(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년05월12일
(11) 등록번호 10-1734883
(24) 등록일자 2017년05월04일

(51) 국제특허분류(Int. Cl.)
G06F 12/00 (2016.01)
(21) 출원번호 10-2013-7019992
(22) 출원일자(국제) 2012년02월16일
심사청구일자 2015년02월10일
(85) 번역문제출일자 2013년07월26일
(65) 공개번호 10-2014-0003517
(43) 공개일자 2014년01월09일
(86) 국제출원번호 PCT/US2012/025441
(87) 국제공개번호 WO 2012/112772
국제공개일자 2012년08월23일
(30) 우선권주장
13/031,034 2011년02월18일 미국(US)
(56) 선행기술조사문헌
JP04205418 A*
JP10200574 A*
JP2010211880 A*
KR1020060048243 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
아브 이니티오 테크놀로지 엘엘시
미국 02421 매사추세츠주 렉싱턴 스프링 스트리트 201
(72) 발명자
스탠필 크레이그 더블유.
미국 01773 매사추세츠주 린콜린 헉클베리 힐 로드 43
파인만 칼 리차드
미국 02468 매사추세츠주 웨이번 모스필드 로드 21
(74) 대리인
유미특허법인

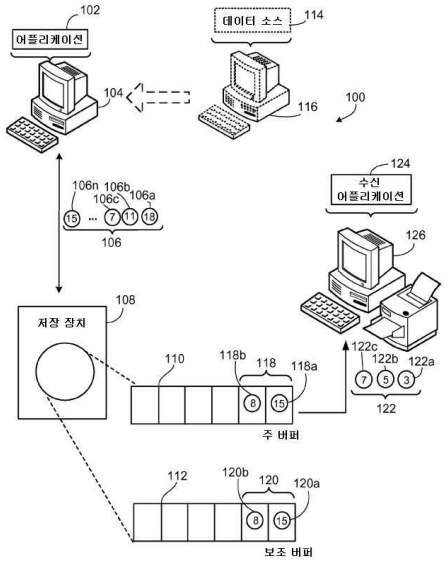
전체 청구항 수 : 총 77 항 심사관 : 황철규

(54) 발명의 명칭 **버퍼 오버플로우 상태 관리**

(57) 요 약

몇 가지 구현예에서, 시스템 및 기술은 제1 버퍼(110)의 오버플로우 상태 검출(306)에 응답하여 제1 버퍼에 있는 데이터 요소(data element)의 일부분을 제2 버퍼(112)에 저장하고(308), 제2 버퍼에 저장된 데이터 요소의 일부분을 나타내기 위하여 제1 버퍼에 프록시 데이터 요소를 삽입하는, 컴퓨터 구현 방법을 포함하고, 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅된다.

대 표 도 - 도1



명 세 서

청구범위

청구항 1

컴퓨터 구현 방법에 있어서,

제1 버퍼의 오버플로우(overflow) 상태의 감지에 응답하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소(data element)를 제2 버퍼에 이동시키는 단계 - 상기 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅됨(sorted) -;

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소를 나타내기 위하여 상기 제1 버퍼에 프록시(proxy) 데이터 요소를 생성하는 단계;

상기 제1 버퍼 내의 프록시 데이터 요소의 위치에 기초하여 상기 제2 버퍼 내에 저장된 데이터 요소 중 하나 이상을 상기 제1 버퍼에 이동시키는 단계 - 상기 제1 버퍼에 이동된 데이터 요소는 상기 미리 정해진 순서에 따라 상기 제1 버퍼 내에 저장됨 -;

상기 제1 버퍼에 저장된 데이터 요소에 대해 경계 조건을 적용하는 단계; 및

상기 경계 조건의 적용 결과에 기초하여, 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 단계

를 포함하는,

컴퓨터 구현 방법.

청구항 2

제1항에 있어서,

상기 제1 버퍼 내의 데이터 요소의 상기 미리 정해진 순서는 상기 제1 버퍼로부터 상기 데이터 요소를 방출하기 위한 우선순위 순서를 포함하는, 컴퓨터 구현 방법.

청구항 3

제1항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 미리 정해진 순서에 따라 실질적으로 소팅되는, 컴퓨터 구현 방법.

청구항 4

제1항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 제1 버퍼에 저장된 데이터 요소의 적어도 절반을 포함하는, 컴퓨터 구현 방법.

