

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2017년 5월 4일 (04.05.2017)



(10) 국제공개번호
WO 2017/073841 A1

- (51) 국제특허분류:
G06F 17/30 (2006.01)
- (21) 국제출원번호: PCT/KR2015/014136
- (22) 국제출원일: 2015년 12월 23일 (23.12.2015)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
10-2015-0149158 2015년 10월 27일 (27.10.2015) KR
- (71) 출원인: 삼성에스디에스 주식회사 (SAMSUNG SDS CO.,LTD.) [KR/KR]; 05510 서울시 송파구 올림픽로 35길 125 (삼성 SDS 타워, West Campus), Seoul (KR).
- (72) 발명자: 강성문 (KANG, Seong Moon); 05510 서울시 송파구 올림픽로 35길 125 (삼성 SDS 타워, West Campus), Seoul (KR).
- (74) 대리인: 특허법인 가산 (KASAN IP & LAW FIRM); 06719 서울시 서초구 남부순환로 2423 한원빌딩 7층, Seoul (KR).

(81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

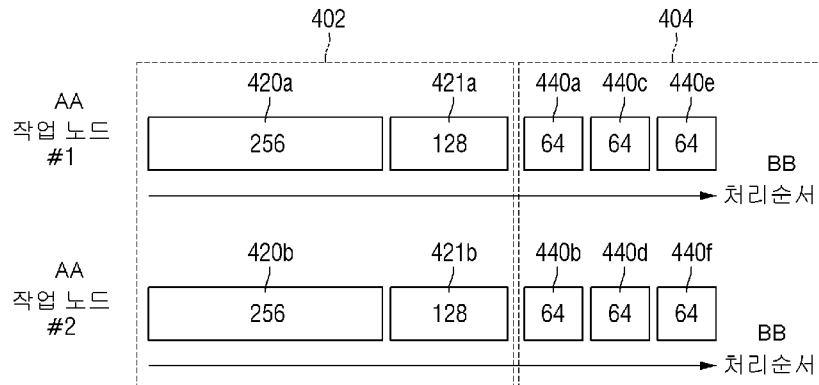
(84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제 21 조(3))

(54) Title: METHOD FOR DIVIDING LARGE-CAPACITY FILE INTO BLOCKS FOR DISTRIBUTED PROCESSING AND APPARATUS THEREOF

(54) 발명의 명칭: 분산 처리를 위한 대용량 파일의 블록화 방법 및 그 장치



AA ... Work node
BB ... Processing sequence

(57) Abstract: Provided is a method for dividing, into blocks, a file to be processed in a distributed manner by a plurality of work nodes belonging to a distributed processing system. A method for dividing a large-capacity file into blocks according to an embodiment of the present invention comprises the steps of: classifying the file into a first area and a second area, the data of which is processed later than the data of the first area; dividing the first area into blocks having various sizes; and dividing the second area into blocks having a fixed size.

(57) 요약서: 분산 처리 시스템에 속한 복수의 작업 노드에 의하여 분산 처리 되는 파일의 블록화 방법이 제공된다. 본 발명의 일 실시예에 따른 대용량 파일의 블록화 방법은, 상기 파일을 제 1 영역 및 상기 제 1 영역의 데이터보다 나중에 처리되는 제 2 영역으로 구분하는 단계, 상기 제 1 영역을 다양한 사이즈의 블록들로 분할하는 단계, 및 상기 제 2 영역을 고정 사이즈의 블록들로 분할하는 단계를 포함한다.



WO 2017/073841 A1

명세서

발명의 명칭: 분산 처리를 위한 대용량 파일의 블록화 방법 및 그 장치

기술분야

- [1] 본 발명은 분산 처리를 위한 대용량 파일의 블록화 방법 및 그 장치에 관한 것이다. 보다 자세하게는, 블록화 된 대용량 파일을 분산 처리 하는 작업 노드들이, 상기 대용량 파일을 실질적으로 동시에 처리 완료 할 수 있도록 대용량 파일을 블록화 하는 방법 및 그 장치에 관한 것이다.

배경기술

- [2] 대용량 데이터를 분산 처리하는 기술이 제공되고 있다. 또한, 대용량 데이터를 포함하는 파일을 서로 다른 컴퓨팅 장치를 통하여 분산 저장하는 기술이 제공되고 있다.
- [3] 예를 들어, 널리 알려진 대용량 데이터 처리 플랫폼인 하둡(hadoop)의 분산 파일 처리 시스템인 HDFS(Hadoop Distributed File System)에서는, 대용량 데이터가 블록 단위로 분할되어 클러스터링 된 데이터 노드(data node)에 분산 저장된다. 또한, 분산 저장된 각 블록의 메타 정보가 네임 노드(name node)에 저장된다. 상기 메타 정보는, 예를 들어, 각 블록의 저장 위치 정보를 포함할 수 있다. 고정 사이즈로 블록을 형성하는 경우, 데이터 사이즈가 증가할 수록 블록의 개수도 늘어나고 그에 따라 메타 정보의 사이즈도 늘어날 것이다. 아래의 표 1은 파일 사이즈에 따라 파일의 분산 저장에 수반되는 메타 정보 사이즈의 증가를 표시한다.

- [4] [Table 1]

파일 사이즈	블록 개수(블록 당 64MB)	메타 정보 사이즈 (단위, MB)(블록 당 150K 바이트)
1GB	16	0.00
10GB	160	0.02
100GB	1,600	0.23
1TB	16,384	2.34
10TB	163,840	23.44
100TB	1,638,400	234.38
1PB	16,777,216	2,400.00

- [5] 상기 표 1에 표시된 바와 같이, 예를 들어 1 PB 사이즈의 파일이라면, 16,777,216개의 블록으로 나누어 저장되고, 그에 따라 메타 정보 사이즈만 2400MB(2.34GB)에 달하는 것을 알 수 있다. 메타 정보는 빈번하게 액세스

되어야 하므로, 메모리 상에 로드 되어야 한다. 약 2.34GB에 달하는 데이터를 메모리에 로드 해두고 운용하는 것은 상당히 부담이 되는 일이다. 분산 저장에 따른 메타 정보 운용상의 부담뿐만 아니라, 분산 처리에 따른 각 블록 처리 태스크(task)의 생성 및 운용상의 부담도 발생할 것이다. 각각의 태스크 처리 이력이 관리되어야 하기 때문이다.

[6] 그렇다고 하여, 블록 사이즈를 무작정 늘릴 수도 없는 일이다. 분산 처리에 따른 작업 분산의 효과가 떨어지기 때문이다.

[7] 따라서, 작업 분산의 효과를 유지하면서도, 파일 사이즈가 증가할수록 블록의 개수가 증가하는 수준을 억제할 수 있는 효율적인 파일 블록화 방법 및 그 방법을 활용하는 대용량 파일의 분산 처리 관리 방법의 제공이 요청된다. 본 명세서에서, 블록화(blocking)는 파일을 블록 단위로 분할하는 것을 가리킨다.

발명의 상세한 설명

기술적 과제

[8] 본 발명이 해결하고자 하는 기술적 과제는, 작업 분산의 효과를 유지하면서도, 파일 사이즈가 증가할수록 블록의 개수가 증가하는 수준을 억제할 수 있는 효율적인 파일 블록화 방법 및 그 장치를 제공하는 것이다.

[9] 본 발명이 해결하고자 하는 다른 기술적 과제는, 종래 기술에 비하여 더 적은 개수의 블록으로 분할된 대용량 데이터를 분산 처리하면서도, 각 작업 노드가 할당된 블록에 대한 처리(processing)를 최대한 동시에 완료하도록 분산 처리를 관리하는 방법 및 그 장치를 제공하는 것이다.

[10] 본 발명이 해결하고자 하는 또 다른 기술적 과제는, 분산 처리를 위한 대용량 파일의 블록화 과정에 분산 처리 시스템의 작업 환경을 자동으로 반영함으로써, 각 작업 노드가 할당된 블록에 대한 처리(processing)를 최대한 동시에 완료하도록 분산 처리를 관리하는 방법 및 그 장치를 제공하는 것이다.

[11] 본 발명의 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급되지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명의 기술분야에서의 통상의 기술자에게 명확하게 이해 될 수 있을 것이다.

과제 해결 수단

[12] 상기 기술적 과제를 해결하기 위한 본 발명의 일 실시예에 따른 분산 처리 시스템에 속한 복수의 작업 노드에 의하여 분산 처리 되는 파일의 블록화 방법은, 상기 파일을 제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되는 제2 영역으로 구분하는 단계와, 상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계와, 상기 제2 영역을 고정 사이즈의 블록들로 분할하는 단계를 포함한다.

[13] 몇몇 실시예들에서, 상기 제1 영역은 각 사이즈의 블록을 M개씩 포함한다. (M은 상기 작업 노드의 개수) 이 때, 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계는, 상기 파일의 제일 앞에서부터, 제1 사이즈의 블록을 M개씩

형성하는 단계(M은 상기 작업 노드의 개수)와, 블록이 형성되지 않은 영역에, 상기 제1 사이즈보다 작은 제2 사이즈의 블록을 M개씩 형성하는 단계와, 블록이 형성되지 않은 영역에, 직전 형성 블록의 사이즈보다 작은 사이즈의 블록을 M개씩 형성하는 단계를, 상기 작은 사이즈가 상기 고정 사이즈보다 큰 것을 조건으로 반복하는 단계를 포함할 수 있다.

- [14] 몇몇 실시예들에서, 상기 제2 영역의 사이즈는, 상기 작업 노드의 개수 및/또는 각 작업 노드에서 실행되는 데이터 분산 처리용 병렬 프로세스의 개수 및/또는 각 작업 노드의 성능이 동일할 수록 작게 설정되거나, 각 작업 노드의 운영 비용이 낮을수록 작게 설정되는 작업 분산 효과 파라미터를 반영하여 결정될 수 있다.
- [15] 일 실시예에서, 상기 분산 처리 시스템에 속한 복수의 작업 노드에 의하여 분산 처리 되는 파일의 블록화 방법은, 상기 파일의 사이즈가 상기 결정된 제2 영역의 사이즈 보다 큰 경우에 한하여, 상기 구분하는 단계, 상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계, 및 상기 제2 영역을 동일 사이즈의 블록들로 분할하는 단계를 수행하는 단계를 더 포함할 수 있다.
- [16] 몇몇 실시예들에서, 상기 제1 영역의 블록은 최초 할당 받은 작업 노드에 의하여 처리되어야 하는 블록이고, 상기 제2 영역의 블록은 최초 할당 받은 작업 노드가 다른 작업 노드에게 이관할 수도 있는 블록이다.
- [17] 다른 몇몇 실시예들에서는, 상기 제1 영역의 블록 중 일부는 최초 할당 받은 작업 노드에 의하여 처리되어야 하는 블록인 반면 다른 일부는 최초 할당 받은 작업 노드가 다른 작업 노드에게 이관할 수 있는 블록이고, 상기 제2 영역의 블록은 최초 할당 받은 작업 노드가 다른 작업 노드에게 이관할 수도 있는 블록이다.
- [18] 상기 기술적 과제를 해결하기 위한 본 발명의 다른 실시예에 따른 대용량 파일의 분산 처리 방법은, 다양한 사이즈의 블록들로 분할된 제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되고 고정 사이즈의 블록들로 분할된 제2 영역을 가지는 파일을 제공받는 단계와, 제1 영역의 블록들 및 제2 영역의 블록들을 분산 처리 시스템에 속한 복수의 작업 노드들에 균등하게 나누어 할당하는 단계와, 상기 작업 노드들의 블록 처리 현황을 모니터링 하여, 각 작업 노드들의 미처리 블록 현황에 따라 부하 리밸런싱 작업을 수행하는 단계를 포함한다. 몇몇 실시예들에서, 상기 부하 리밸런싱 작업을 수행하는 단계는, 상기 제2 영역의 블록 만을 상기 부하 리밸런싱 대상으로 하는 단계를 포함할 수 있다.
- [19] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 실시예에 따른 대용량 파일의 분산 처리 관리 장치는, 하나 이상의 프로세서와, 상기 프로세서에 의하여 수행 되는 컴퓨터 프로그램이 로드(load)된 메모리와, 분산 처리 시스템에 속한 복수의 작업 노드에 연결된 네트워크 인터페이스를 포함할 수 있다. 이 때, 상기 컴퓨터 프로그램은, 상기 복수의 작업 노드에 의하여 분산 처리되는 파일을

제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되는 제2 영역으로 구분하는 오퍼레이션과, 상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 오퍼레이션과, 상기 제2 영역을 고정 사이즈의 블록들로 분할하는 오퍼레이션과, 제1 영역의 블록들 및 제2 영역의 블록들을 상기 복수의 작업 노드들에 상기 네트워크 인터페이스를 통하여 균등하게 나누어 송신하는 오퍼레이션을 포함할 수 있다.

