(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0042873
(43) 공개일자 2015년04월21일

(51) 국제특허분류(Int. Cl.)
G06F 11/14 (2006.01) *G06F 9/46* (2006.01)
(52) CPC특허분류
G06F 11/1438 (2013.01)
G06F 9/46 (2013.01)
(21) 출원번호 10-2015-7008164(분할)
(22) 출원일자(국제) 2010년07월13일
심사청구일자 없음
(62) 원출원 특허 10-2012-7003290
원출원일자(국제) 2010년07월13일
심사청구일자 2015년02월17일
(85) 번역문제출일자 2015년03월30일
(86) 국제출원번호 PCT/US2010/041791
(87) 국제공개번호 WO 2011/008734
국제공개일자 2011년01월20일
(30) 우선권주장
12/502,851 2009년07월14일 미국(US)

(71) 출원인
아브 이니티오 테크놀로지 엘엘시
미국 02421 매사추세츠주 렉싱턴 스프링 스트리트 201
(72) 발명자
두로스 브라이언 필
미국 10701 매사추세츠주 프래밍험 레이크뷰 로드 92
애터버리 매튜 달시
미국 02421 매사추세츠주 렉싱턴 제라드 테라스 6
웨이클링 팀
미국 01810 매사추세츠주 앤도버 애봇 스트리트 11
(74) 대리인
유미특허법인

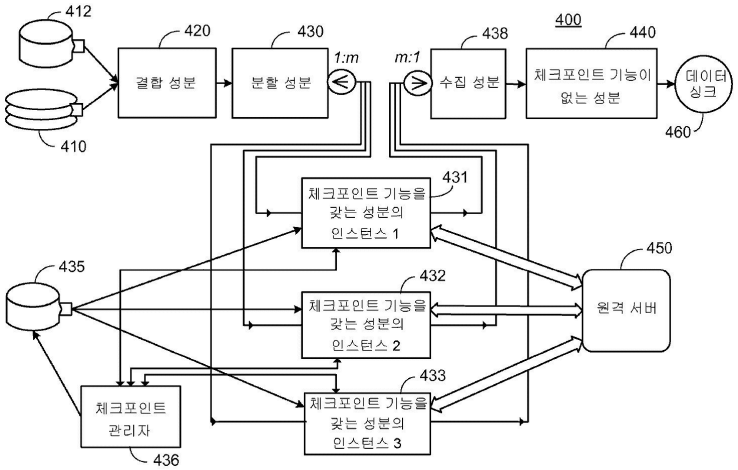
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **결함 내성 배치 처리**

(57) 요 약

입력 데이터의 배치의 처리는, 다수의 레코드를 포함하는 배치를 판독하는 과정과, 데이터 플로우 그래프를 통해 배치를 통과시키는 과정을 포함한다. 그래프 성분 중의 하나 이상이면서 전부보다는 적은 개수의 성분은, 하나 또는 둘 이상의 레코드와 연관된 다수의 작업 유닛의 각각에 대해 수행되는 동작에 대한 체크포인트 프로세스를 포함한다. 체크포인트 프로세스는, 배치에 대한 처리를 개시할 때에, 체크포인트 버퍼를 오픈하는 과정을 포함한다. 작업 유닛에 대한 동작의 수행 결과가 이미 체크포인트 버퍼에 세이브되어 있으면, 동작을 다시 수행하지 않고, 세이브된 결과를 사용하여 작업 유닛의 처리를 완료한다. 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 세이브되어 있지 않으면, 동작을 수행하여 작업 유닛의 처리를 완료하고, 동작의 수행 결과를 체크포인트 버퍼에 세이브한다.

대 표 도



명 세 서

청구범위

청구항 1

발명의 상세한 설명에 기재된, 또는 도면에 도시된 바와 같은 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 데이터의 배치(batch)를 결함 내성 방식으로 처리하는 것에 관한 것이다.

배 경 기 술

[0002] 복잡한 계산은 방향성 그래프(directed graph)를 사용하여 "데이터 플로우 그래프"(dataflow graph)로 표현될 수 있는데, 이러한 계산의 성분(component)은 그래프의 노드(또는 정점) 및 그래프의 노드 사이의 링크(또는 아크, 에지)에 대응하는 성분들 사이에서의 데이터 흐름과 관련되어 있다. 성분에는, 데이터를 처리하는 데이터 처리 성분과 데이터 흐름의 소스 또는 싱크(sink)로서 작용하는 성분이 포함된다. 데이터 처리 성분은 다수 스테이지의 데이터를 동시에 처리할 수 있는 파이프라인 시스템을 형성한다. 이러한 그래프 기반의 계산을 구현하는 시스템은, "EXECUTING COMPUTATIONS EXPRESSED AS GRAPHS"란 명칭의 미국특허 5,966,072호에 개시되어 있다. 일부의 경우에, 그래프 기반의 계산은 입력 데이터의 흐름을 수신하고, 데이터의 연속하는 흐름을 처리하여, 하나 이상의 성분으로부터의 결과를, 계산이 종료될 때까지 무기한으로 제공하도록 구성된다. 어떤 경우에는, 그래프 기반의 계산은 입력 데이터의 배치(batch)를 수신하고, 데이터의 배치를 처리하며, 해당 배치에 대한 결과를 제공하여, 배치의 처리가 완료된 후에 종료되거나 아이들 상태로 돌아가도록 구성된다.

발명의 내용

[0003] 하나의 관점에서, 일반적으로, 입력 데이터의 배치(batch)를 결함 내성(fault tolerant) 방식으로 처리하기 위한 방법은, 하나 또는 둘 이상의 데이터 소스로부터 다수의 레코드(record)를 포함하는 입력 데이터의 배치를 판독(read)하는 단계; 및 성분(component)들 사이의 데이터의 흐름을 나타내는 링크(link)에 의해 연결된 성분을 나타내는 둘 또는 셋 이상의 노드를 포함하는 데이터 플로우 그래프(dataflow graph)를 통해 배치를 통과시키는 단계를 포함하며, 성분 중의 하나 이상이면서 전부보다는 적은 개수의 성분은, 하나 또는 둘 이상의 레코드와 연관된 다수의 작업 유닛의 각각에 대해 수행되는 동작에 대한 체크포인트 프로세스(checkpoint process)를 포함한다. 체크포인트 프로세스는, 배치에 대한 처리를 개시할 때에, 비휘발성 메모리 내에 기억된 체크포인트 버퍼(checkpoint buffer)를 오픈하는 단계; 배치로부터의 각각의 작업 유닛에 대하여, 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 미리 세이브되어 있으면, 동작을 다시 수행하지 않고, 세이브되어 있는 결과를 사용하여 작업 유닛의 처리를 완료하고; 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 세이브되어 있지 않으면, 동작을 수행하여 작업 유닛의 처리를 완료하고, 동작의 수행 결과를 체크포인트 버퍼에 세이브하는 단계를 포함한다.