청구항 5

제1항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소에 대응하는 상기 제1 버퍼의 위치에 데이터 요소를 저장하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 6

제1항에 있어서,

상기 미리 정해진 순서에 따라 상기 제1 버퍼로부터 상기 하나 이상의 데이터 요소를 방출하는 단계; 및

상기 제1 버퍼로부터 방출된 하나 이상의 데이터 요소를 상기 미리 정해진 순서에 따라 실질적으로 소팅된 데이

터 요소의 출력 스트림(output stream)으로 제공하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 7

제2항에 있어서,

상기 프록시 데이터 요소가 상기 제1 버퍼 내에서 가장 높은 우선순위의 데이터 요소가 되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 8

제1항에 있어서,

상기 프록시 데이터 요소가 상기 제1 버퍼로부터의 제거를 위해 식별되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 9

제7항에 있어서,

병합된 데이터 요소가 상기 미리 정해진 순서에 기초하여 소팅되는, 컴퓨터 구현 방법.

청구항 10

제7항 또는 제8항에 있어서,

병합이 출력 데이터 요소의 생성과 동시에 일어나는, 컴퓨터 구현 방법.

청구항 11

제1항에 있어서,

상기 프록시 데이터 요소는 상기 제2 버퍼로부터의 제거를 위한 가장 높은 우선순위를 갖는 상기 제2 버퍼로부터의 데이터 요소의 복사본(copy)인, 컴퓨터 구현 방법.

청구항 12

제6항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 작은 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 컴퓨터 구현 방법.

청구항 13

제6항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 큰 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 컴퓨터 구현 방법.

청구항 14

삭제

청구항 15

제1항에 있어서,

각각의 데이터 요소는 데이터 레코드 생성 시간을 나타내는, 컴퓨터 구현 방법.

청구항 16

제1항에 있어서,

상기 경계 조건을 상기 제1 버퍼에 저장된 데이터 요소에 대해 적용하는 단계는,

상기 제1 버퍼에 저장된 데이터 요소의 가장 작은 값을 나타내는 데이터 요소를 식별하는 단계; 및

상기 가장 작은 값을 나타내는 데이터 요소와 상기 제1 버퍼에 입력하기 위해 수신된 데이터 요소 간의 차이를 판단하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 17

제1항에 있어서,

상기 경계 조건에 기초하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 단계는,

하나 이상의 데이터 요소가 미리 정해진 값을 초과하는지를 판단하는 단계; 및

상기 하나 이상의 데이터 요소가 미리 정해진 값을 초과하면, 상기 하나 이상의 데이터 요소를 상기 제1 버퍼로부터 방출하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 18

제1항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 스킵 리스트 데이터 구조(skip list data structure)에 의해 구현되는, 컴퓨터 구현 방법.

청구항 19

제1항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 트립 데이터 구조(treap data structure)에 의해 구현되는, 컴퓨터 구현 방법.

청구항 20

제1항에 있어서,

상기 제1 버퍼 및 제2 버퍼 내의 개개의 데이터 요소의 하나 이상은 그래프 기반 연산 시스템(graph-based computation system)의 컴포넌트로 입력되는 데이터 레코드로부터 도출되는, 컴퓨터 구현 방법.

청구항 21

데이터 요소를 소팅하기 위한 컴퓨터 프로그램이 저장된 컴퓨터 판독가능 저장 매체로서,

상기 컴퓨터 프로그램은, 컴퓨팅 시스템으로 하여금,

제1 버퍼의 오버플로우 상태 검출에 응답하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 제2 버퍼에 이동하도록 하고 - 상기 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅됨 -;

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소를 나타내기 위하여 상기 제1 버퍼에 프록시 데이터 요소를 생성하도록 하고;

상기 제1 버퍼 내의 프록시 데이터 요소의 위치에 기초하여 상기 제2 버퍼 내에 저장된 데이터 요소 중 하나 이상을 상기 제1 버퍼에 이동하도록 하고 - 상기 제1 버퍼에 이동된 데이터 요소는 상기 미리 정해진 순서에 따라 상기 제1 버퍼 내에 저장됨 -;

상기 제1 버퍼에 저장된 데이터 요소에 대해 경계 조건을 적용하도록 하고;

상기 경계 조건의 적용 결과에 기초하여, 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하도록 하기 위한 명령을 포함하는,

컴퓨터 판독가능 저장 매체.