- [20] 상기 기술적 과제를 해결하기 위한 본 발명의 또 다른 실시예에 따른 컴퓨터 프로그램이 제공된다. 상기 컴퓨터 프로그램은, 분산 처리 시스템에 속한 복수의 작업 노드에 의하여 분산 처리 되는 파일을 제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되는 제2 영역으로 구분하는 단계와, 상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계와, 상기 제2 영역을 고정 사이즈의 블록들로 분할하는 단계를 각각 수행한다. 상기 컴퓨터 프로그램은, 몇몇 실시예들에서, 제1 영역의 블록들 및 제2 영역의 블록들을 상기 복수의 작업 노드들에 상기 네트워크 인터페이스를 통하여 균등하게 나누어 송신하는 단계와, 상기 작업 노드들의 블록 처리 현황을 모니터링 하여, 각 작업 노드들의 미처리 블록 현황에 따라 상기 제2 영역의 블록을 대상으로 한 부하 리밸런싱 작업을 수행하는 단계를 더 수행할 수 있다.

발명의 효과

- [21] 상기와 같은 본 발명에 따르면, 작업 분산의 효과를 유지하면서도, 파일 사이즈가 증가할수록 블록의 개수가 증가하는 수준을 억제하여 파일 블록화를 효율적으로 수행할 수 있다.
- [22] 본 발명의 효과들은 이상에서 언급한 효과들로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해 될 수 있을 것이다.

도면의 간단한 설명

- [23] 도 1은 본 발명의 일 실시예에 따른 대용량 파일의 분산 처리 시스템 구성도이다.
- [24] 도 2는 본 발명의 다른 실시예에 따른 대용량 파일의 블록화 방법의 순서도이다.
- [25] 도 3은 본 발명의 몇몇 실시예들에서 고려되는 대용량 파일의 제1, 2 영역 구분 시, 제2 영역의 사이즈를 결정하는 방법을 설명하기 위한 개념도이다.
- [26] 도 4는 본 발명의 몇몇 실시예들에서 고려되는 대용량 파일의 제1, 2 영역을 설명하기 위한 개념도이다.
- [27] 도 5a는 도 2의 대용량 파일의 블록화 방법에 따라 파일의 제1 영역이 블록화 되는 것의 예시도이다.
- [28] 도 5b는 도 2의 대용량 파일의 블록화 방법에 따라 파일의 제2 영역이 블록화 되는 것의 예시도이다.

- [29] 도 6은 도 5a 및 도 5b에 예시된 블록화 이후, 각 블록이 복수의 작업 노드에 할당된 결과를 설명하기 위한 개념도이다.
- [30] 도 7은, 도 6과 비교되는, 종래기술에 따른 데이터 블록의 할당 결과를 나타내는 개념도이다.
- [31] 도 8은 도 2의 일부 동작을 보다 자세히 설명하기 위한 상세 순서도이다.
- [32] 도 9는 도 8의 일부 동작을 보다 자세히 설명하기 위한 상세 순서도이다.
- [33] 도 10은 본 발명의 또 다른 실시예에 따른 대용량 파일의 분산 처리 방법의 순서도이다.
- [34] 도 11은 본 발명의 또 다른 실시예에 따른 대용량 파일의 분산 처리 장치의 구성도이다.

발명의 실시를 위한 형태

- [35] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세히 설명한다. 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시 예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시 예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시 예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.
- [36] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다. 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다.

[37] 대용량 파일의 분산 처리 시스템

- [38] 이하, 도 1을 참조하여, 본 발명의 일 실시예에 따른 대용량 파일의 분산 처리 시스템의 구성 및 동작을 설명한다. 본 실시예에 따른 분산 처리 시스템은 분산 처리 관리 장치(100) 및 복수의 작업 노드(202)들로 구성된 작업 노드 클러스터(200)를 포함할 수 있다.
- [39] 분산 처리 관리 장치(100)는, 대용량 파일 생성 시스템(300)으로부터 제공된 대용량 파일을 블록화하여, 블록화의 결과 형성된 각각의 블록을 각 작업 노드(202)에 할당할 수 있다. 각각의 작업 노드(202)는 할당된 블록에 대하여 사전 정의된 처리 로직을 수행한다. 대용량 파일 생성 시스템(300)은 대용량 데이터를 수집하여 대용량 파일을 생성하는 다양한 기능의 시스템을 지칭한다.

예를 들어, 대용량 파일 생성 시스템(300)은 신용카드 거래 내역을 집계하는 시스템, 소셜 네트워크 시스템 상의 사용자 활동 내역을 집계하는 시스템, 각 단말 장치의 위치 정보를 수집하는 시스템 등 일 수 있다.

- [40] 도 1에는 분산 처리 관리 장치(100)와 대용량 파일 생성 시스템(300)이 서로 물리적으로 분리된 것으로 도시되어 있으나, 몇몇 실시예에서는, 분산 처리 관리 장치(100)가 대용량 파일 생성 시스템(300) 내부의 한 모듈로서 구성될 수도 있다.
- [41] 이미 언급한 바와 같이, 분산 처리 관리 장치(100)는 상기 대용량 파일을 블록화한다. 상기 대용량 파일은 제1 영역 및 제2 영역으로 나누어질 수 있다. 상기 제2 영역은, 각각의 작업 노드(202)에 의하여 처리될 때, 상기 제1 영역의 데이터보다 나중에 처리되는 영역이다. 분산 처리 관리 장치(100)는 상기 제1 영역과 상기 제2 영역을 서로 다른 방식으로 블록화한다. 분산 처리 관리 장치(100)는 상기 제1 영역을 다양한 사이즈의 블록들로 분할한다. 또한, 분산 처리 관리 장치(100)는 상기 제2 영역을 고정 사이즈의 블록들로 분할한다.
- [42] 상기 제1 영역에 형성된 대부분의 블록들은 상기 고정 사이즈의 블록들보다 더 큰 블록 사이즈를 갖는다. 따라서, 분산 처리 관리 장치(100)는 상기 대용량 파일 전체를 상기 고정 사이즈의 블록으로 블록화하는 것에 비하여, 더 작은 수의 블록을 형성한다.
- [43] 또한, 나중에 처리되는 영역인 상기 제2 영역은 고정 사이즈의 블록들로 분할되는데, 상기 고정 사이즈는 고정된 시스템 설정 또는 사용자 설정에 따라 결정된다. 상기 고정 사이즈는, 예를 들어 64MB(MEGA BYTE) 또는 128MB 등 부하 리밸런싱(load re-balancing)의 단위가 될 수 있도록, 기 지정된 데이터 사이즈를 초과하지 않는 범위에서 설정된다.
- [44] 요컨대, 분산 처리 관리 장치(100)는 부하 리밸런싱이 필요하지 않은 대용량 파일의 초반부 블록들은 큰 사이즈의 블록으로, 부하 리밸런싱이 필요할 수 있는 대용량 파일의 후반부 블록들은 작은 고정 사이즈의 블록으로 각각 블록화한다. 본 발명에서 제시되는 신규의 블록화 방법은, 종래기술에 따른 고정 사이즈 블록화에 비하여 더 작은 수의 블록으로 동일한 작업 분산의 효과를 누릴 수 있도록 한다. 분산 처리 관리 장치(100)가 수행하는 블록화 방법에 대하여는 몇몇 실시예들을 통하여 보다 자세히 설명될 것이다.
- [45] 분산 처리 관리 장치(100)는, HDFS에 따라 상기 대용량 파일을 블록화할 수 있다. 이 때, 각각의 블록은 작업 노드(202)에 분산하여 저장되고, 각 블록의 위치 등에 대한 메타 정보는 분산 처리 관리 장치(100)에 저장될 수 있다. 즉, 이 때에는 작업 노드(202)가 HDFS의 데이터 노드이고, 분산 처리 관리 장치(100)가 HDFS의 네임 노드인 것으로 이해될 수 있을 것이다.
- [46] 분산 처리 관리 장치(100)는 분산 처리에 있어서 태스크 스케줄러의 역할을 추가로 수행할 수 있다. 즉, 분산 처리 관리 장치(100)는 각 작업 노드(202)에 처리 및 저장 대상인 블록들을 제공함으로써 태스크를 할당하는 동작을 수행하고, 각

작업 노드(202)이 할당 받은 블록들에 대하여 처리 로직을 수행하는 상황을 모니터링 함으로써, 각 작업 노드(202)의 처리 진척 상황에 따라 부하 리밸런싱 작업을 수행할 수 있다.

- [47] 본 발명의 몇몇 실시예들에서, 상기 부하 리밸런싱 작업은 각각의 작업 노드(202)가 실질적으로 동일한 시간에 할당 받은 블록들에 대하여 처리를 완료하는 것을 목표로 한다. 실질적으로 동일한 시간에 처리를 완료하는 것은, 처리 완료 예상 시간 사이의 격차가, 기 지정된 시간을 초과하지 않는 것을 의미한다. 따라서, 상기 부하 리밸런싱 작업은, 많은 미처리 블록을 가지는 작업 노드(202)에 할당된 블록들을, 적은 미처리 블록을 가지는 작업 노드(202)로 재할당하는 작업을 포함한다. 분산 처리 관리 장치(100)가 수행하는 부하 리밸런싱 방법에 대하여도, 몇몇 실시예들을 통하여 보다 자세히 설명될 것이다.

[48] 대용량 파일의 블록화

- [49] 이하, 도 2를 참조하여, 본 발명의 다른 실시예에 따른 대용량 파일의 블록화 방법을 설명한다. 본 실시예에 따른 대용량 파일의 블록화 방법은 컴퓨팅 장치에 의하여 수행될 수 있다. 예를 들어, 본 실시예에 따른 대용량 파일의 블록화 방법은 도 1에 도시된 분산 처리 관리 장치(100)에 의하여 수행될 수 있다. 이하, 도 2와 관련된 설명에 있어서는, 이해의 편의를 돕기 위하여 동작의 주체가 생략될 수 있음을 유의한다.