[0004] 본 발명의 관점은 이하의 특징 중의 하나 이상을 포함할 수 있다.

[0005] 동작(action)은 원격 서버와의 통신을 포함한다.

[0006] 동작의 수행 결과는 작업 유닛에 대한 원격 서버와의 통신으로부터의 정보를 포함한다.

[0007] 본 방법은, 배치의 처리가 완료되면, 체크포인트 버퍼를 삭제하는 단계를 더 포함한다.

[0008] 원격 서버와의 통신은 사용 요금이 부과(toll)된다.

[0009] 원격 서버와의 통신의 결과는, 휘발성 메모리 내에 기억되고, 트리거 이벤트가 발생한 경우 체크포인트 버퍼에 분류되어 세이브된다.

[0010] 트리거 이벤트는 체크포인트 관리자(checkpoint manager)로부터의 신호이다.

[0011] 트리거 이벤트는 체크포인트 버퍼에 대한 마지막 기록(last write) 이후에 다수의 레코드의 처리이다.

[0012] 트리거 이벤트는 체크포인트 버퍼에 대한 마지막 기록 이후의 시간의 경과이다.

[0013] 체크포인트 프로세스를 포함하는 성분은 다수의 처리 장치에서 병렬로 실행된다.

[0014] 다수의 병렬 처리 장치 중의 데이터 레코드의 할당은 배치의 실행 간에 일관되며, 각각의 처리 장치는 독립된 체크포인트 버퍼를 유지한다.

[0015] 다수의 병렬 처리 장치 중의 데이터 레코드의 할당은 동적이며, 처리 장치는 체크포인트 관리자에 의해 제어되는 체크포인트 버퍼에의 기록과 함께, 공유된 비휘발성 메모리 내에 기억된 단일의 체크포인트 버퍼에 대한 액세스를 공유한다.

[0016] 본 방법은, 결함 조건이 발생한 이후에 데이터 플로우 그래프 내의 모든 성분을 재시작하는 단계; 하나 또는 둘 이상의 데이터 소스로부터 다수의 레코드를 포함하는 입력 데이터의 배치를 판독하는 단계; 및 데이터 플로우 그래프를 통해 배치 모두를 통과시키는 단계를 더 포함한다.

[0017] 동작은 원격 서버와의 통신을 포함한다.

[0018] 다른 관점에서, 일반적으로, 컴퓨터로 판독가능한 매체는 입력 데이터의 배치(batch)를 결함 내성(fault tolerant) 방식으로 처리하기 위한 컴퓨터 프로그램을 기억한다. 컴퓨터 프로그램은 컴퓨터로 하여금, 하나 또는 둘 이상의 데이터 소스로부터 다수의 레코드(record)를 포함하는 입력 데이터의 배치를 판독(read)하도록 하고; 및 성분(component)들 사이의 데이터의 흐름을 나타내는 링크(link)에 의해 연결된 성분을 나타내는 둘 또는 셋 이상의 노드를 포함하는 데이터 플로우 그래프(dataflow graph)를 통해 배치를 통과시키도록 하는 명령어를 포함한다. 성분 중의 하나 이상이면서 전부보다는 적은 개수의 성분은, 하나 또는 둘 이상의 레코드와 연관된 다수의 작업 유닛의 각각에 대해 수행되는 동작에 대한 체크포인트 프로세스(checkpoint process)를 포함한다. 체크포인트 프로세스는, 배치에 대한 처리를 개시할 때에, 비휘발성 메모리 내에 기억된 체크포인트 버퍼(checkpoint buffer)를 오픈하는 단계; 및 배치로부터의 각각의 작업 유닛에 대하여, 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 미리 세이브되어 있으면, 동작을 다시 수행하지 않고, 세이브되어 있는 결과를 사용하여 작업 유닛의 처리를 완료하고; 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 세이브되어 있지 않으면, 동작을 수행하여 작업 유닛의 처리를 완료하고, 동작의 수행 결과를 체크포인트 버퍼에 세이브하는 단계를 포함한다.

[0019] 다른 관점으로서, 일반적으로, 입력 데이터의 배치(batch)를 결함 내성(fault tolerant) 방식으로 처리하기 위한 시스템은, 하나 또는 둘 이상의 데이터 소스로부터 다수의 레코드를 포함하는 입력 데이터의 배치를 수신하는 수단; 및 성분들 사이의 데이터의 흐름을 나타내는 링크에 의해 연결된 성분을 나타내는 둘 또는 셋 이상의 노드를 포함하는 데이터 플로우 그래프를 통해 상기 배치를 통과시키는 수단을 포함하며, 성분 중의 하나 이상이면서 전부보다는 적은 개수의 성분은, 하나 또는 둘 이상의 레코드와 연관된 다수의 작업 유닛의 각각에 대해 수행되는 동작에 대한 체크포인트 프로세스를 포함한다. 체크포인트 프로세스는, 배치에 대한 처리를 개시할 때에, 비휘발성 메모리 내에 기억된 체크포인트 버퍼를 오픈하는 단계; 및 배치로부터의 각각의 작업 유닛에 대하여, 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 미리 세이브되어 있으면, 동작을 다시 수행하지 않고, 세이브되어 있는 결과를 사용하여 작업 유닛의 처리를 완료하고; 작업 유닛에 대한 동작의 수행 결과가 체크포인트 버퍼에 세이브되어 있지 않으면, 동작을 수행하여 작업 유닛의 처리를 완료하고, 동작의 수행 결과를 체크포인트 버퍼에 세이브하는 단계를 포함한다.

[0020] 본 발명의 관점은 이하의 장점들 중의 하나 이상을 포함할 수 있다.

[0021] 데이터 플로우 그래프 내의 여러 성분 사이에서의 체크포인트와 관련된 통신에 대한 요구를 제거할 수 있다. 결함 복원 동안, 다중 단계의 배치 처리에서의 복잡하거나 비용이 많이 드는 단계를, 전체 파이프라인 시스템의 체크포인트를 실행하는 것에 의한 복잡도의 증가 및 비용을 발생시키지 않고도, 선택적으로 회피할 수 있다. 예를 들어, 본 발명의 방법은 요금이 부과된 서비스에 대한 반복된 호출을 회피함으로써 비용을 절약하는 데에 사용될 수 있다.

[0022] 본 발명의 다른 특징과 장점에 대해서는, 이하의 상세한 설명과 청구범위로부터 명백할 것이다.

도면의 간단한 설명

[0023] 도 1은 입력/출력 체크포인트 기능을 갖는 배치 데이터 처리 시스템의 블록도이다.

도 2는 체크포인트 프로세스의 플로차트이다.

도 3은 병렬처리 기능을 가진 입력/출력 체크포인트 기능을 갖는 배치 데이터 처리 시스템의 블록도이다.