청구항 22

데이터 요소를 소팅하기 위한 컴퓨팅 시스템으로서,

제1 버퍼와 제2 버퍼 내의 데이터 요소들을 위한 데이터 저장 시스템; 및

상기 데이터 저장 시스템에 연결되고, 수신된 데이터 요소 스트림을 프로세싱하도록 구성되는 적어도 하나의 프로세서

를 포함하고,

상기 프로세싱은,

제1 버퍼의 오버플로우 상태 검출에 응답하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 상기 제2 버퍼에 이동시키는 과정 - 상기 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅됨 -;

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소를 나타내기 위하여 상기 제1 버퍼에 프록시 데이터 요소를 생성하는 과정;

상기 제1 버퍼 내의 프록시 데이터 요소의 위치에 기초하여 상기 제2 버퍼 내에 저장된 데이터 요소 중 하나 이상을 상기 제1 버퍼에 이동시키는 과정 - 상기 제1 버퍼에 이동된 데이터 요소는 상기 미리 정해진 순서에 따라 상기 제1 버퍼 내에 저장됨 -;

상기 제1 버퍼에 저장된 데이터 요소에 대해 경계 조건을 적용하는 과정;

상기 경계 조건의 적용 결과에 기초하여, 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 과정을 포함하는,

데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 23

데이터 요소를 소팅하기 위한 컴퓨팅 시스템으로서,

제1 버퍼와 제2 버퍼 내에 데이터 요소들을 저장하기 위한 수단; 및

수신된 데이터 요소 스트림을 프로세싱하기 위한 수단

을 포함하고,

상기 프로세싱은,

제1 버퍼의 오버플로우 상태 검출에 응답하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 상기 제2 버퍼에 이동시키는 과정 - 상기 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅됨 -;

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소를 나타내기 위하여 상기 제1 버퍼에 프록시 데이터 요소를 생성하는 과정;

상기 제1 버퍼 내의 프록시 데이터 요소의 위치에 기초하여 상기 제2 버퍼 내에 저장된 데이터 요소 중 하나 이상을 상기 제1 버퍼에 이동시키는 과정 - 상기 제1 버퍼에 이동된 데이터 요소는 상기 미리 정해진 순서에 따라 상기 제1 버퍼 내에 저장됨 -;

상기 제1 버퍼에 저장된 데이터 요소에 대해 경계 조건을 적용하는 과정;

상기 경계 조건의 적용 결과에 기초하여, 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 과정을 포함하는,

데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 24

제21항에 있어서,

상기 제1 버퍼 내의 데이터 요소의 상기 미리 정해진 순서는 상기 제1 버퍼로부터 상기 데이터 요소를 방출하기 위한 우선순위 순서를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 25

제21항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 미리 정해진 순서에 따라 실질적으로 소팅되는, 컴퓨터 판독가능 저장 매체.

청구항 26

제21항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 제1 버퍼에 저장된 데이터 요소의 적어도 절반을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 27

제21항에 있어서,

상기 컴퓨터 프로그램은, 컴퓨팅 시스템으로 하여금:

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소에 대응하는 상기 제1 버퍼의 위치에 데이터 요소를 저장하도록 하기 위한 명령을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 28

제21항에 있어서,

상기 컴퓨터 프로그램은, 컴퓨팅 시스템으로 하여금:

상기 미리 정해진 순서에 따라 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하도록 하고;

상기 제1 버퍼로부터 방출된 하나 이상의 데이터 요소를 상기 미리 정해진 순서에 따라 실질적으로 소팅된 데이터 요소의 출력 스트림으로 제공하도록 하기 위한 명령을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 29

제24항에 있어서,

상기 컴퓨터 프로그램은, 컴퓨팅 시스템으로 하여금:

상기 프록시 데이터 요소가 상기 제1 버퍼 내에서 가장 높은 우선순위의 데이터 요소가 되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하도록 하기 위한 명령을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 30

제21항에 있어서,

상기 컴퓨터 프로그램은, 컴퓨팅 시스템으로 하여금:

상기 프록시 데이터 요소가 상기 제1 버퍼로부터의 제거를 위해 식별되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하도록 하기 위한 명령을 더 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 31

제29항에 있어서,

병합된 데이터 요소가 상기 미리 정해진 순서에 기초하여 소팅되는, 컴퓨터 판독가능 저장 매체.