- [50] 도 2에 도시된 바와 같이, 블록화 대상인 파일의 사이즈가 제2 영역의 사이즈보다 큰 경우에 한하여, 제1 영역 및 제2 영역에 대하여 서로 다른 방식의 블록화가 적용된다(S14, S15). 즉, 도 2에는 블록화 대상인 파일의 사이즈가 제2 영역의 사이즈보다 작거나 같은 경우, 전체 파일에 대하여 기 지정된 고정 사이즈의 블록들로 블록화가 수행(S12)되는 것으로 도시되어 있다. 본 발명의 블록화 방법은 도 2에 도시된 것으로 한정되지 않는다. 본 발명은, 도 2에 도시된 것과는 달리, 블록화 대상인 파일의 사이즈에 무관하게 항상 제1 영역 및 제2 영역에 대하여 서로 다른 방식의 블록화가 적용되는 실시도 포함하는 점을 유의한다.

- [51] 다시 도 2에 도시된 실시예를 설명하면, 먼저, 분할 대상인 파일의 사이즈가 제2 영역의 사이즈보다 큰지 판단한다(S10). 상기 제2 영역은, 이미 언급된 바와 같이, 상기 파일에서 나중에 처리 되는 영역이다. 제2 영역은 부하 분산의 효과를 달성하기 위하여 적절한 사이즈를 가져야 한다. 제2 영역의 사이즈를 구하는 방법에 대하여 도 3을 참조하여 설명한다.

- [52] 이미 언급된 바와 같이, 제2 영역에는 고정 사이즈의 블록(이하, ‘기본 블록’이라 함)이 형성된다. 상기 기본 블록의 사이즈가 64MB인 것으로 가정하자. 제2 영역에 형성된 기본 블록은 각 작업 노드(202)에 균등하게 나누어 할당될 것이다. 예를 들어, 제2 영역에 기본 블록이 100개 형성되고, 작업 노드가 20개라면, 각 작업 노드(202)는 기본 블록을 5개씩($100 / 20$) 할당 받을 것이다.

- [53] 각각의 작업 노드(202)는 복수개의 병렬 프로세스가 실행 되는 환경을 가질 수

있다. 상기 병렬 프로세스는, 멀티 태스킹(multi-tasking) 기술을 이용하여, 각 작업 노드(202)에서 동시에 병렬적으로 실행된다. 각각의 병렬 프로세스는, 적어도 하나의 기본 블록을 할당 받아야 한다. 따라서, 일 실시예에서, 상기 제2 영역의 사이즈는, 작업 노드의 개수와 데이터 분산 처리용 병렬 프로세스의 개수를 곱한 값을 이용하여 연산될 수 있다.

[54] 상기 기본 블록은, 부하 리밸런싱의 대상으로 사용될 수 있는 점에서, 일종의 버퍼 역할을 한다. 따라서, 각 작업 노드(202)가 기본 블록을 많이 가질수록, 상기 부하 리밸런싱을 이용하여 각 작업 노드의 블록 처리가 종료되는 시간을 동일하게 맞출 수 있는 가능성이 높아진다. 각 작업 노드(202)의 각각의 상기 병렬 프로세스에 할당되는 기본 블록의 개수를 작업 분산 효과 파라미터라고 지칭한다. 상기 작업 분산 효과 파라미터는 각각의 상기 작업 노드가 동시에 상기 파일 처리를 완료해야 할 필요성이 높을 수록 큰 값으로 설정되는 1이상의 자연수인 것으로 이해할 수 있다. 이 때, 상기 제2 영역의 사이즈는, 상기 고정 사이즈와, 상기 분산 처리 시스템에 속한 작업 노드의 개수와, 상기 데이터 분산 처리용 병렬 프로세스의 개수와, 작업 분산 효과 파라미터를 모두 곱한 값으로 자동 연산된다.

[55] 분산 처리 시스템의 관리자는, 각각의 상기 작업 노드가 동시에 상기 파일 처리를 완료해야 할 필요성이 높을 수록 상기 작업 분산 효과 파라미터를 높은 값으로 설정할 수 있다. 그러나, 상기 작업 분산 효과 파라미터가 커질수록 블록의 개수도 많아지기 때문에, 상기 작업 분산 효과 파라미터를 무작정 큰 값으로 설정하는 것은 피해야 한다. 이하, 분산 처리 시스템의 컴퓨팅 환경에 따라, 분산 처리 시스템이 상기 작업 분산 효과 파라미터를 자동으로 설정하는 실시예에 대하여 설명한다.

[56] 상기 작업 분산 효과 파라미터는, 각 작업 노드(202)간 성능 차이를 반영하여 자동으로 설정될 수 있다. 각 작업 노드(202)간 성능 차이에 대한 데이터는, 각 작업 노드(202)에 대하여 주기적으로 수행되거나 특정 이벤트 발생에 트리거 되어 수행된 연산 속도 테스트 수행 결과를 자동으로 수집하는 방식 등을 통하여 사전에 확보 될 수 있다. 상기 작업 분산 효과 파라미터는, 각 작업 노드(202)간 성능이 동일할수록 작게 설정되고, 각 작업 노드(202)간 성능 차이가 클수록 크게 설정될 수 있다.

[57] 상기 작업 분산 효과 파라미터는, 기 입력된 각 작업 노드(202)의 운영 비용을 반영하여 자동으로 설정될 수도 있다. 각 작업 노드(202)가 각 블록에 대한 처리를 실질적으로 동시에 마치도록 부하 리밸런싱을 수행하는 이유는, 일부 작업 노드(202)가 먼저 작업을 마치고 다른 작업 노드(202)들이 작업을 마칠 때까지 유휴 상태로 대기하는 것을 방지하기 위함이다. 따라서, 각 작업 노드(202)의 시간당 운영비용이 높을수록, 유휴 상태의 발생 확률을 최소화하기 위하여 상기 작업 분산 효과 파라미터를 높이는 것이 바람직하다. 반대로, 각 작업 노드(202)의 시간당 운영비용이 낮을수록, 유휴 상태의 발생 확률이 어느

정도 높아지더라도 블록 분할 개수를 낮추는 것이 오히려 상대적으로 이익일 수도 있으므로(블록 분할에 관한 메타 정보의 사이즈를 낮추는 것이 더 이익), 상기 작업 분산 효과 파라미터를 낮추는 것이 바람직하다.

- [58] 상기 작업 분산 효과 파라미터는, 각 작업 노드를 통하여 분산 처리되는 파일에 적용되는 로직의 프로그래밍 복잡도(programming complexity)를 이용하여 자동 설정될 수도 있다. 상기 프로그래밍 복잡도는 소프트웨어 메트릭(software metric) 중 하나로, 소프트웨어 내부의 복수 엔터티 간의 인터랙션이 어느 정도 존재할 것인지를 가리킨다. 각각의 블록에 적용되는 로직이 동일하지만, 입력 되는 데이터가 서로 다르기 때문에, 상기 로직의 상기 프로그래밍 복잡도가 높을수록 처리에 소요되는 연산 부하가 서로 다를 가능성이 높다. 따라서, 상기 로직의 프로그래밍 복잡도가 높을수록 상기 작업 분산 효과 파라미터를 높여서, 각 작업 노드의 블록 처리 속도가 달라지는 것에 대비하여 부하 리밸런싱 역량을 좀더 증강하는 것이 바람직하다.
- [59] 상기 프로그래밍 복잡도는, 예를 들어, Thomas J. McCabe, Sr.에 의하여 1976년 정의된 Cyclomatic Complexity 값(이하, "McCabe의 Cyclomatic Complexity"라 함)을 가리킬 수 있다. 상기 McCabe의 Cyclomatic Complexity는 프로그램 소스 코드의 독립적인 실행 경로의 개수를 가리키는 값이다. 상기 McCabe의 Cyclomatic Complexity를 구하는 방법에 대하여는, 제1 문헌: McCabe (December 1976). "A Complexity Measure". IEEE Transactions on Software Engineering: 308-320과, 제2 문헌: Arthur H. Watson and Thomas J. McCabe (1996). "Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric" (PDF). NIST Special Publication 500-235 등의 널리 알려진 문헌들을 참조할 수 있다.
- [60] 상기 로직에 대한 McCabe의 Cyclomatic Complexity를 구하기 위하여, 제3 문헌: A small C/C++ source code analyzer using the cyclometric complexity metric(<https://sites.google.com/site/maltapplication/>) 등의 널리 알려진 소스 코드 분석기를 활용할 수 있다.
- [61] 요컨대, 상기 작업 분산 효과 파라미터는, 작업 노드 간 성능 차이가 클수록, 작업 노드의 시간당 운영비용이 높을수록, 처리 로직의 McCabe의 Cyclomatic Complexity가 클수록, 더 큰 값으로 자동 설정될 수 있다.
- [62] 도 3에 도시된 상황의 경우, 각 작업 노드(202)는 3개의 병렬 프로세스를 동시에 실행하고, 작업 분산 효과 파라미터는 4로 설정되어 있다. 또한, 기본 블록의 사이즈는 64MB이다. 따라서, 작업 노드의 개수를 n 이라 하면, 제2 영역의 사이즈는 $(768 * n)$ MB로 자동으로 연산된다. 예를 들어, 작업 노드의 개수가 100개라면, 제2 영역의 사이즈는 76800MB=75GB가 된다.
- [63] 도 4를 참조하면, 제2 영역(404)의 사이즈가 75GB이고, 제2 영역은 파일의 뒤쪽 75GB 영역에 위치한다. 본 명세서에서, 파일(400)은 앞쪽에서 뒤쪽의 순서로 처리(processing)되며, 가장 앞쪽은 파일의 오프셋이 0인 데이터이고, 가장 뒤쪽은 파일의 오프셋이 (FILE_SIZE - 1)인 데이터가 될 것이다. 파일의 구분

방법은, 파일을 기준 오프셋(파일 사이즈 - 제2 영역 사이즈)을 기준으로 2개의 연속된 영역으로 나누되, 먼저 처리 되는 앞쪽 영역을 제1 영역(402)으로 결정하고, 나중에 처리 되는 뒤쪽 영역을 제2 영역(404)으로 결정한다.