도 4는 병렬 특성을 가진 입/출력 체크포인트 기능을 가진 배치 데이터 처리 시스템 및 체크포인트 관리자의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0024] 그래프 기반의 데이터 처리 시스템(graph-based data processing system)은, 데이터 플로우 그래프(dataflow graph)에서의 하나의 성분(component)의 중간 결과를 버퍼에 세이브하는 과정을 포함하여, 입력 데이터의 배치(batch)를 결함 내성(fault tolerant) 방식으로 처리하도록 구성될 수 있으며, 결함 조건(fault condition)에 의해 입력 데이터의 배치의 처리가 강제적으로 재시작되는 경우에, 버퍼로부터 중간 결과를 찾아내서 재사용할 수 있다.

[0025] 도 1은 데이터 처리 시스템(100)의 예를 나타내는 블록도이다. 데이터는 하나 또는 둘 이상의 데이터 소스(data source)로부터 하나 또는 둘 이상의 데이터 싱크(data sink)까지 데이터의 흐름을 처리하는 데이터 플로우 그래프의 데이터 처리 성분의 시퀀스를 통과한다. 데이터 플로우 그래프 내의 다양한 데이터 처리 성분이 별개의 처리 장치에서 실행되는 프로세스에 의해 구현되거나, 다수의 데이터 처리 성분이 단일의 처리 장치에서 실행되는 하나 또는 둘 이상의 프로세스에 의해 구현될 수 있다. 데이터는 데이터 처리 시스템(100)에 의해 처리되는 일련의 입력 데이터 레코드를 식별하는 배치 내에서 처리될 수 있다.

[0026] 데이터 처리 시스템(100)에 의한 데이터의 배치의 처리는 사용자 입력 또는 타이머 만료 등과 같은 다른 이벤트에 의해 개시될 수 있다. 데이터의 배치의 처리를 시작하면, 입력 데이터 레코드가 하나 이상의 입력 데이터 소스로부터 판독된다. 예를 들어, 입력 데이터는 데이터 저장 성분(110)으로 나타낸 것과 같은 컴퓨터로 판독가능한 기억 장치에 기억된 하나 이상의 파일로부터 판독될 수 있다. 입력 데이터 레코드는 데이터 저장 성분(112)으로 나타낸 것과 같이, 서버에서 실행되는 데이터베이스로부터도 판독될 수 있다. 결합 성분(join component)(120)은 다수의 데이터 소스로부터의 데이터(예를 들어, 레코드)를 순차적으로 판독하고, 입력 데이터를 이산적인 작업 유닛(discrete work unit)의 시퀀스로 정렬시킨다. 작업 유닛(work unit)은, 예를 들어 입력 레코드에 기초한 미리 정해진 포맷으로 기억된 레코드를 나타내거나, 처리할 트랜잭션(transaction)을 나타낼 수 있다. 일부 예에서, 각각의 작업 유닛은 처리되는 작업 유닛의 수와 같은, 배치 내의 고유한 개수만큼 식별될 수 있다. 작업 유닛은 데이터 플로우 그래프 내의 다음 성분으로 순차적으로 통과한다.

[0027] 데이터 처리 시스템(100)을 구현하는 데이터 플로우 그래프의 예는 데이터 처리 성분(130, 140)을 포함한다. 데이터 처리 성분(130)은 배치 처리 과정 중에 그 처리에 관한 상태 정보(state information)를 비휘발성 메모리(non-volatile memory)에 규칙적으로 세이브하는 체크포인트 프로세스(checkpoint process)를 포함한다. 결함 조건(fault condition)이 발생하고 배치가 재시작되어야 하는 경우에, 체크포인트된(checkpointed) 성분(130)은 기억된 상태 정보에 액세스하여, 배치의 반복 실행 동안 반복되어야 하는 처리량을 감소시킬 수 있다. 따라서, 체크포인트 처리는 결함 내성(fault tolerance)을 제공하기 위해 비휘발성 메모리 리소스를 사용하여야 하며, 데이터 처리 성분(130)이 복잡하게 된다. 데이터 처리 성분(140)은 체크포인트 기능이 없는 성분이다. 다른 데이터 플로우 그래프는 더 많은 수의 또는 더 적은 수의 데이터 처리 성분을 포함할 수 있다. 체크포인트 프로세스 기능을 포함하기 위해 필요에 따른 개수의 데이터 처리 성분이 구성될 수 있다. 일반적으로, 체크포인트 프로세스 기능을 포함하기 위해서는, 지연(delay) 또는 일부 다른 측정 기준(metric)에 대해 비용이 많이 드는 성분이 구성되기 때문에, 결함 조건이 발생한 경우에, 데이터 처리 시스템(100)에서 비용이 많이 드는 처리 단계를 배치 내의 모든 작업 유닛에 대하여 반복하지 않아도 된다.

[0028] 데이터 처리 성분(130)은 원격 서버(150)에 액세스하는 단계를 포함한다. 처리되는 각각의 작업 유닛에 대하여, 제1 처리 성분(130)은 요청을 원격 서버(150)에 발송하고 원격 서버로부터 결과(예를 들어, 데이터베이스로부터의 데이터)를 수신할 것이다. 이러한 동작은 원격 서버에 의해 제공되는 서비스 요금 부과 또는 원격 서버와의 통신에 생기는 네트워크 지연을 포함하는 다양한 이유에 대해 비용이 많이 들 수 있다. 결과를 수신한 후에, 데이터 처리 성분(130)은 다음 데이터 처리 성분(140)을 위한 출력을 생성한다. 이 데이터 처리 성분(130)은 체크포인트 프로세스를 포함하도록 구성되었기 때문에, 다음 데이터 처리 성분(140)에 작업 유닛을 위한 출력을 전달하고 다음 작업 유닛의 처리를 시작함으로써, 처리를 완료하기 전에 상태 정보를 처리하는 일부의 과정으로서, 원격 서버로부터의 결과를 세이브한다. 처리를 위한 상태 정보는 체크포인트 프로세스를 실행하는 처리 장치에서의 휘발성 메모리에 임시로 기억될 수 있다. 규칙적인 시간에, 하나 이상의 작업 유닛을 위한 처

리용 상태 정보가 비휘발성 메모리 내에 기억된 체크포인트 버퍼에 기록됨으로써, 나중에 결함 조건이 발생한 경우에 사용할 수 있게 된다.