청구항 32

제29항 또는 제30항에 있어서,

병합이 출력 데이터 요소의 생성과 동시에 일어나는, 컴퓨터 판독가능 저장 매체.

청구항 33

제21항에 있어서,

상기 프록시 데이터 요소는 상기 제2 버퍼로부터의 제거를 위한 가장 높은 우선순위를 갖는 상기 제2 버퍼로부터의 데이터 요소의 복사본인, 컴퓨터 판독가능 저장 매체.

청구항 34

제28항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 작은 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 35

제28항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 큰 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 36

삭제

청구항 37

제21항에 있어서,

각각의 데이터 요소는 데이터 레코드 생성 시간을 나타내는, 컴퓨터 판독가능 저장 매체.

청구항 38

제21항에 있어서,

상기 경계 조건을 상기 제1 버퍼에 저장된 데이터 요소에 대해 적용하는 것은,

상기 제1 버퍼에 저장된 데이터 요소의 가장 작은 값을 나타내는 데이터 요소를 식별하는 것; 및

상기 가장 작은 값을 나타내는 데이터 요소와 상기 제1 버퍼에 입력하기 위해 수신된 데이터 요소 간의 차이를 판단하는 것을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 39

제21항에 있어서,

상기 경계 조건에 기초하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 것은,

하나 이상의 데이터 요소가 미리 정해진 값을 초과하는지를 판단하는 것; 및

상기 하나 이상의 데이터 요소가 미리 정해진 값을 초과하면, 상기 하나 이상의 데이터 요소를 상기 제1 버퍼로부터 방출하는 것을 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 40

제21항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 스킵 리스트 데이터 구조에 의해 구현되는, 컴퓨터 판독가능 저장 매체.

청구항 41

제21항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 트립 데이터 구조에 의해 구현되는, 컴퓨터 판독가능 저장 매

체.

청구항 42

제21항에 있어서,

상기 제1 버퍼 및 제2 버퍼 내의 개개의 데이터 요소의 하나 이상은 그래프 기반 연산 시스템의 컴포넌트로 입력되는 데이터 레코드로부터 도출되는, 컴퓨터 판독가능 저장 매체.

청구항 43

제22항에 있어서,

상기 제1 버퍼 내의 데이터 요소의 상기 미리 정해진 순서는 상기 제1 버퍼로부터 상기 데이터 요소를 방출하기 위한 우선순위 순서를 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 44

제22항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 미리 정해진 순서에 따라 실질적으로 소팅되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 45

제22항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 제1 버퍼에 저장된 데이터 요소의 적어도 절반을 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 46

제22항에 있어서,

상기 프로세싱은:

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소에 대응하는 상기 제1 버퍼의 위치에 데이터 요소를 저장하는 것을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 47

제22항에 있어서,

상기 프로세싱은:

상기 미리 정해진 순서에 따라 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 것; 및

상기 제1 버퍼로부터 방출된 하나 이상의 데이터 요소를 상기 미리 정해진 순서에 따라 실질적으로 소팅된 데이터 요소의 출력 스트림(output stream)으로 제공하는 것을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 48

제43항에 있어서,

상기 프로세싱은:

상기 프록시 데이터 요소가 상기 제1 버퍼 내에서 가장 높은 우선순위의 데이터 요소가 되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하는 것을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 49

제22항에 있어서,

상기 프로세싱은:

상기 프록시 데이터 요소가 상기 제1 버퍼로부터의 제거를 위해 식별되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하는 것을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 50

제48항에 있어서,

병합된 데이터 요소가 상기 미리 정해진 순서에 기초하여 소팅되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 51

제48항 또는 제49항에 있어서,

병합이 출력 데이터 요소의 생성과 동시에 일어나는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 52

제22항에 있어서,

상기 프록시 데이터 요소는 상기 제2 버퍼로부터의 제거를 위한 가장 높은 우선순위를 갖는 상기 제2 버퍼로부터의 데이터 요소의 복사본인, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 53

제47항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 작은 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 54

제47항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 큰 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 55

삭제

청구항 56

제22항에 있어서,

각각의 데이터 요소는 데이터 레코드 생성 시간을 나타내는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 57

제22항에 있어서,

상기 경계 조건을 상기 제1 버퍼에 저장된 데이터 요소에 대해 적용하는 것은,

상기 제1 버퍼에 저장된 데이터 요소의 가장 작은 값을 나타내는 데이터 요소를 식별하는 것; 및

상기 가장 작은 값을 나타내는 데이터 요소와 상기 제1 버퍼에 입력하기 위해 수신된 데이터 요소 간의 차이를 판단하는 것을 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 58