- [64] 다시 도 2로 돌아와서 설명하면, 파일이 제2 영역 사이즈(제2 영역 사이즈 구하는 방법은 도 3 참조)보다 더 큰 사이즈를 가지면(S10), 상기 파일을 제1 영역 및 제2 영역으로 구분하고(구분 방법은 도 4 참조)(S13), 제1 영역은 다양한 사이즈의 블록들로 분할한다(S14).
- [65] 일 실시예에서, 제1 영역은 각 사이즈의 블록을 M개씩 포함할 수 있다(M은 작업 노드의 개수). 이는, 각 작업 노드가 각 사이즈의 블록을 1개씩 할당 받도록 하기 위함이다. 도 5a는, 작업 노드가 2개인 경우, 제1 영역에 256MB 사이즈의 블록 2개(420a, 420b)와 128MB 사이즈의 블록 2개(421a, 421b)가 형성되는 것을 도시한다.
- [66] 제2 영역은 고정 사이즈의 블록들, 즉 기본 블록들로 분할한다(S15). 도 5b는 작업 노드가 2개, 각 작업 노드 당 병렬 프로세스 개수가 1개, 상기 작업 분산 효과 파라미터가 3으로 설정된 경우, 제2 영역에 6개($2 * 1 * 3$)의 기본 블록이 형성되는 것을 도시한다.
- [67] 도 6은, 도 5a 및 도 5b에 도시된 것과 같이 파일이 블록화 된 경우, 2개의 작업 노드 각각이 블록을 어떻게 할당 받는지 도시한다. 작업 노드#1은 256MB의 블록(420a), 128MB의 블록(421a) 및 기본 블록(440a, 440c, 440e)의 순서로 각 블록을 처리한다. 작업 노드#2는 256MB의 블록(420b), 128MB의 블록(421b) 및 기본 블록(440b, 440d, 440f)의 순서로 각 블록을 처리한다. 제1 영역(402)의 다양한 사이즈의 블록들(420a, 420b, 421a, 421b)이 작업 노드#1, 작업 노드#2에 균등하게 할당된 것을 확인할 수 있다. 또한, 제2 영역(404)의 6개의 기본 블록들(440a, 440b, 440c, 440d, 440e, 440f)도 작업 노드#1, 작업 노드#2에 3개씩 균등하게 할당된 것을 확인할 수 있다.
- [68] 도 7은, 종래 기술에 따라, 전체 파일을 모두 기본 블록들로만 블록화 한 경우, 작업 노드#1, 작업 노드#2각 상기 기본 블록을 같은 개수 할당 받은 것을 도시한다. 본 발명에 따르면, 파일이 총 10개의 블록으로 블록화 되는 반면, 종래 기술에서는 파일이 총 18개의 블록으로 블록화 된다. 본 발명에서는 각각의 작업 노드가 큰 사이즈를 가지는 블록을 대상으로 초기 작업을 수행하는 반면, 후기 작업에서는 작은 사이즈를 가지는 블록을 대상으로 하므로, 필요 시에 부하 리밸런싱에 큰 부담이 없다. 따라서, 본 발명에서는 더 작은 개수의 블록으로도 부하 분산의 효과를 달성할 수 있는 효과가 있다. 또한, 본 발명에서는 분산 처리 시스템의 작업 환경에 대응 되는 사이즈를 가진 제2 영역이 대용량 파일의 뒤쪽에 배치됨으로써, 상기 분산 처리 시스템에서 부하 리밸런싱 작업이 수행되기에 적합한 블록 배치를 제공한다.
- [69] 이하, 도 8 내지 도 9를 참조하여, 제1 영역을 다양한 사이즈의 블록들로 분할하는 방법에 대하여 보다 상세하게 설명한다.

- [70] 도 8은 도 2에 도시된 제1 영역 블록화 단계(S14)의 상세 순서도이다.
- [71] 먼저, 제1 영역의 블록 미형성 영역 사이즈(F)가 기본 블록의 고정 사이즈(b)와 작업 노드의 개수(M)을 곱한 값보다 큰지 여부가 판단된다(S140). 이 판단 동작은, 블록을 형성해 줄 수 있는 영역이 각 작업 노드에 대하여 b를 초과하는 사이즈의 블록을 형성해 줄 수 있을 정도의 크기를 가지는 지 여부를 결정하기 위한 것으로 이해 될 수 있다. 예를 들어, $b=64\text{MB}$ 이고, $M=10$ 인 경우, 제1 영역의 블록 미형성 영역이 640MB 미만이면(500MB 라고 가정하자), 상기 블록 미형성 영역의 사이즈(F)를 10으로 나눈 크기(50MB)를 블록 사이즈로 결정한다(S146). 다음으로, 블록 미형성 영역의 앞에서부터 50MB 의 블록을 10개 형성해 주는 것으로 제1 영역에 대한 블록 형성을 마무리 한다(S148).
- [72] 제1 영역의 블록 미형성 영역 사이즈(F)가 기본 블록의 고정 사이즈(b)와 작업 노드의 개수(M)을 곱한 값보다 큰 경우(S140), 형성해 줄 블록의 사이즈는 $F \cdot i / M$ 으로 결정된다(S142). i 는 0과 1사이의 값을 갖는 가변 분할 파라미터이다. 그리고, i 는 한번 정해지면 그 값이 변하지 않고 일정하게 유지된다. 블록 사이즈가 결정되면, 그 사이즈의 블록을 블록 미형성 영역의 앞에서부터 M개 순차로 형성한다(S144).
- [73] 예를 들어, 가변 분할 파라미터가 0.6이라면, 제1 영역의 블록 미형성 영역 중 60%의 영역을 작업 노드의 개수로 나누면, 형성해 줄 블록의 사이즈가 결정된다. 다음 회차에 형성해 줄 블록의 사이즈는, 남은 블록 미형성 영역의 60% 영역을 작업 노드의 개수로 나누어 결정된다. 따라서, 시간이 지날수록 형성되는 블록의 사이즈는 작아지게 된다. 형성되는 블록의 사이즈가 상기 기본 블록의 고정 사이즈보다 작아지는 경우, 남은 영역을 작업 노드의 개수로 나누어 동일 크기의 블록을 마지막으로 형성해 준 후, 제1 영역에 대한 블록화를 마무리한다.
- [74] 또한, 블록의 형성은, 블록 미형성 영역의 앞쪽(즉, 파일의 오프셋이 작은 쪽)에서부터 수행된다.
- [75] 따라서, 제1 영역에 대한 블록화는, 제1 내지 제n 사이즈의 블록(n 은 2이상의 자연수)이 파일의 앞에서부터 순차로 형성되는 것을 의미한다. 파일의 앞, 뒤는 파일에 포함된 각 데이터의 처리 순서를 가리키는 개념이다. 즉, 파일의 제1 오프셋 영역이 제2 오프셋 영역보다 앞에 있다는 것은, 제1 오프셋 영역이 제2 오프셋 영역보다 더 먼저 처리되는 것을 의미한다.
- [76] 요컨대, 상기 제1 영역의 블록화는, 상기 파일의 제일 앞에서부터 제1 사이즈의 블록을 M개씩 형성하는 단계(M은 상기 작업 노드의 개수)와, 블록이 형성되지 않은 영역에 상기 제1 사이즈보다 작은 제2 사이즈의 블록을 M개씩 형성하는 단계를 포함한다. 물론, 파일의 사이즈가 충분히 큰 경우, 아직 블록이 형성되지 않은 영역에 상기 제2 사이즈보다 작은 제3 사이즈의 블록을 M개씩 형성하는 단계, 아직 블록이 형성되지 않은 영역에 상기 제3 사이즈보다 작은 제4 사이즈의 블록을 M개씩 형성하는 단계 등이 추가로 수행 될 수 있을 것이다.
- [77] 본 발명의 블록 개수 감소 효과를 달성하기 위하여, 적어도 상기 제1 사이즈는

상기 기본 블록의 고정 사이즈(64MB으로 수 차례 예시됨)보다 큰 것이 바람직하다. 따라서, 상기 가변 분할 파라미터(i)의 값은 0과 1사이의 실수 값으로 설정되며, 제1 사이즈(즉, 제1 영역 내에 최초로 형성되는 블록의 사이즈)는 상기 기본 블록의 고정 사이즈보다 큰 값을 가질 수 있도록, 가변 분할 파라미터(i) 값의 최소 값을 제약하는 것이 바람직하다. 예를 들어, 제1 영역의 사이즈가 6400MB이고, $M=10$, 기본 블록의 고정 사이즈=64MB인 경우, $6400 * 0.1 / 10 = 64MB$ 이므로, 상기 가변 분할 파라미터 값은 적어도 0.1 이상의 값이 되어야 한다. 즉, 상기 가변 분할 파라미터(i) 값의 최소 값은, 제1 영역의 사이즈, 작업 노드의 개수, 기본 블록의 고정 사이즈를 이용하여 자동으로 연산될 수 있다.

[78] 상기 가변 분할 파라미터의 값이 1에 가까워질수록, 제1 영역에 형성되는 블록의 개수는 감소할 것이고, 반대로 가변 분할 파라미터의 값이 0에 가까워질수록, 제1 영역에 형성되는 블록의 개수는 증가할 것이다. 본 발명의 블록 감소 효과를 극대화하기 위하여는, 상기 가변 분할 파라미터의 값을 1으로 설정하여 제1 영역 전체를 하나의 블록으로 형성하는 것도 고려할 수 있을 것이다. 다만, 이렇게 제1 영역 전체를 하나의 블록으로 형성하면, 특정 작업 노드에서 비정상 상황이 발생되어 작업 속도가 극단적으로 느려진 경우에 적절히 대응할 수 없다. 이미 설명된 몇몇 실시예에서는 제2 영역의 블록들만 부하 리밸런싱의 대상이 될 수 있으나, 본 발명의 또 다른 몇몇 실시예에서는 제2 영역의 블록들뿐만 아니라 제1 영역의 블록들 중 기 지정된 사이즈 이하의 사이즈를 갖는 블록들은 부하 리밸런싱의 대상으로 포함시킬 수 있다. 이러한 실시예를 고려하면, 제1 영역에도 부하 리밸런싱이 진행 되어야 하는 상황을 고려하여, 한 작은 사이즈의 블록들을 형성시킬 필요가 있다. 따라서, 상기 가변 분할 파라미터 값은, 1 미만의 값으로 설정하는 것이 바람직하다.

[79] 상기 가변 분할 파라미터 값(i)은, 관리자에 의하여 수동으로 설정 될 수도 있으나, 시스템에 의하여 최적화된 값이 자동 설정될 수도 있다. 이하, 상기 가변 분할 파라미터 값(i)이 자동 설정되는 실시예에 대하여, 도 9를 참조하여 자세히 설명한다.

[80] 각 작업 노드의 성능 균일도를 이용하여, 상기 가변 분할 파라미터 값이 자동 설정 될 수 있다. 파일을 분산 처리하는 각각의 작업 노드가 동일한 성능을 가질수록 각 블록에 대한 동일한 처리 속도를 기대할 수 있기 때문에, 제1 영역에 형성되어야 하는 블록의 개수를 줄일 수 있다. 따라서, 각 작업 노드의 성능이 균일할수록 상기 가변 분할 파라미터는 높은 값으로 자동 설정될 수 있다.

[81] 각 작업 노드의 성능 균일도는 하드웨어 스펙(spec.) 정보로부터 측정될 수 있다. 각 작업 노드에 상기 블록의 처리를 위한 프로세스 이외의 다른 프로세스가 많이 실행되지 않는 이상적인 환경인 경우, 상기 하드웨어 스펙 정보는 상기 블록의 처리 속도를 예측하기 위한 신뢰할 수 있는 정보이기 때문이다.