[0029] 작업 유닛이 데이터 플로우 그래프의 데이터 처리 성분을 통해 진행함에 따라, 각각의 작업 유닛과 관련된 최종 결과가 데이터 싱크(160)로 전달된다. 작업 유닛은 개별적으로 전달되거나, 또는 다른 예에서는 최종 결과가 데이터 싱크(160)에 전달되기 전에, 작업 유닛이 최종 결과를 증분적으로 갱신하는 데에 사용되거나 축적(예를 들어, 대기 열로)될 수 있다. 데이터 싱크(160)는 작업 유닛에 기초하여 일부 축적된 출력이나 작업 유닛을 기억하는 데이터 저장 성분이 되거나, 또는 데이터 싱크(160)가 작업 유닛이 발행되는 대기 열(queue)이 되거나, 최종 결과를 수신하기 위한 다른 타입의 싱크가 될 수 있다. 배치 처리는 배치 내의 모든 작업 유닛에 대한 결과가 데이터 싱크(160)로 전달된 후에 종료한다. 이 시점에서, 데이터 플로우 그래프 내의 성분이 종료될 수 있다. 체크포인트된 성분과 관련된 체크포인트 프로세스는 종료 루틴(termination routine)의 일부로서 체크포인트 버퍼를 삭제할 수 있다.

[0030] 도 2는 체크포인트된 성분(checkpointed component)의 체크포인트 처리를 위한 프로세스(200)의 예에 대한 플로차트이다. 이 프로세스(200)는 데이터 플로우 그래프를 통한 배치 처리를 구현하는 소프트웨어로부터의 외부 호출(external call)이 있을 때에 개시한다(210). 개시 단계는 체크포인트된 성분이 실행되고 임의의 다른 필요한 리소스를 보류하는 처리 장치에서의 프로세스(200)에 대해 휘발성 메모리를 할당하는 과정을 포함할 수 있다. 다음으로, 프로세스(200)는 이러한 프로세스와 관련된 체크포인트 버퍼가 비휘발성 메모리에 이미 세이브되어 있는지 여부를 체크한다(단계 205). 체크포인트 버퍼가 존재하지 않으면, 비휘발성 메모리에 새로운 체크포인트 버퍼가 생성된다(단계 207). 체크포인트 버퍼가 이미 기억되어 있다면, 체크포인트 버퍼를 오픈한다(단계 208). 체크포인트 버퍼를 오픈하는 단계(208)는 비휘발성 메모리 내의 체크포인트 버퍼의 위치를 찾는 과정, 또는 처리 장치에서 체크포인트 버퍼의 일부 또는 모두를 휘발성 메모리에 복제하는 과정을 포함할 수 있다.

[0031] 각각의 작업 유닛을 처리하기 위한 루프를 시작할 때에, 작업 유닛과 관련된 입력 데이터가 데이터 플로우 그래프 내의 이전의 성분으로부터 또는 소스로부터 수신된다(단계 210). 작업 유닛에 대하여 전처리(pre-processing)(220)가 임의 선택적으로 수행된다. 전처리(220)는 작업 유닛과 관련된 결과를 위한 체크포인트 버퍼를 검색하기 위해 사용될 수 있는 값을 판정하거나 데이터 레코드를 재포맷(reformat)하는 과정을 포함할 수 있다. 체크포인트 프로세스(200)의 체크포인트 버퍼는 이러한 작업 유닛에 대한 결과(예를 들어, 인터럽트된 배치의 이전의 처리로부터)가 체크포인트 버퍼 내에 기억되어 있는지를 판정하기 위해 체크된다(단계 225).

[0032] 관련된 결과가 체크포인트 버퍼에 기억되어 있지 않다면, 처리 공정은 작업 유닛에 대해 비용이 많이 드는 동작(costly action)(230)을 포함한다. 비용이 많이 드는 동작의 예는, 네트워크를 통해 원격 서버에 있는 리소스에 액세스하는 과정과 중요한 지연을 생기게 하거나 비용을 부과하는 과정을 포함할 수 있다. 이러한 처리 과정의 결과는 체크포인트 버퍼 내에 기억된다(단계 240). 이 결과는 작업 유닛과 동일한 카운터 값에 의해 관련된 결과를 식별하는 증분하는 카운터를 사용하여 처리되는 작업 유닛과 관련될 수 있다. 결과는 비휘발성 메모리에 직접 기록되거나, 비휘발성 메모리에 복제되는 이벤트를 트리거할 때까지 휘발성 메모리 내에 임시로 버퍼링될 수 있다. 이벤트의 트리거링은 고정된 개수의 작업 유닛, 경과된 시간, 또는 외부 프로세스로부터의 신호를 처리하는 과정을 포함한다.

[0033] 관련된 결과가 체크포인트 버퍼에 기억되어 있다면, 그 결과를 체크포인트 버퍼로부터 꺼낸다(250).

[0034] 작업 유닛의 처리를 완료하기 위해 후처리(post-processing)(260)가 임의 선택적으로 수행된다. 후처리(260)는, 예를 들어 데이터 플로우 그래프에서 다음 성분에 데이터를 전달하거나 데이터를 재포맷하는 과정을 포함할 수 있다. 작업 유닛의 처리가 완료된 후에, 체크포인트 프로세스(200)는 다른 작업 유닛이 계속해서 처리되어야 하는지 여부를 체크한다(270). 다른 작업 유닛이 이용가능하다면, 체크포인트 프로세스(200)는 다음 작업 유닛과 관련된 입력 데이터를 판독하도록 루프백한다. 처리할 작업 유닛이 더 이상 남아 있지 않으면, 체크포인트 프로세스(200)는 배치 처리가 완료되었음을 나타내고 배치 처리를 종료할 것을 지시하는 외부 신호를 대기한다(280). 종료 신호를 수신하면, 체크포인트 프로세스(200)는 종료 시퀀스(290)를 완료하기 전에, 체크포인트 버퍼를 비휘발성 메모리로부터 삭제(285)한다. 종료 시퀀스(290)의 완료 단계는 휘발성 메모리를 처리 장치 또는 다른 보류된 리소스로부터 석방(release)하는 과정을 포함할 수 있다.

[0035] 도 3은 데이터 처리 시스템(300)의 블록도로서, 데이터 처리 시스템(300)을 구현하는 데이터 플로우 그래프가 분산된 체크포인트 프로세스 기능을 갖는 병렬 성분을 포함하는 도면이다. 데이터 플로우 그래프 내의 하나 이상의 성분은 다수의 처리 장치(예를 들어, 다수의 컴퓨터 또는 다수의 프로세서 또는 병렬 프로세서의 프로세서 코어)에서 병렬로 실행될 수 있다. 본 예에서, 체크포인트된 병렬 성분의 여러 인스턴스(instance)(331, 332,

333)가 명시적으로 도시되어 있다. 병렬 성분의 인스턴스는 각각의 처리 장치에서 실행되며 각각의 인스턴스는 배치 내의 작업 유닛의 서브세트를 처리한다. 이러한 분산된 체크포인트 방법의 예에서, 여러 상이한 체크포인트 프로세스가 병렬 성분의 3가지 인스턴스의 각각에 대해 실행된다.