제22항에 있어서,

상기 경계 조건에 기초하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 것은,

하나 이상의 데이터 요소가 미리 정해진 값을 초과하는지를 판단하는 것; 및

상기 하나 이상의 데이터 요소가 미리 정해진 값을 초과하면, 상기 하나 이상의 데이터 요소를 상기 제1 버퍼로부터 방출하는 것을 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 59

제22항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 스킵 리스트 데이터 구조에 의해 구현되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 60

제22항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 트립 데이터 구조에 의해 구현되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 61

제22항에 있어서,

상기 제1 버퍼 및 제2 버퍼 내의 개개의 데이터 요소의 하나 이상은 그래프 기반 연산 시스템의 컴포넌트로 입력되는 데이터 레코드로부터 도출되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 62

제23항에 있어서,

상기 제1 버퍼 내의 데이터 요소의 상기 미리 정해진 순서는 상기 제1 버퍼로부터 상기 데이터 요소를 방출하기 위한 우선순위 순서를 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 63

제23항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 미리 정해진 순서에 따라 실질적으로 소팅되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 64

제23항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소는 상기 제1 버퍼에 저장된 데이터 요소의 적어도 절반을 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 65

제23항에 있어서,

상기 제2 버퍼에 이동된 하나 이상의 데이터 요소에 대응하는 상기 제1 버퍼의 위치에 데이터 요소를 저장하기 위한 수단을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 66

제23항에 있어서,

상기 미리 정해진 순서에 따라 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하기 위한 수단; 및

상기 제1 버퍼로부터 방출된 하나 이상의 데이터 요소를 상기 미리 정해진 순서에 따라 실질적으로 소팅된 데이터 요소의 출력 스트림(output stream)으로 제공하기 위한 수단을 더 포함하는, 데이터 요소를 소팅하기 위한

컴퓨팅 시스템.

청구항 67

제62항에 있어서,

상기 프록시 데이터 요소가 상기 제1 버퍼 내에서 가장 높은 우선순위의 데이터 요소가 되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하기 위한 수단을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 68

제23항에 있어서,

상기 프록시 데이터 요소가 상기 제1 버퍼로부터의 제거를 위해 식별되는 것에 응답하여, 상기 제2 버퍼로부터의 하나 이상의 데이터 요소를 상기 제1 버퍼 내의 데이터 요소와 병합하기 위한 수단을 더 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 69

제67항에 있어서,

병합된 데이터 요소가 상기 미리 정해진 순서에 기초하여 소팅되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 70

제67항 또는 제68항에 있어서,

병합이 출력 데이터 요소의 생성과 동시에 일어나는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 71

제23항에 있어서,

상기 프록시 데이터 요소는 상기 제2 버퍼로부터의 제거를 위한 가장 높은 우선순위를 갖는 상기 제2 버퍼로부터의 데이터 요소의 복사본인, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 72

제66항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 작은 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 73

제66항에 있어서,

상기 방출된 하나 이상의 데이터 요소는 상기 제1 버퍼 내의 데이터 요소 중 가장 큰 값을 나타내는, 상기 제1 버퍼로부터의 데이터 요소를 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 74

삭제

청구항 75

제23항에 있어서,

각각의 데이터 요소는 데이터 레코드 생성 시간을 나타내는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 76

제23항에 있어서,

상기 경계 조건을 상기 제1 버퍼에 저장된 데이터 요소에 대해 적용하는 것은,

상기 제1 버퍼에 저장된 데이터 요소의 가장 작은 값을 나타내는 데이터 요소를 식별하는 것; 및

상기 가장 작은 값을 나타내는 데이터 요소와 상기 제1 버퍼에 입력하기 위해 수신된 데이터 요소 간의 차이를 판단하는 것을 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 77

제23항에 있어서,

상기 경계 조건에 기초하여 상기 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 것은,

하나 이상의 데이터 요소가 미리 정해진 값을 초과하는지를 판단하는 것; 및

상기 하나 이상의 데이터 요소가 미리 정해진 값을 초과하면, 상기 하나 이상의 데이터 요소를 상기 제1 버퍼로부터 방출하는 것을 포함하는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 78

제23항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 스킵 리스트 데이터 구조에 의해 구현되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 79

제23항에 있어서,

상기 제1 버퍼와 상기 제2 버퍼 중 적어도 하나는 트립 데이터 구조에 의해 구현되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 80

제23항에 있어서,

상기 제1 버퍼 및 제2 버퍼 내의 개개의 데이터 요소의 하나 이상은 그래프 기반 연산 시스템의 컴포넌트로 입력되는 데이터 레코드로부터 도출되는, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템.