[82] 도 9를 참조하여 하드웨어 스펙 정보로부터 각 작업 노드의 성능 균일도를

측정하는 방법을 설명한다. 먼저, 스펙(spec) 정보를 분산 처리 시스템의 자원 관리 정보에서 조회한다(S1420). 다음으로, 각 작업 노드의 스펙을 스코어링(scoring) 한다(S1422). 상기 스코어링 과정에서 복수의 기준이 적용될 수 있다. 예를 들어, 상기 파일에 적용되어야 하는 로직이 많은 연산을 요하는 로직인 경우, CPU 성능에 높은 가중치가 부여된 기준이 스코어링 과정에 적용되고, 상기 파일에 적용되어야 하는 로직이 많은 메모리를 요하는 로직인 경우, 메모리 용량에 높은 가중치가 부여된 기준이 스코어링 과정에 적용되고, 상기 파일에 적용되어야 하는 로직이 많은 I/O(스토리지에 대한 입출력 작업)를 요하는 것인 경우, 스토리지 용량 및 스토리지 속도에 높은 가중치가 부여된 기준이 스코어링 과정에 적용될 수 있다. 다음으로, 스코어링(S1422)의 결과 연산된 각 작업 노드의 스펙 스코어에 대하여 분산 값(Var_node)(또는, 표준 편차 값)을 연산함으로써, 상기 성능 균일도를 측정할 수 있다(S1424). 상기 성능 균일도는 상기 분산 값(또는, 표준 편차 값)이 낮을수록 높은 값을 가진다.

[83] 각 작업 노드에 상기 블록의 처리를 위한 프로세스 이외의 다른 다양한 프로세스가 실행되고 있는 상황이라면, 상기 하드웨어 스펙 정보는 각 작업 노드의 작업 환경을 정확하게 반영하지 못한다. 이때에는 실제 연산 속도에 대한 벤치마킹 테스트 결과를 활용하여 상기 블록의 처리 속도를 예측할 수 있다. 이때에는, 각 작업 노드에서 주기적으로, 또는 비주기적으로 실시되는 연산 속도 테스트 수행 결과를 자동으로 수집한 후, 상기 테스트 수행 결과에 따라 매겨진 스코어의 분산 값(또는 표준 편차 값)을 연산함으로써, 상기 성능 균일도를 측정할 수도 있을 것이다.

[84] 각 작업 노드를 통하여 분산 처리되는 파일에 적용되는 로직의 프로그래밍 복잡도를 이용하여 상기 가변 분할 파라미터 값을 자동 설정할 수도 있다. 이미 언급한 바와 같이, 각각의 블록에 적용되는 로직이 동일하지만, 입력 되는 데이터가 서로 다르기 때문에, 상기 로직의 상기 프로그래밍 복잡도가 높을수록 처리에 소요되는 연산 부하가 서로 다를 가능성이 높다. 따라서, 상기 로직의 프로그래밍 복잡도가 높을수록 상기 가변 분할 파라미터를 낮춰서, 각 작업 노드의 블록 처리 속도가 달라지는 것에 대비하여 부하 리밸런싱 역량을 좀더 증강하는 것이 바람직하다. 상기 프로그래밍 복잡도는 McCabe의 Cyclomatic Complexity를 가리킬 수 있다.

[85] 일 실시예에서, 상기 프로그래밍 복잡도가 상기 제2 영역의 사이즈를 구할 때 작업 분산 효과 파라미터의 형태로 이미 반영되었다면, 상기 제1 영역의 블록화에 활용되는 상기 가변 분할 파라미터 값의 자동 설정에는 상기 프로그래밍 복잡도를 중복하여 적용하지 않을 수 있다.

[86] 반면에, 다른 실시예에서는, 상기 프로그래밍 복잡도가 상기 제2 영역의 사이즈를 구할 때 작업 분산 효과 파라미터의 형태로 이미 반영되었더라도, 상기 제1 영역의 블록화에 활용되는 상기 가변 분할 파라미터 값의 자동 설정에 상기 프로그래밍 복잡도를 중복하여 적용할 수 있다.

- [87] 또 다른 실시예에서, 상기 프로그래밍 복잡도가 기 설정된 한계치 이하의 값을 가지는 경우에는 상기 프로그래밍 복잡도가 상기 제2 영역의 사이즈를 구할 때 작업 분산 효과 파라미터의 형태로 이미 반영되었다면, 상기 제1 영역의 블록화에 활용되는 상기 가변 분할 파라미터 값의 자동 설정에는 상기 프로그래밍 복잡도를 중복하여 적용하지 않고, 상기 프로그래밍 복잡도가 상기 한계치 이상의 값을 가지는 경우에는 상기 제2 영역의 사이즈를 구할 때 작업 분산 효과 파라미터의 형태로 이미 반영되었더라도, 상기 제1 영역의 블록화에 활용되는 상기 가변 분할 파라미터 값의 자동 설정에 상기 프로그래밍 복잡도를 중복하여 적용할 수 있다.
- [88] 상기 가변 분할 파라미터 값(i)을 설정하는데, 상기 프로그래밍 복잡도 및 상기 성능 균일도가 모두 반영될 수 있는 것은 물론이다(S1428). 이 때, 상기 가변 분할 파라미터 값(i)은 각 작업 노드의 스펙 스코어의 분산 값(Var_node) 및 프로그래밍 복잡도(PC)에 반비례할 수 있다. 도 9에 도시된 수식에서 α 는 테스트 과정에서 적응적으로 설정될 수 있는 계수이다.
- [89] 분산 처리에 대한 관리(태스크 스케줄링)
- [90] 이하, 도 10을 참조하여, 본 발명의 또 다른 실시예에 따른 분산 처리 관리 방법을 설명한다. 본 실시예에 따른 분산 처리 관리 방법은 컴퓨팅 장치에 의하여 수행될 수 있다. 예를 들어, 본 실시예에 따른 분산 처리 관리 방법은 도 1에 도시된 분산 처리 관리 장치(100)에 의하여 수행될 수 있다. 이하, 본 실시예와 관련된 설명에 있어서는, 이해의 편의를 돕기 위하여 동작의 주체가 생략될 수 있음을 유의한다.
- [91] 먼저, 도 2에 도시된 파일 블록화 동작이 수행된다(S10, S12, S13, S14, S15). 이 때, 도 3 내지 도 9를 참조하여 설명한 본 발명의 파일 블록화 관련 실시예들이 적용될 수 있음은 물론이다.
- [92] 파일 블록화가 완료된 후, 각각의 블록이 각 작업 노드에 균등하게 할당된다(S16). “균등하게 할당”되는 것은, 특정 사이즈의 블록이 특정 작업 노드에 편중되지 않도록 할당하는 것을 의미한다. 예를 들어, 제1 작업 노드가 제1 사이즈를 가진 블록을 M개 가졌다면, 다른 작업 노드들도 상기 제1 사이즈를 가진 블록을 M개 가지도록 블록을 할당하는 것을 의미한다. 물론, 블록의 개수가 모든 작업 노드에 할당하기에 모자라는 경우, 일부 작업 노드는 다른 작업 노드에 할당된 블록을 할당 받지 못하는 경우가 생길 수도 있다. “균등하게 할당”되는 것의 의미는 도 6과 관련된 설명을 참조한다.
- [93] 블록 할당이 마무리된 후, 각 작업 노드의 블록 처리 현황을 모니터링한다(S17). 상기 모니터링은, 예를 들어, 각 작업 노드가 기록한 로그(LOG)를 확인하거나, 각 작업 노드가 블록 처리 완료 시 보고 메시지를 송신하는 등의 방식에 의하여 수행될 수 있을 것이다. 블록 처리(block processing)는, 각 작업 노드가 할당된 블록에 포함된 데이터에, 블록화 된 파일의 처리 로직을 적용하는 것을 가리킨다.

- [94] 상기 모니터링의 결과, 부하 리밸런싱이 필요한 상황으로 판단된 경우, 분산 처리 관리 장치(100)는 제2 영역의 블록을 대상으로 부하 리밸런싱을 수행한다(S18). 예를 들어, 제1 작업 노드는 상기 제1 작업 노드에 할당된 8개의 제2 영역 블록 중 1개의 미처리 블록이 존재하나, 제2 작업 노드는 상기 제2 작업 노드에 할당된 8개의 제2 영역 블록 중 7개의 미처리 블록이 존재하는 경우, 분산 처리 관리 장치(100)는 부하 리밸런싱이 필요한 상황으로 판단하고, 제2 작업 노드에 할당된 제2 영역 블록 중 적어도 일부를 상기 제1 작업 노드로 재할당할 수 있다.
- [95] 일 실시예에서, 분산 처리 관리 장치(100)는 제2 영역의 블록 중 미처리 블록의 상하위 편차가 기 지정된 기준을 초과할 정도로 심각해진 경우(예를 들어, 최상위와 최하위의 미처리 블록 보유 차이가, 각 작업 노드에 할당된 제2 영역 블록의 개수의 50%를 초과한 경우), 부하 리밸런싱을 수행할 수 있다.
- [96] 분산 처리 관리 장치(100)는 상기 부하 리밸런싱을 수행하는 것에 의하여, 각 작업 노드가 실질적으로 동일한 시간에 할당된 블록을 모두 처리할 수 있도록 관리한다(S19).
- [97] 지금까지 도 10 및 관련된 설명에서, 제2 영역의 블록 만이 부하 리밸런싱의 대상이 되는 것으로 설명하였으나, 몇몇 실시예들에서는, 제2 영역의 블록뿐만 아니라, 제1 영역의 블록 중 일부 블록에 대하여도 부하 리밸런싱이 수행될 수 있는 점에 유의한다. 상기 제1 영역의 블록 중, 부하 리밸런싱이 수행되는 블록은, 기 지정된 사이즈 보다 작은 사이즈를 가지는 블록이다.
- [98] 지금까지 도 1 내지 도 10을 참조하여 설명된 본 발명의 실시예에 따른 방법들은 컴퓨터가 읽을 수 있는 코드로 구현된 컴퓨터프로그램의 실행에 의하여 수행될 수 있다. 상기 컴퓨터프로그램은 인터넷 등의 네트워크를 통하여 제1 컴퓨팅 장치로부터 제2 컴퓨팅 장치에 전송되어 상기 제2 컴퓨팅 장치에 설치될 수 있고, 이로써 상기 제2 컴퓨팅 장치에서 사용될 수 있다. 상기 제1 컴퓨팅 장치 및 상기 제2 컴퓨팅 장치는, 서버 장치, 클라우드 서비스를 위한 서버 풀에 속한 물리 서버, 데스크탑 피씨와 같은 고정식 컴퓨팅 장치를 모두 포함한다.
- [99] 상기 컴퓨터프로그램은 DVD-ROM, 플래시 메모리 장치 등의 기록매체에 저장된 것일 수도 있다.
- [100] 이하, 도 11을 참조하여, 본 발명의 또 다른 실시예에 따른 대용량 파일의 분산 처리 관리 장치의 구성 및 동작을 설명한다.
- [101] 도 11에 도시된 바와 같이, 본 실시예에 따른 대용량 파일의 분산 처리 관리 장치(100)는 하나 이상의 프로세서(106), 분산 처리 시스템에 속한 복수의 작업 노드를 포함하는 작업 노드 클러스터(200)에 연결된 네트워크 인터페이스(102), 스토리지(104), 및 프로세서(106)에 의하여 수행되는 컴퓨터 프로그램이 로드(load)된 메모리(RAM)(108)를 포함할 수 있다. 프로세서(106), 네트워크 인터페이스(102), 스토리지(104) 및 메모리(108)는 시스템 버스(110)를 통하여

데이터를 송수신한다.