[0036] 데이터의 배치의 처리가 개시되면, 입력 데이터 레코드가 하나 이상의 입력 데이터 소스로부터 판독된다. 예를 들어, 입력 데이터는 데이터 저장 성분(310)으로 나타낸 바와 같이, 컴퓨터로 판독가능한 기억 장치에 기억된 하나 이상의 파일로부터 판독될 수 있다. 입력 데이터 레코드는 데이터 저장 성분(312)에 의해 나타낸 바와 같이, 서버에서 실행되는 데이터베이스로부터 판독될 수 있다. 결합 성분(320)은 다수의 데이터 소스로부터 데이터를 순차적으로 판독하고, 입력 데이터를 이산적인 작업 유닛의 시퀀스로 정렬시킨다. 이어서, 작업 유닛은 데이터 플로우 그래프 내의 다음 성분으로 순차적으로 통과된다.

[0037] 데이터 플로우 그래프 내의 다음 데이터 처리 성분은 병렬 성분이기 때문에, 작업 유닛은 작업 유닛 분할 성분(330)에 의해 분할되어 다수의 성분 인스턴스에 할당된다. 본 예에서, 이러한 인스턴스에서의 작업 유닛의 할당은 여러 배치 처리 구동 사이에서 일관되기 때문에, 인스턴스는 다른 인스턴스에 할당된 작업 유닛에 대한 상태 정보를 액세스할 필요가 없다. 작업 유닛 분할 성분(330)은 결함 조건이 발생하고 배치를 다시 구동시킬 필요가 있는 경우에 일관된 결과를 가지고 반복될 수 있는 일관된 알고리즘(consistent algorithm)에 기초하여, 작업 유닛을 특정의 인스턴스에 할당한다. 예를 들어, 작업 유닛 할당 분할 성분(330)은 각각의 성분 인스턴스에 차례에 한번에 하나씩 작업 유닛을 할당하면 되며, 작업 유닛의 계수가 병렬 인스턴스의 개수를 초과하게 되면, 제1 인스턴스로 루프하게 된다. 다른 예에서, 작업 유닛 분할 성분(330)은 실행 사이의 일정한 할당을 제공하고 할당 정보를 세이브하도록 보장되지 않는 분할 알고리즘을 비휘발성 메모리에 인가할 수 있어서, 배치의 반복 실행이 필요한 경우, 동일한 할당이 반복될 수 있다.

[0038] 체크포인트된 병렬 성분의 각각의 인스턴스(331, 332, 333)는 도 1의 체크포인트된 성분(130)과 관련해서 언급한 방법을 사용하여 할당된 작업 유닛을 독립적으로 처리한다. 각각의 인스턴스(331, 332, 333)는 자신의 체크포인트 버퍼를 생성하여 비휘발성 메모리에 유지한다. 작업 유닛이 처리되는 경우, 인스턴스는 자신의 체크포인트 버퍼를 체크하여, 작업 유닛이 배치의 이전의 실행 동안에 이미 처리되었는지를 판정할 수 있다. 시스템(300)에서, 체크포인트된 병렬 성분은 각각의 작업 유닛에 대한 정보를 취득하기 위해 원격 서버(350)와 통신을 행하는 동작을 포함한다. 다른 예에서, 체크포인트된 병렬 성분은 결함 내성에 대한 체크포인트 버퍼의 유지를 정당화하는 것들과 관련된 높은 비용이 드는 다른 동작을 포함할 수 있다.

[0039] 작업 유닛의 처리가 완료된 경우, 결과는 수집 성분(gather component)(338)으로 전달되고, 이 수집 성분에서 다수의 인스턴스로부터의 결과를 수집하고, 데이터 플로우 그래프 내의 다음 데이터 처리 성분으로 전달한다.

[0040] 데이터 처리 성분(340)은 체크포인트 기능이 없는 성분이다. 다른 예에서, 데이터 플로우 그래프 내의 임의의 개수의 성분은 체크포인트 기능을 포함할 수 있다. 일부의 경우에, 비용이 많이 드는 동작이 수행되는 성분에 대해서는 체크포인트 프로세스 기능을 제한하는 것이 바람직하다. 다른 데이터 플로우 그래프는 임의의 소정의 데이터 성분에 대해 병렬 특성을 갖는 또는 병렬 특성을 갖지 않는 더 많은 수의 또는 더 적은 수의 데이터 처리 성분을 포함할 수 있다.

[0041] 작업 유닛이 데이터 플로우 그래프의 성분을 통해 진행함에 따라, 각각의 작업 유닛과 관련된 최종 결과가 데이터 싱크(360)로 전달된다. 배치 내의 모든 작업 유닛에 대한 결과가 데이터 싱크(360)로 전달되면, 배치 처리가 종료된다. 이 시점에서, 데이터 플로우 그래프 내의 성분과 관련된 프로세스가 중단될 수 있다. 소정의 인스턴스에 대한 체크포인트 프로세스는 중단 루틴의 일부로서 그 체크포인트 버퍼를 삭제할 수 있다.

[0042] 도 4는 데이터 처리 시스템(400)의 블록도로서, 시스템(400)을 구현하는 데이터 플로우 그래프가 중앙 집중식 체크포인트 프로세스 기능을 갖는 병렬 성분을 포함하는 것을 나타내는 도면이다. 본 예에서, 체크포인트된 병렬 성분의 다수의 인스턴스(431, 432, 433)가 명시적으로 도시되어 있다. 병렬화된 성분의 인스턴스는 각각의 처리 장치에서 실행되고 각각의 인스턴스는 배치 내의 작업 유닛의 서브세트를 처리한다. 이러한 중앙 집중식 체크포인트 방법의 예에서, 체크포인트 관리자(checkpoint manager)(436)는 병렬 성분의 인스턴스가 실행되는 처리 장치 중의 하나 또는 별개의 처리 장치에서 실행될 수 있다.

[0043] 데이터의 배치의 처리가 개시되면, 입력 데이터 레코드가 데이터 저장 성분(410, 412)으로부터 판독된다. 결합 성분(420)은 다수의 데이터 소스로부터 데이터를 순차적으로 판독하고, 입력 데이터를 기억된 이산적인 작업 유닛의 시퀀스로 정렬시킨다. 이어서, 작업 유닛은 데이터 플로우 그래프 내의 다음 성분, 본 예에서는 체크포인트된 병렬 성분까지 순차적으로 통과된다.

