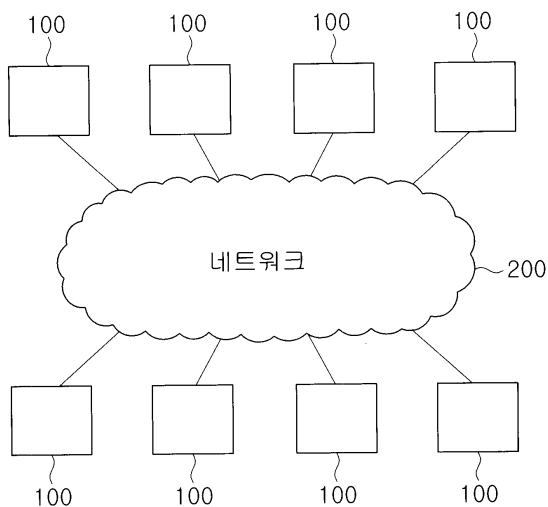
- [0085] 이상 도 3을 통해 살펴본 바와 같이 본 극대 매칭 알고리즘은 복수의 머신(100)으로 구성된 분산 시스템에서 수행되고 각 머신(100)들에 할당된 그래프를 이용하여 병렬적으로 또는 독립적으로 수행된다. 또한 본 극대 매칭 알고리즘은 바람직하게 그래프 처리 아키텍처로서 분산 처리를 위한 BSP 모델을 이용하는 프리켈 프레임워크 상에서 수행될 수 있다.
- [0086] 또한 본 극대 매칭 알고리즘은 인접하는 정점의 차수를 이용하고 낮은 차수를 가진 인접 정점과의 매칭을 시도하여 극대 매칭 처리를 위한 속도를 향상시킬 수 있다.
- [0087] 본 극대 매칭 알고리즘은 최대 매칭 대비 약 98.4%의 정확도로 매칭을 찾을 수 있었고, 칼프 및 사이프서의 제안 알고리즘에 비해 약 100 배 이상의 수행 속도 향상을 가져올 수 있다. 또한 본 극대 매칭 알고리즘은 클라우드 시스템에서의 머신(100)들의 확장에 따라 용이하게 확장될 수 있는 구조를 제공한다.
- [0088] 이상에서 설명한 본 발명은, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 있어 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하므로 전술한 실시 예 및 첨부된 도면에 의해 한정되는 것이 아니다.

부호의 설명

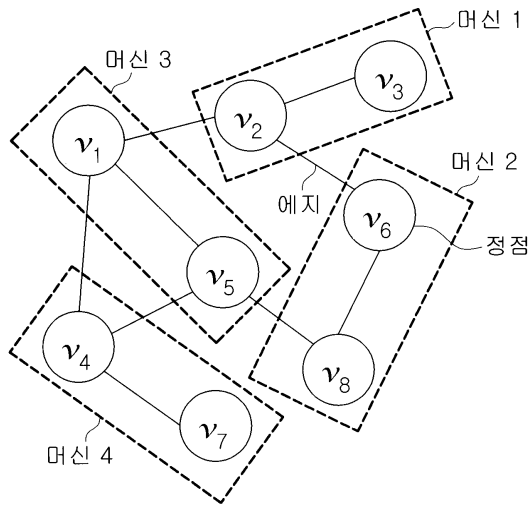
- [0089] 100 : 머신 200 : 네트워크

도면

도면1



도면2



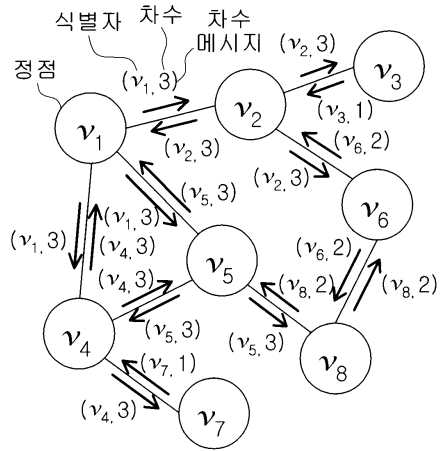
도면3

```

1: Initialize: .....
2:   Result matching set  $M \leftarrow \phi$ ;
3:   Set step as Degree Broadcast; ..... } 초기화 스텝(S1)
4: while this vertex is not Matched do
5:   if step is Degree Broadcast then .....
6:     Send degree messages to adjacent vertices;
7:     Set step as Match Trial; ..... } 차수 전송 스텝(S2)
8:   else if step is Match Trial then .....
9:     for all Received messages do
10:      Find a vertex which contains the lowest degree
11:    end for
12:    candidate mate  $\leftarrow$  sender of the lowest degree vertex;
13:    Send a match request message to candidate mate;
14:    Set step as Confirm; ..... } 매치 시도 스텝(S3)
15:   else if step is Confirm then .....
16:     for all Received messages do
17:       if Sender of the message == candidate mate then
18:          $e \leftarrow$  edge (this vertex, candidate mate);
19:          $M \leftarrow M \cup e$ ;
20:         Set this vertex as Matched;
21:       end for
22:       Set step as Degree Broadcast; ..... } 확인 스텝(S4)
23:   Sync for superstep; } 글로벌 동기화(S5)
24: end while

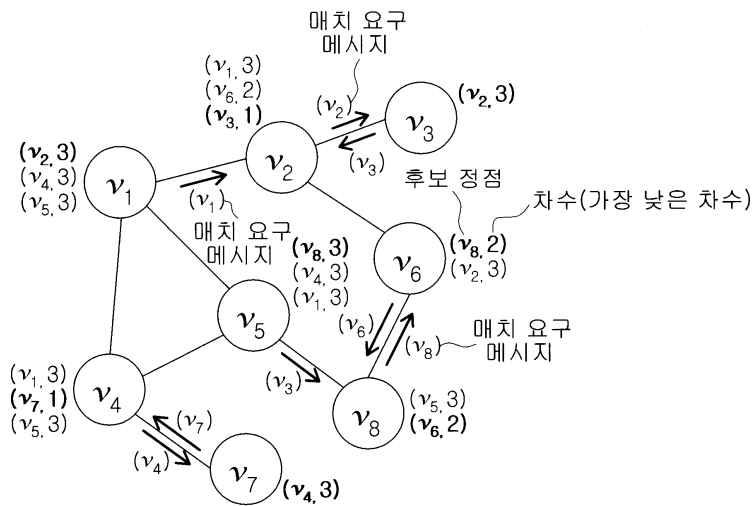
```

도면4



<차수 전송 스텝>

도면5



<매치 시도 스텝>

도면6