- [102] 스토리지(104)는, 네트워크 인터페이스(102)를 통하여 대용량 데이터 생성 시스템으로부터 수신한 대용량 파일(400)을 일시적으로 저장할 수 있다.
- [103] 상기 컴퓨터 프로그램은 파일 블록화 소프트웨어(180)를 포함한다.
- [104] 파일 블록화 소프트웨어(180)는, 대용량 파일(400)을 제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되는 제2 영역으로 구분하는 오퍼레이션(operation), 상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 오퍼레이션, 상기 제2 영역을 고정 사이즈의 블록들로 분할하는 오퍼레이션을 포함한다. 파일 블록화 방법은, 이미 도 1 내지 도 9를 참조하여 자세히 설명한 바 있다. 파일 블록화 소프트웨어(180)는 대용량 파일(400)의 블록화 결과를, 메모리(108) 상에 블록 분할 정보(184)로서 기록한다. 블록 분할 정보(184)는, 각 블록의 오프셋 레인지(offset range)를 포함할 수 있다.
- [105] 분산 처리 관리 소프트웨어(182)는 태스크 스케줄링을 담당하는 오퍼레이션들을 포함한다. 분산 처리 관리 소프트웨어(182)는, 블록 분할 정보(184)를 참조하여, 제1 영역의 블록들 및 제2 영역의 블록들을 상기 복수의 작업 노드들에 네트워크 인터페이스(102)를 통하여 균등하게 나누어 송신하는 오퍼레이션과 상기 작업 노드들의 블록 처리 현황을 모니터링 하여, 각각의 상기 작업 노드들의 미처리 블록 현황에 따라 상기 제2 영역의 블록을 대상으로 한 부하 리밸런싱 작업을 수행하는 오퍼레이션을 포함할 수 있다.
- [106] 본 실시예에 따른 대용량 파일의 분산 처리 관리 장치(100)는 부하 리밸런싱이 필요하지 않은 대용량 파일의 초반부 블록들은 큰 사이즈의 블록으로, 부하 리밸런싱이 필요할 수 있는 대용량 파일의 후반부 블록들은 작은 고정 사이즈의 블록으로 각각 블록화한다. 따라서, 분산 처리 관리 장치(100)는 종래기술에 따른 고정 사이즈 블록화에 비하여 더 작은 수의 블록으로 동일한 작업 분산의 효과를 누릴 수 있도록 한다.
- [107] 이상 첨부된 도면을 참조하여 본 발명의 실시예들을 설명하였지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로 이해해야만 한다.

청구범위

- [청구항 1] 분산 처리 시스템에 속한 복수의 작업 노드에 의하여 분산 처리되는 파일에 대한 블록화 방법에 있어서,
상기 파일을 제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되는 제2 영역으로 구분하는 단계;
상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계; 및
상기 제2 영역을 고정 사이즈의 블록들로 분할하는 단계를 포함하는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 2] 제1 항에 있어서,
상기 제1 영역은 각 사이즈의 블록을 M개씩 포함하는(M은 상기 작업 노드의 개수),
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 3] 제2 항에 있어서,
제1 영역을 다양한 사이즈의 블록들로 분할하는 단계는,
상기 파일의 제일 앞에서부터, 제1 사이즈의 블록을 M개씩 형성하는 단계(M은 상기 작업 노드의 개수); 및
블록이 형성되지 않은 영역에, 상기 제1 사이즈보다 작은 제2 사이즈의 블록을 M개씩 형성하는 단계를 포함하되,
상기 파일은, 파일의 앞에서 뒤쪽 방향으로 처리되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 4] 제3 항에 있어서,
상기 제1 사이즈는 상기 고정 사이즈 보다 큰,
파일 분할 방법.
- [청구항 5] 제3 항에 있어서,
상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계는,
블록이 형성되지 않은 영역에, 직전 형성 블록의 사이즈보다 작은 사이즈의 블록을 M개씩 형성하는 단계를, 상기 작은 사이즈가 상기 고정 사이즈보다 큰 것을 조건으로 반복하는 단계를 더 포함하는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 6] 제3 항에 있어서,
상기 제2 사이즈는, $(F * i / M)$ 의 사이즈를 가리키되,
 F = 블록이 형성되지 않은 영역의 사이즈,
 i = 가변 분할 파라미터, $(0 < i < 1)$
 M = 상기 작업 노드의 개수인,
분산 처리를 위한 대용량 파일의 블록화 방법.

- [청구항 7] 제6 항에 있어서,
i는 각 작업 노드의 성능이 동일할 수록 높은 값으로 설정되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 8] 제6 항에 있어서,
i는 상기 파일을 처리하는 로직의 복잡도가 높을 수록 낮은 값으로 설정되고,
상기 복잡도는, 프로그래밍 복잡도(programming complexity)에 의하여 결정되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 9] 제1 항에 있어서,
상기 제2 영역의 사이즈는, 상기 분산 처리 시스템에 속한 작업 노드의 개수를 이용하여 결정되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 10] 제9 항에 있어서,
상기 제2 영역의 사이즈는, 각각의 상기 작업 노드에서 실행되는 데이터 분산 처리용 병렬 프로세스의 개수를 더 이용하여 결정되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 11] 제10 항에 있어서,
상기 제2 영역의 사이즈는, 상기 고정 사이즈와, 상기 분산 처리 시스템에 속한 작업 노드의 개수와, 상기 데이터 분산 처리용 병렬 프로세스의 개수와, 작업 분산 효과 파라미터를 모두 곱한 값이고,
상기 작업 분산 효과 파라미터는, 각각의 상기 작업 노드가 동시에 상기 파일 처리를 완료해야 할 필요성이 높을 수록 큰 값으로 설정되는 1이상의 자연수인,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 12] 제1 항에 있어서,
상기 제2 영역의 사이즈는, 각 작업 노드의 성능이 동일할 수록 작게 설정되는 작업 분산 효과 파라미터를 반영하여 결정되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 13] 제1 항에 있어서,
상기 제2 영역의 사이즈는, 각 작업 노드의 운영 비용이 낮을수록 작게 설정되는 작업 분산 효과 파라미터를 반영하여 결정되는,
분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 14] 제1 항에 있어서,
상기 제2 영역의 사이즈는, 상기 파일의 처리 로직의 프로그래밍 복잡도가 높을수록 크게 설정되는 작업 분산 효과 파라미터를 반영하여 결정되되,

- 상기 프로그래밍 복잡도는, McCabe의 Cyclomatic Complexity 값을 가리키는,
 분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 15] 제9 내지 제14 항 중 어느 한 항에 있어서,
 상기 파일의 사이즈가 상기 결정된 제2 영역의 사이즈 보다 큰 경우에 한하여, 상기 구분하는 단계, 상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계, 및 상기 제2 영역을 동일 사이즈의 블록들로 분할하는 단계를 수행하는 단계를 더 포함하는,
 분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 16] 제1 항에 있어서,
 상기 제1 영역의 블록은 최초 할당 받은 작업 노드에 의하여 처리되어야 하는 블록이고, 상기 제2 영역의 블록은 최초 할당 받은 작업 노드가 다른 작업 노드에게 이관할 수 있는 블록인,
 분산 처리를 위한 대용량 파일의 블록화 방법.
- [청구항 17] 다양한 사이즈의 블록들로 분할된 제1 영역 및 상기 제1 영역의 데이터보다 나중에 처리되고 고정 사이즈의 블록들로 분할된 제2 영역을 가지는 파일을 제공받는 단계;
 제1 영역의 블록들 및 제2 영역의 블록들을 분산 처리 시스템에 속한 복수의 작업 노드들에 균등하게 나누어 할당하는 단계; 및
 상기 작업 노드들의 상기 블록들에 대한 처리 현황을 모니터링 하여, 각 작업 노드들의 미처리 블록 현황에 따라 부하 리밸런싱 작업을 수행하는 단계를 포함하는,
 대용량 파일의 분산 처리 방법.
- [청구항 18] 제17 항에 있어서,
 상기 제1 영역의 블록들 중 적어도 일부 블록들의 사이즈는 상기 고정 사이즈보다 큰,
 대용량 파일의 분산 처리 방법.
- [청구항 19] 제17 항에 있어서,
 상기 부하 리밸런싱 작업을 수행하는 단계는,
 상기 제2 영역의 블록 만을 상기 부하 리밸런싱 대상으로 하는 단계를 포함하는,
 대용량 파일의 분산 처리 방법.
- [청구항 20] 제17 항에 있어서,
 상기 부하 리밸런싱 작업을 수행하는 단계는,
 상기 제2 영역의 블록 및 상기 제1 영역의 블록들 중, 기 지정된 사이즈 이하의 사이즈를 갖는 블록들 만을 상기 부하 리밸런싱 대상으로 하는 단계를 포함하는,

[청구항 21]

대용량 파일의 분산 처리 방법.

하나 이상의 프로세서;

상기 프로세서에 의하여 수행 되는 컴퓨터 프로그램이

로드(load)된 메모리; 및

분산 처리 시스템에 속한 복수의 작업 노드에 연결된 네트워크

인터페이스를 포함하되,

상기 컴퓨터 프로그램은,

상기 복수의 작업 노드에 의하여 분산 처리되는 파일을 제1 영역

및 상기 제1 영역의 데이터보다 나중에 처리되는 제2 영역으로

구분하는 오퍼레이션;

상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 오퍼레이션;

상기 제2 영역을 고정 사이즈의 블록들로 분할하는 오퍼레이션; 및

제1 영역의 블록들 및 제2 영역의 블록들을 상기 복수의 작업

노드들에 상기 네트워크 인터페이스를 통하여 균등하게 나누어

송신하는 오퍼레이션을 포함하는,

대용량 파일의 분산 처리 관리 장치.

[청구항 22]

제21 항에 있어서,

상기 컴퓨터 프로그램은,

상기 작업 노드들의 블록 처리 현황을 모니터링 하여, 각각의 상기

작업 노드들의 미처리 블록 현황에 따라 상기 제2 영역의 블록을

대상으로 한 부하 리밸런싱 작업을 수행하는 오퍼레이션을 더

포함하는,

대용량 파일의 분산 처리 관리 장치.