[0044] 도 4의 예에서, 체크포인트 관리자(436)는 여러 처리 장치에서 실행되는 인스턴스(431, 432, 433)에 의해 공유되는 단일의 체크포인트 버퍼에 대한 액세스를 제어한다. 배치 내의 모든 작업 유닛에 대한 단일의 체크포인트 버퍼를 공유함으로써, 작업 유닛은 배치의 이전 실행으로부터 할당을 매칭시킬 필요 없이, 인스턴스에 동적으로 할당될 수 있다. 공유된 체크포인트 버퍼는 공유된 비휘발성 메모리(435)에 기억되며, 이 공유된 비휘발성 메모리에서 모든 인스턴스가 버스 또는 통신 네트워크를 통해 직접 액세스할 수 있거나, 체크포인트 관리자(436)에 의해 통신 네트워크를 통해 간접적으로 액세스할 수 있다. 인스턴스(431, 432, 433)는 공유된 비휘발성 메모리(435)를 판독하여, 이들 인스턴스가 작업 유닛을 처리할 때에 체크포인트 버퍼를 체크할 수 있다. 현재의 작업 유닛에 대한 결과가 체크포인트 버퍼 내에서 발견되면, 기억된 결과를 사용하여 비용이 많이 드는 동작을 반복하는 것을 피할 수 있다. 현재의 작업 유닛에 대한 결과를 체크포인트 버퍼 내에서 발견하지 못하면, 작업 유닛에 대한 동작이 실행되고 체크포인트 버퍼에 결과가 기억된다. 체크포인트 버퍼에 기록하기 위해, 인스턴스(431, 432, 433)는 기록 요청 메시지를 체크포인트 관리자(436)에 발송한다. 체크포인트 관리자(436)는 체크포인트 버퍼를 갱신하기 위해 공유된 비휘발성 메모리(435)에 기록한다. 다른 실시예에서, 체크포인트 관리자(436)는 체크포인트 버퍼를 갱신하기 위해 공유된 비휘발성 메모리(435)에 대한 기록을 허가하는 요청 인스턴스에 토큰을 발송한다.

[0045] 공유된 체크포인트 버퍼는 모든 인스턴스(431, 432, 433)에 의해 사용되기 때문에, 작업 유닛 분할 성분(430)은 데이터의 배치의 각각의 실행 동안 인스턴스 사이에서 작업 유닛을 서로 상이하게 동적으로 할당할 수 있다. 예를 들어, 작업 유닛 분할 성분(430)은 실행마다 다를 수 있는 실행 시간에 각각의 처리 장치에서의 이용가능한 용량에 기초하여 각각의 작업 유닛을 동적으로 할당할 수 있다. 이 방법에 의하면, 작업 유닛 분할 성분(430)에 의해 상이한 개수의 병렬 인스턴스를 사용할 수 있다. 예를 들어, 결함 조건 이후에, 인스턴스(433)와 같이, 병렬 성분의 인스턴스를 구동시키는 처리 장치 중의 하나가 디스에이블 상태로 되거나 그외 다른 방식으로 사용할 수 없게 될 수 있다. 이 경우, 배치가 재시작되면, 작업 유닛 분할 성분(430)은 모든 작업 유닛을, 디스에이블된 인스턴스(433)에 의해 이미 처리된 작업 유닛에 대한 체크포인트 버퍼 엔트리에 끊어짐 없이(seamless) 액세스할 수 있는, 남아 있는 인스턴스(431, 432, 433)에 할당할 수 있다.

[0046] 체크포인트 관리자(436)는 개별의 처리 장치에서 실행되는 프로세스에 의해 실현되거나, 병렬 성분의 인스턴스가 실행되는 처리 장치 중의 하나에서 실행되는 프로세스에 의해 실현될 수 있다. 인스턴스(431, 432, 433)는 체크포인트 버퍼 갱신 이벤트 사이에서 로컬 휘발성 메모리 내의 체크포인트 버퍼 갱신을 버퍼링할 수 있다. 체크포인트 관리자(436)는 휘발성 메모리 내에 버퍼링된 임의의 정보로 체크포인트 버퍼 갱신을 개시하도록 인스턴스를 트리거하는 신호를 인스턴스에 발송할 수 있다.

[0047] 작업 유닛의 처리가 완료되면, 결과는 수집 성분(438)으로 전달된다. 이 수집 성분에서는, 다수의 인스턴스로부터의 결과를 수집해서 데이터 플로우 그래프 내의 다음 데이터 처리 성분으로 전달한다.

[0048] 데이터 처리 성분(440)은 체크포인트 기능이 없는 성분이다. 다른 예에서, 데이터 플로우 그래프 내의 임의의 개수의 성분은 체크포인트 기능을 포함할 수 있다. 일부의 경우에는, 비용이 많이 드는 동작이 수행되는 성분에 대해 체크포인트 프로세스 기능을 제한하는 것이 바람직하다. 다른 데이터 플로우 그래프는 임의의 소정의 데이터 처리 성분에 대해 병렬 특성을 갖는 또는 갖지 않는 더 많은 수의 또는 더 적은 수의 처리 성분을 포함할 수 있다.

[0049] 작업 유닛이 데이터 플로우 그래프의 성분을 통해 진행함에 따라, 각각의 작업 유닛과 관련된 최종 결과가 데이터 싱크(460)로 전달된다. 배치 내의 모든 작업 유닛에 대한 결과가 데이터 싱크(460)로 전달되면, 배치 처리가 종료된다. 이 시점에서, 데이터 플로우 그래프 내의 성분과 관련된 프로세스가 종료될 수 있다. 체크포인트 관리자(436)는 종료 루틴의 일부로서 체크포인트 버퍼를 삭제할 수 있다.

[0050] 이상 개시한 결함 내성 배치 처리 방법은 컴퓨터에서의 실행을 위한 소프트웨어를 사용하여 구현될 수 있다. 예를 들어, 소프트웨어는 하나 이상의 프로세서, 하나 이상의 데이터 기억 시스템(휘발성 및 비휘발성 메모리 및/또는 기억 요소를 포함), 하나 이상의 입력 장치 또는 포트, 및 하나 이상의 출력 장치 또는 포트를 각각 포함하는 하나 이상의 프로그램된 또는 프로그램가능한 컴퓨터 시스템(분산형, 클라이언트/서버형, 또는 그리드형의 여러 구조가 될 수 있음)에서 실행되는 하나 이상의 컴퓨터 프로그램 내의 프로시저를 형성한다. 소프트웨어는 연산 그래프의 설계 및 구성에 관련된 다른 서비스를 제공하는 더 큰 프로그램의 하나 이상의 모듈을 형성할 수 있다. 그래프의 노드 및 요소는 컴퓨터로 판독가능한 매체에 기억된 데이터 구조 또는 데이터 레포지토리에 기억된 데이터 모델에 일치하는 다른 구조화된 데이터로서 구현될 수 있다.