[청구항 23]

컴퓨팅 장치와 결합하여,

분산 처리 시스템에 속한 복수의 작업 노드에 의하여 분산 처리

되는 파일을 제1 영역 및 상기 제1 영역의 데이터보다 나중에

처리되는 제2 영역으로 구분하는 단계;

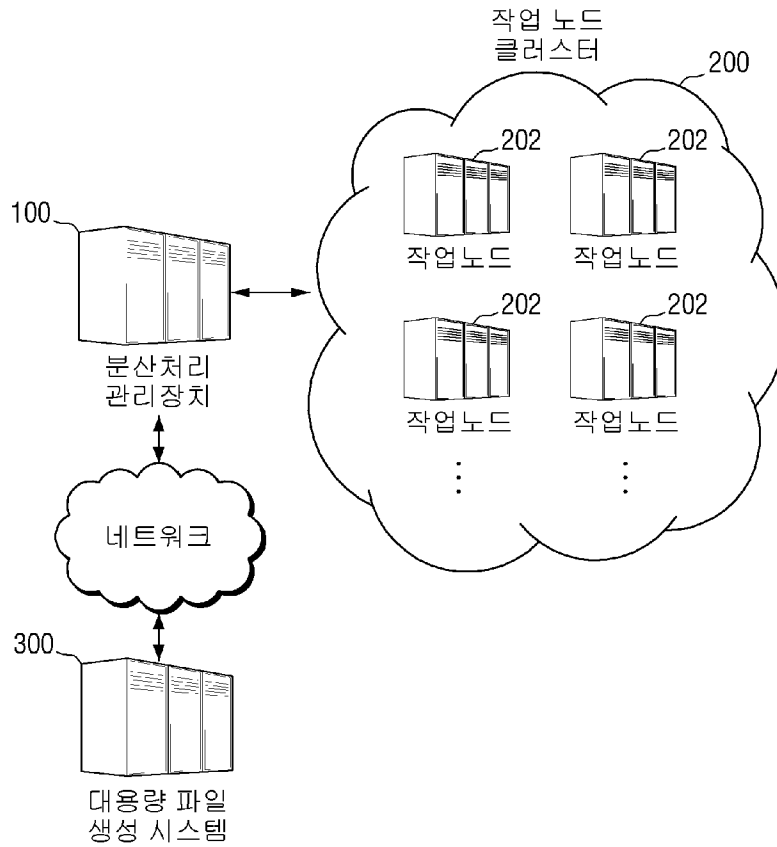
상기 제1 영역을 다양한 사이즈의 블록들로 분할하는 단계; 및

상기 제2 영역을 고정 사이즈의 블록들로 분할하는 단계를

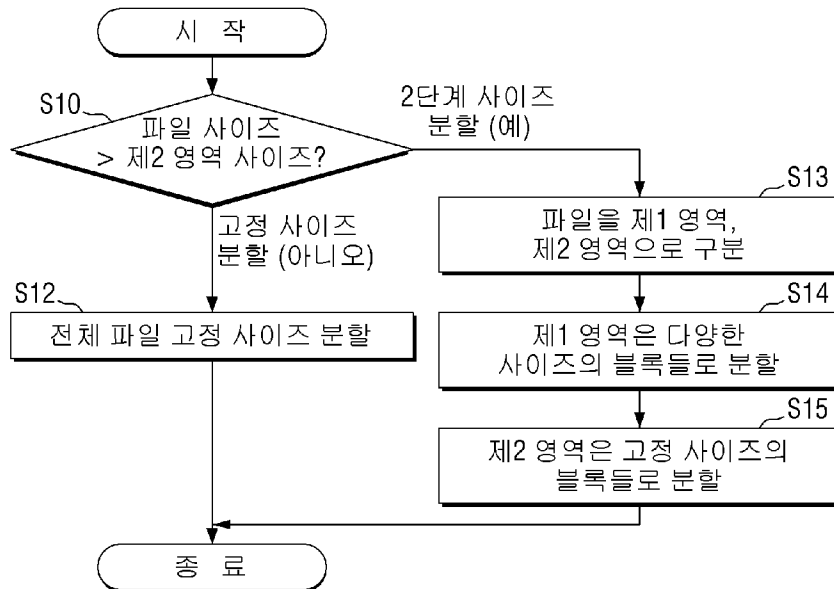
실행시키기 위하여 기록 매체에 저장된,

컴퓨터 프로그램.

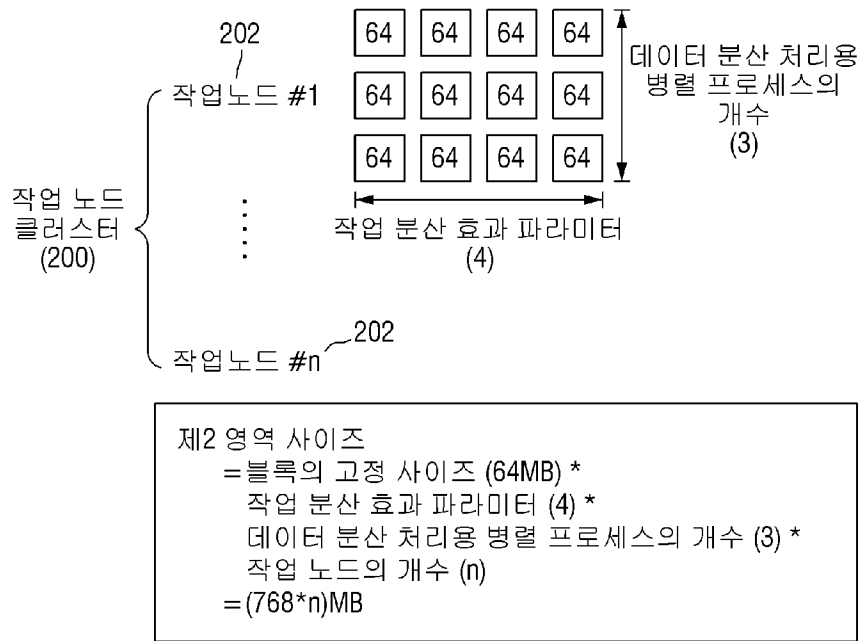
[Fig. 1]



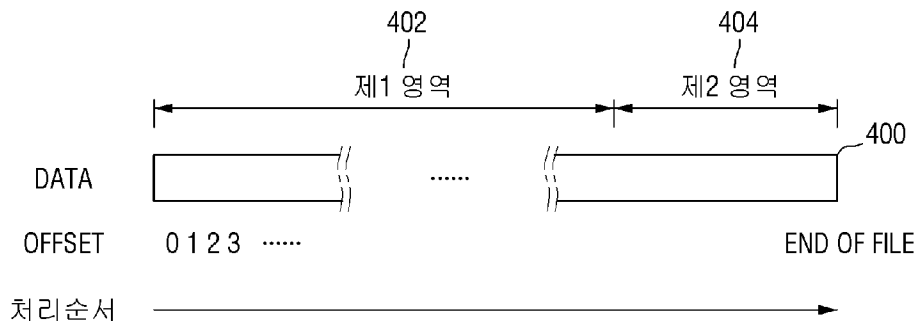
[Fig. 2]



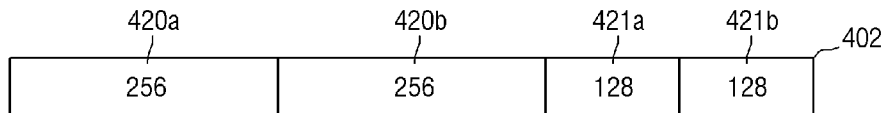
[Fig. 3]



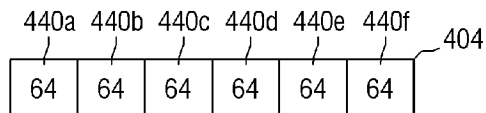
[Fig. 4]



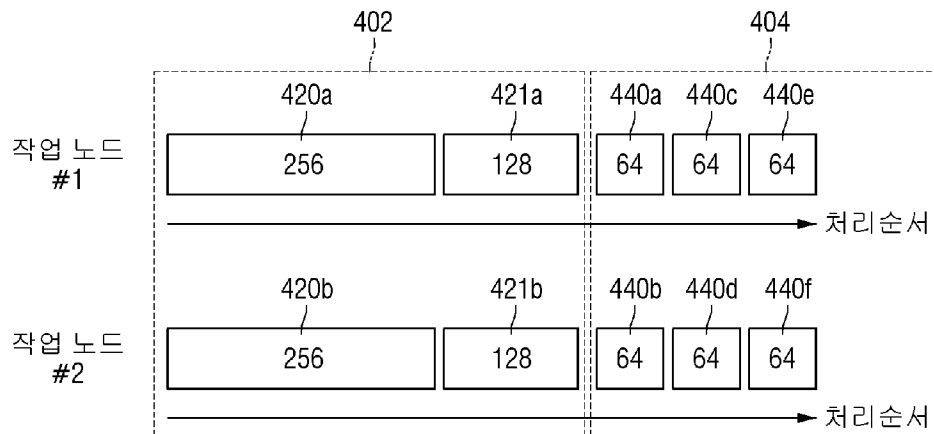
[Fig. 5a]



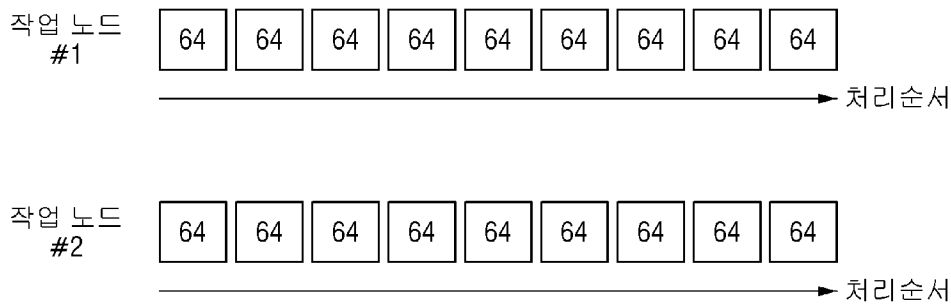
[Fig. 5b]



[Fig. 6]

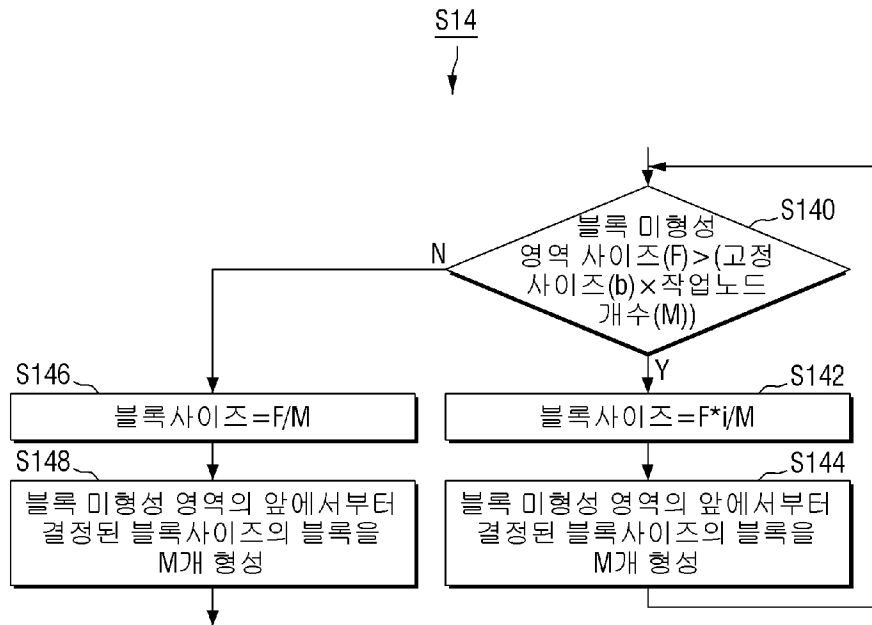


[Fig. 7]