[0051] 소프트웨어는 범용 또는 전용의 프로그래머블 컴퓨터에 의해 판독가능한 CD-ROM 과 같은 저장 매체로 제공되거

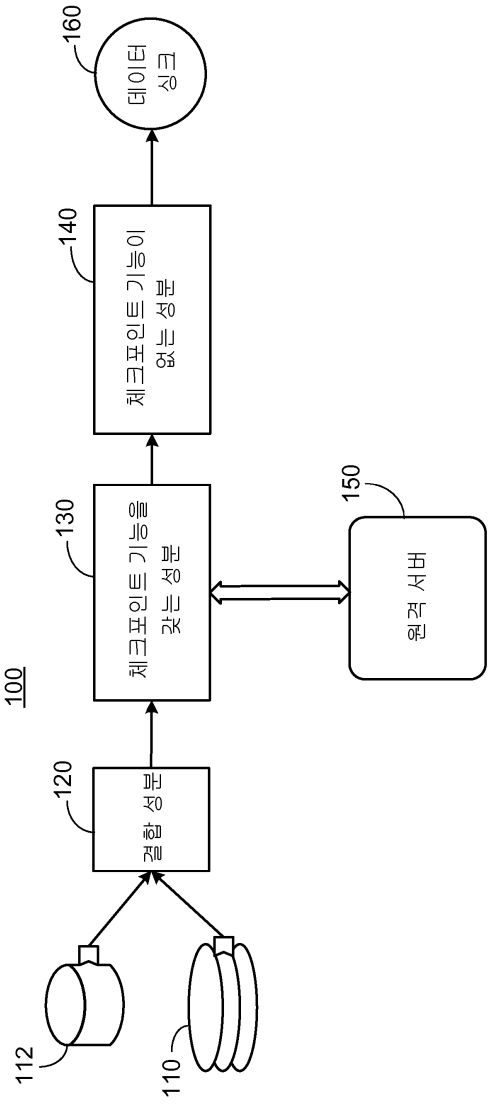
나, 그 소프트웨어가 실행되는 컴퓨터에 네트워크의 통신매체를 통해 전달(전파 신호에서 부호화되어)될 수도 있다. 모든 기능은 전용 컴퓨터상에서, 또는 코프로세서와 같은 전용 하드웨어를 사용하여 수행될 수도 있다. 소프트웨어에 의해 사양화된 연산의 여러 다른 부분이 여러 다른 컴퓨터에 의해 수행되는 분산 방법으로 소프트웨어를 구현할 수도 있다. 컴퓨터 시스템에 의해 저장매체 또는 저장장치가 판독되어 위에 설명한 과정이 수행될 때 그 컴퓨터를 구성하여 동작시키기 위해서는, 이러한 컴퓨터 프로그램 각각은 범용 또는 전용 프로그래머블 컴퓨터에 의해 판독가능한 저장매체 또는 저장장치(예를 들면 솔리드 스테이트 메모리, 또는 솔리드 스테이트 매체, 또는 마그네틱 매체 또는 광학 매체)에 저장되거나 다운로드되는 것이 바람직하다. 본 발명의 진보성 있는 시스템은 또한 컴퓨터 프로그램으로 구성되는 컴퓨터 판독가능 저장매체로서 구현될 수도 있으며, 이 경우, 이와 같이 구성된 저장매체는 컴퓨터 시스템으로 하여금 특정의 미리 규정된 방법으로 동작하여 본 명세서에서 설명하는 기능을 수행하도록 작용한다.

[0052] 이상과 같이, 여러 실시예를 설명하였지만, 본 발명의 사상 및 그 범위를 일탈하지 않고도 다양한 변형이 가능함을 알 수 있다. 예를 들면, 상기 설명한 단계들 중 일부는 순서와 무관한 것일 수도 있으며, 이에 따라서 상기 설명한 것과는 다른 순서로 수행될 수도 있다.

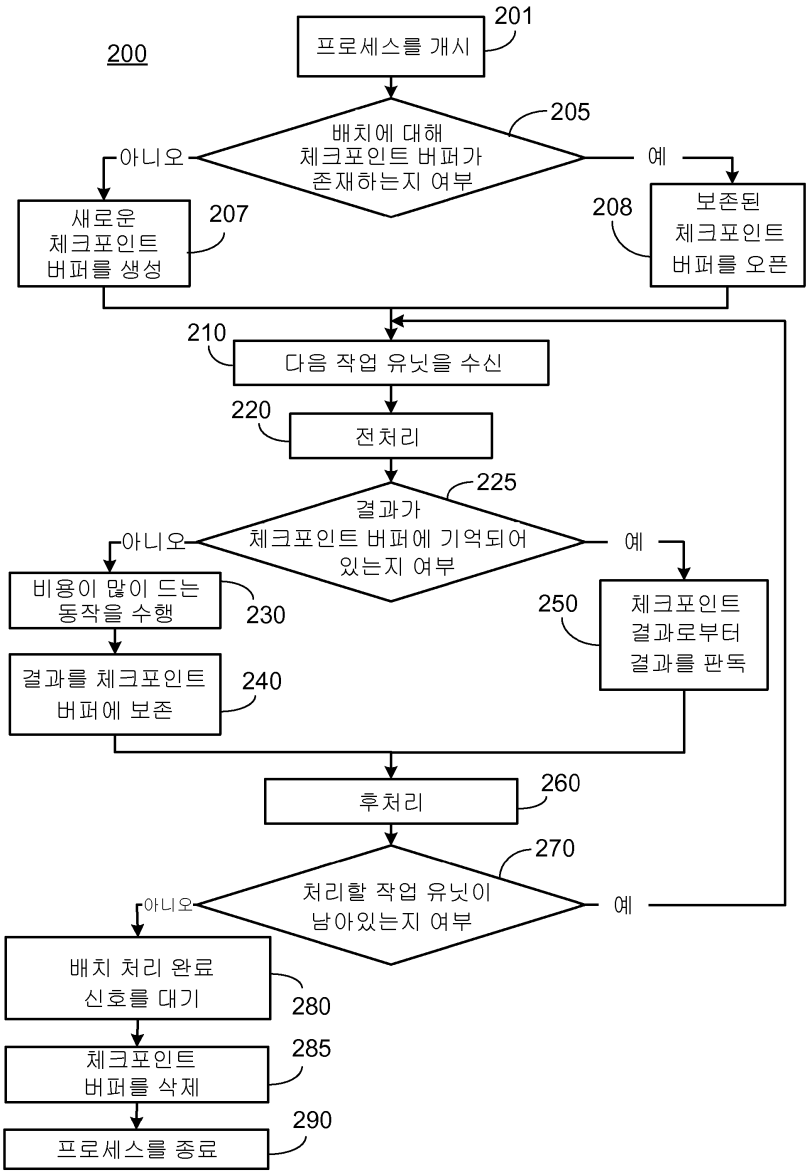
[0053] 상기 설명한 내용은 예시를 위한 것으로서 본 발명의 범위를 한정하기 위한 것이 아니며, 그 범위는 다음의 특허청구범위에 의해 규정된다. 예를 들면, 상기 설명한 여러 기능 단계는 전체 프로세스에 실질적인 영향을 주지 않고 다른 순서로 수행될 수 있으며, 그외의 다른 실시예도 이하의 청구범위 내에 포함된다.