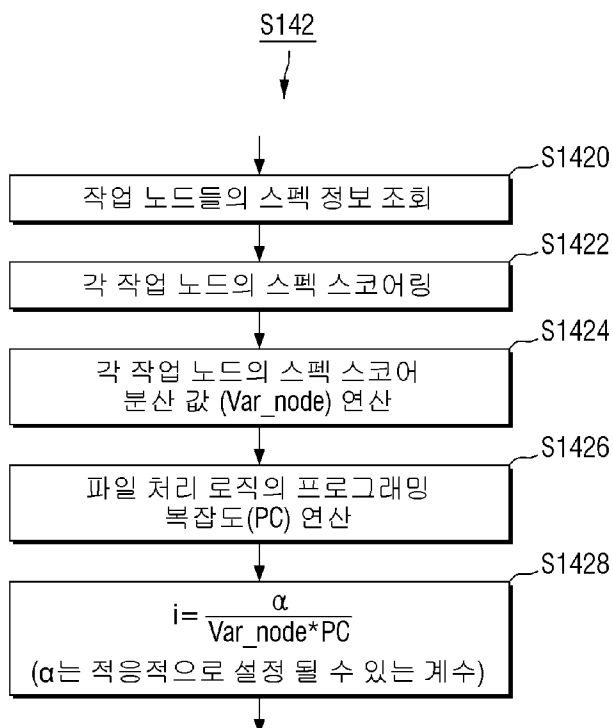


종래 기술

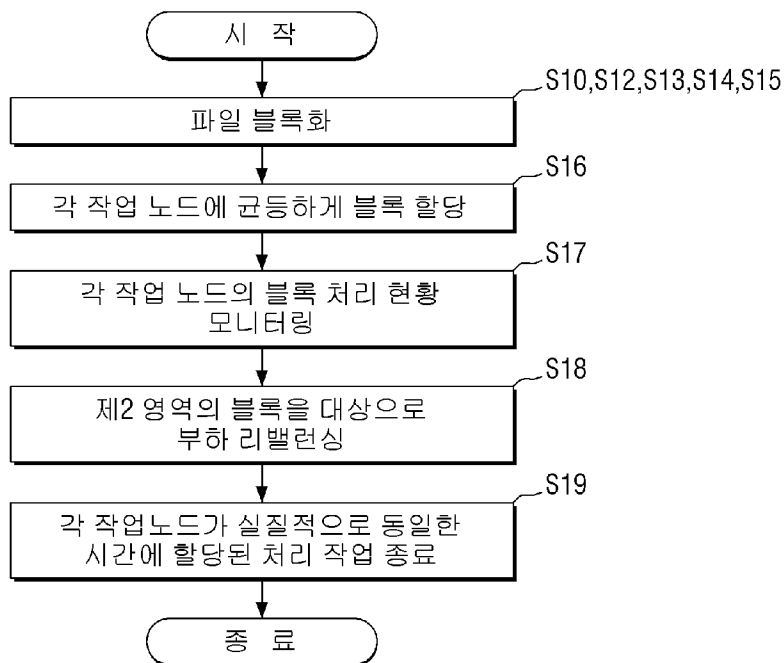
[Fig. 8]



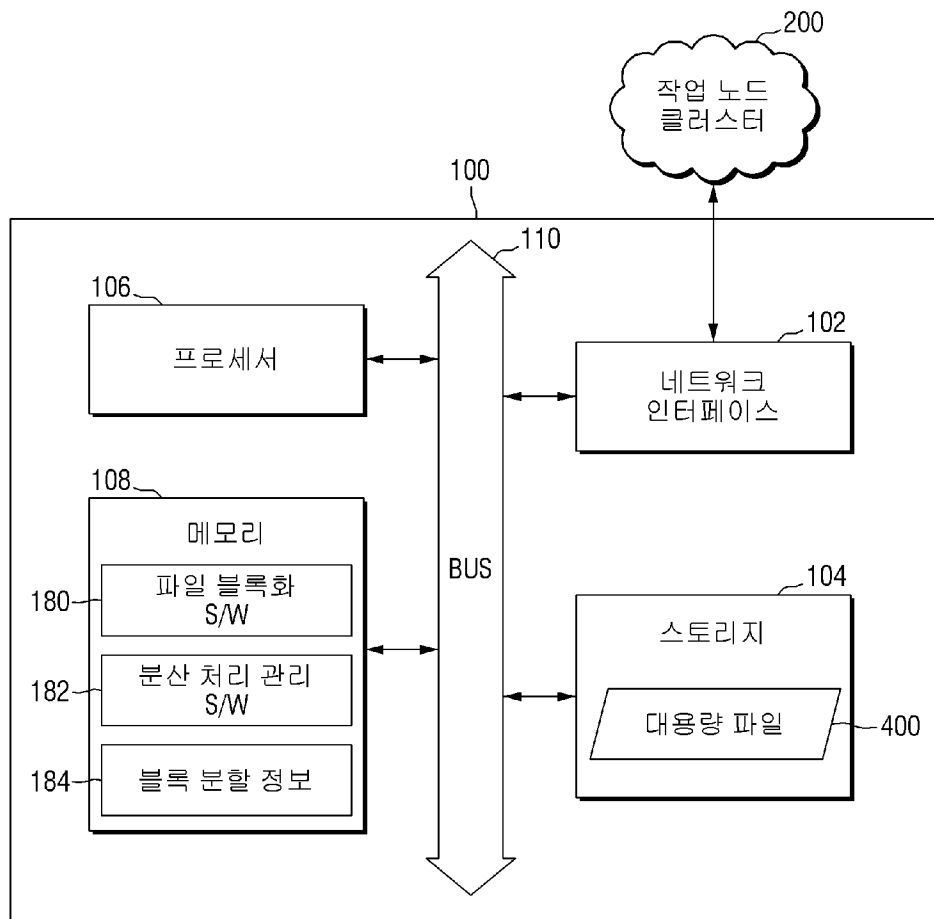
[Fig. 9]



[Fig. 10]



[Fig. 11]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2015/014136**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/30(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 17/30; H04L 12/911; G06F 3/06; G06F 7/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean Utility models and applications for Utility models: IPC as above

Japanese Utility models and applications for Utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & Keywords: distributing, processing, block, data, node, separation, size, size, region, group.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014-0359126 A1 (ADVANCED MICRO DEVICES, INC.) 04 December 2014 See paragraphs [0011], [0015]-[0021] and [0041]-[0044]; claim 1; and figures 1 and 5A-5B.	1-23
A	US 2012-0078844 A1 (KIM, Tae Woong) 29 March 2012 See paragraphs [0025]-[0027] and [0037]-[0040]; and figures 1 and 3.	1-23
A	WO 2011-128257 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 20 October 2011 See page 8, lines 15-29; page 10, line 8-page 11, line 21; and figures 1-2.	1-23
A	US 2013-0117273 A1 (LEE, Joo Young et al.) 09 May 2003 See paragraphs [0081]-[0087] and figure 6.	1-23
A	US 2012-0278587 A1 (CAUFIELD, Brian K. et al.) 01 November 2012 See paragraphs [0015]-[0021] and figures 1-3.	1-23
A	US 2006-0117071 A1 (SOBKO, Sergey) 01 June 2006 See paragraph [0052]; claims 7 and 13; and figure 3.	1-23



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

22 JULY 2016 (22.07.2016)

Date of mailing of the international search report

25 JULY 2016 (25.07.2016)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Seonsa-ro, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2015/014136

Patent document cited in search report	Publication date	Patent family member	Publication date
US 2014-0359126 A1	04/12/2014	NONE	
US 2012-0078844 A1	29/03/2012	JP 2012-074039 A KR 10-1374655 B1 KR 10-2012-0032920 A	12/04/2012 24/03/2014 06/04/2012
WO 2011-128257 A1	20/10/2011	CN 102844734 A CN 102844734 B DE 112011101317 T5 GB 2492870 A JP 2013-527524 A JP 5643421 B2 US 2011-0258378 A1 US 9021229 B2	26/12/2012 10/02/2016 31/01/2013 16/01/2013 27/06/2013 17/12/2014 20/10/2011 28/04/2015
US 2013-0117273 A1	09/05/2013	KR 10-2013-0049111 A US 8799291 B2	13/05/2013 05/08/2014
US 2012-0278587 A1	01/11/2012	WO 2012-146471 A1	01/11/2012
US 2006-0117071 A1	01/06/2006	KR 10-0694069 B1 KR 10-2006-0059568 A	12/03/2007 02/06/2006

A. 발명이 속하는 기술분류(국제특허분류(IPC))**G06F 17/30(2006.01)i****B. 조사된 분야**

조사된 최소문헌(국제특허분류를 기재)

G06F 17/30; H04L 12/911; G06F 3/06; G06F 7/00

조사된 기술분야에 속하는 최소문헌 이외의 문헌

한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))

eKOMPASS(특허청 내부 검색시스템) & 키워드: 분산, 처리, 블록, 데이터, 노드, 분할, 사이즈, 크기, 영역, 그룹.

C. 관련 문헌

카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
A	US 2014-0359126 A1 (ADVANCED MICRO DEVICES, INC.) 2014.12.04 단락 [0011], [0015]-[0021]과 [0041]-[0044]; 청구항 1; 및 도면 1과 5A-5B 참조.	1-23
A	US 2012-0078844 A1 (TAE WOONG KIM) 2012.03.29 단락 [0025]-[0027]과 [0037]-[0040]; 및 도면 1과 3 참조.	1-23
A	WO 2011-128257 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 2011.10.20 페이지 8, 라인 15-29; 페이지 10, 라인 8 - 페이지 11, 라인 21; 및 도면 1-2 참조.	1-23
A	US 2013-0117273 A1 (JOO YOUNG LEE 등) 2003.05.09 단락 [0081]-[0087] 및 도면 6 참조.	1-23
A	US 2012-0278587 A1 (BRIAN K. CAUFIELD 등) 2012.11.01 단락 [0015]-[0021] 및 도면 1-3 참조.	1-23
A	US 2006-0117071 A1 (SERGEY SOBKO) 2006.06.01 단락 [0052]; 청구항 7과 13; 및 도면 3 참조.	1-23

☐ 추가 문헌이 C(계속)에 기재되어 있습니다.☒ 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:

“A” 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌

“T” 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌

“E” 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌

“X” 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.

“L” 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌

“Y” 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.

“O” 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌

“&” 동일한 대응특허문헌에 속하는 문헌

“P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌

국제조사의 실제 완료일

2016년 07월 22일 (22.07.2016)

국제조사보고서 발송일

2016년 07월 25일 (25.07.2016)

ISA/KR의 명칭 및 우편주소



대한민국 특허청

(35208) 대전광역시 서구 청사로 189,

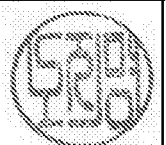
4동 (둔산동, 정부대전청사)

팩스 번호 +82-42-481-8578

심사관

노지명

전화번호 +82-42-481-8528



국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
US 2014-0359126 A1	2014/12/04	없음	
US 2012-0078844 A1	2012/03/29	JP 2012-074039 A KR 10-1374655 B1 KR 10-2012-0032920 A	2012/04/12 2014/03/24 2012/04/06
WO 2011-128257 A1	2011/10/20	CN 102844734 A CN 102844734 B DE 112011101317 T5 GB 2492870 A JP 2013-527524 A JP 5643421 B2 US 2011-0258378 A1 US 9021229 B2	2012/12/26 2016/02/10 2013/01/31 2013/01/16 2013/06/27 2014/12/17 2011/10/20 2015/04/28
US 2013-0117273 A1	2013/05/09	KR 10-2013-0049111 A US 8799291 B2	2013/05/13 2014/08/05
US 2012-0278587 A1	2012/11/01	WO 2012-146471 A1	2012/11/01
US 2006-0117071 A1	2006/06/01	KR 10-0694069 B1 KR 10-2006-0059568 A	2007/03/12 2006/06/02