도면

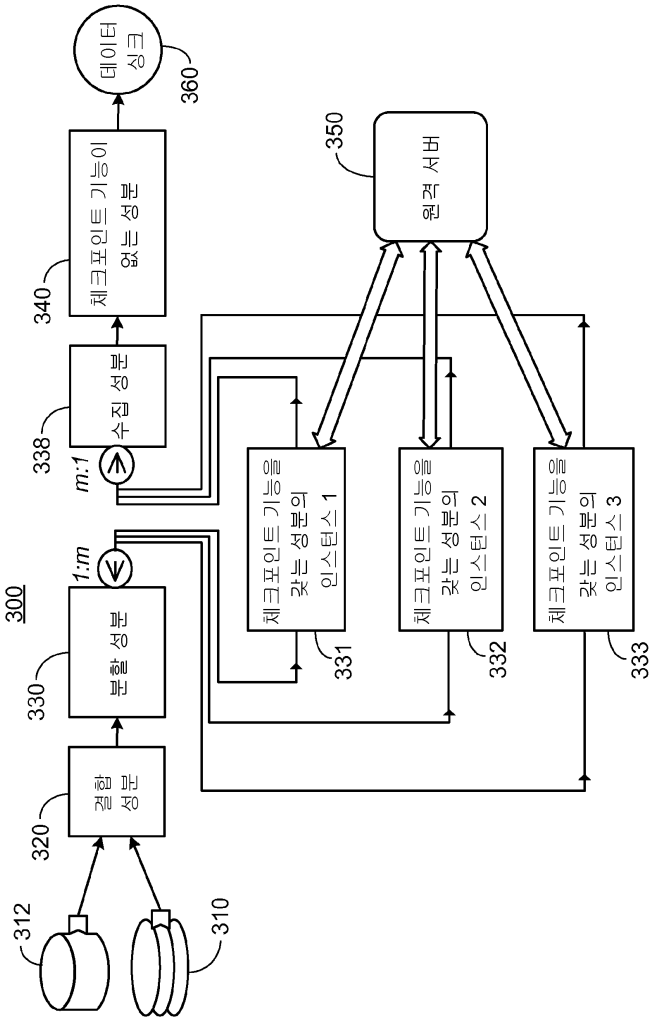
도면1


100
112
110
120
결합 성분
130
체크포인트 기능을 갖는 성분
140
체크포인트 기능이 없는 성분
150
원격 서버
160
데이터 싱크

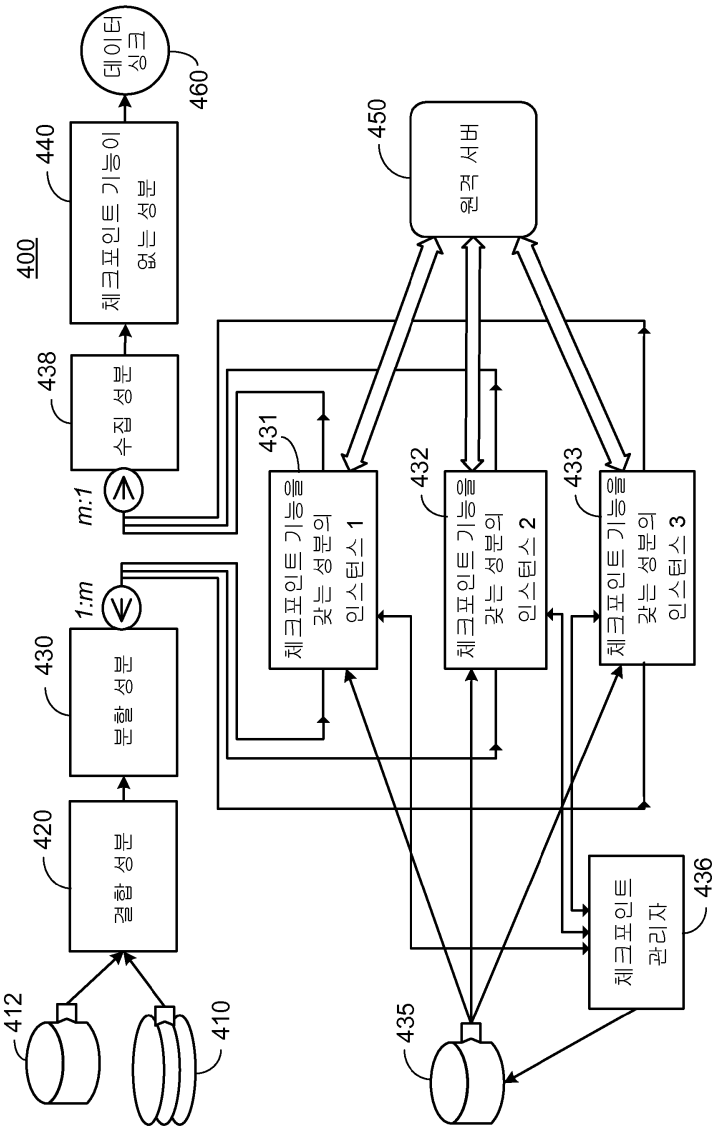
도면2


200
프로세스를 개시
201
배치에 대해 체크포인트 버퍼가 존재하는지 여부
205
아니오
예
새로운 체크포인트 버퍼를 생성
207
보존된 체크포인트 버퍼를 오픈
208
다음 작업 유닛을 수신
210
전처리
220
결과가 체크포인트 버퍼에 기억되어 있는지 여부
225
아니오
예
비용이 많이 드는 동작을 수행
230
체크포인트 결과로부터 결과를 판독
250
결과를 체크포인트 버퍼에 보존
240
후처리
260
처리할 작업 유닛이 남아있는지 여부
270
아니오
예
배치 처리 완료 신호를 대기
280
체크포인트 버퍼를 삭제
285
프로세스를 종료
290

도면3


데이터 싱크
360
350
원격 서버
340
체크포인트 기능이 없는 성분
338
수집 성분
m:1
1:m
300
330
분할 성분
320
결합 성분
312
310
체크포인트 기능을 갖는 성분의 인스턴스 1
331
체크포인트 기능을 갖는 성분의 인스턴스 2
332
체크포인트 기능을 갖는 성분의 인스턴스 3
333

도면4


데이터 싱크
460
400
440
체크포인트 기능이 없는 성분
438
수집 성분
m:1
1:m
430
분할 성분
420
결합 성분
412
410
450
원격 서버
431
체크포인트 기능을 갖는 성분의 인스턴스 1
432
체크포인트 기능을 갖는 성분의 인스턴스 2
433
체크포인트 기능을 갖는 성분의 인스턴스 3
436
체크포인트 관리자
435