청구항 81

제1항에 있어서,

상기 경계 조건은 상기 제1 버퍼 내에 저장된 임의의 두 개의 데이터 요소의 값들 간의 최대 허용 차이를 나타내는 것인, 컴퓨터 구현 방법.

발명의 설명

기 술 분 야

[0001] 본 출원은 2011년 2월 18일자로 출원된 발명의 명칭이 "버퍼 상태 관리(Managing Buffer Condditions)"인 미국 출원 제13/031,034호에 대한 우선권의 이익을 주장하며, 상기 문헌의 내용은 원용에 의해 본 명세서에 포함된다.

[0002] 본 발명은 버퍼 상태 관리에 관한 것이다.

배 경 기 술

[0003] 여러 가지의 데이터 프로세싱 기술과 유사하게, 버퍼 관리 기술은 많은 어플리케이션에서 중요한 역할을 한다. 일 구현예에서, 버퍼 오버플로우 상태는 컴퓨터 시스템에 중대한 장애(disruption)를 초래할 수 있는데, 예를 들어, 메모리 용량을 초과하는 양의 데이터를 저장하려고 시도할 때, 프로세싱이 느려지거나 완전히 중단될 수

있다. 오버플로우를 막기 위해서, 데이터가 다른 목적에 대한 의무가 있는 다른 메모리 영역에 기록될 수 있으며, 이에 따라 잠재적으로 프로세싱의 둔화(slowdown) 또는 종료(termination)를 또한 초래할 수 있다.

발명의 내용

[0004] 일 측면에서, 일반적으로, 컴퓨터 구현 방법은 제1 버퍼의 오버플로우(overflow) 상태 검출에 응답하여 제1 버퍼에 있는 데이터 요소(data elements)의 일부분을 제2 버퍼에 저장하는 단계 및 제2 버퍼에 저장된 데이터 요소의 일부분을 나타내기 위하여 제1 버퍼에 프록시(proxy) 데이터 요소를 삽입하는 단계를 포함할 수 있고, 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅될 수 있다.

[0005] 이러한 본 발명의 측면은 다음의 특징 중 하나 이상을 포함할 수 있다.

[0006] 제1 버퍼 내의 데이터 요소의 미리 정해진 순서는 제1 버퍼로부터 데이터 요소를 제거하기 위한 우선순위 순서를 포함할 수 있다. 제2 버퍼에 저장된 데이터 요소의 일부분은 미리 정해진 순서에 따라 실질적으로 소팅될 수 있다. 제2 버퍼에 저장된 데이터 요소의 일부분은 제1 버퍼에 저장된 데이터 요소의 적어도 절반을 포함할 수 있다. 상기 방법은 제2 버퍼에 저장된 데이터 요소의 일부분에 대응하는 제1 버퍼의 위치에 데이터 요소를 저장하는 단계를 더 포함할 수 있다. 상기 방법은 미리 정해진 순서에 따라 제1 버퍼로부터 하나 이상의 데이터 요소를 제거하는 단계 및 제1 버퍼로부터 제거된 하나 이상의 데이터 요소를 미리 정해진 순서에 따라 실질적으로 소팅된 데이터 요소의 출력 스트림(output stream)으로 제공하는 단계를 더 포함할 수 있다. 상기 방법은 프록시 데이터 요소가 제1 버퍼에 저장된 데이터 요소 중 가장 높은 우선순위의 데이터 요소가 되는 것에 응답하여, 제2 버퍼로부터의 하나 이상의 데이터 요소를 제1 버퍼 내의 데이터 요소와 병합하는 단계를 더 포함할 수 있다. 상기 방법은 프록시 요소가 제1 버퍼로부터의 제거를 위해 식별되는 것에 응답하여, 제2 버퍼로부터의 하나 이상의 데이터 요소를 제1 버퍼 내의 데이터 요소와 병합하는 단계를 더 포함할 수 있다. 병합된 데이터 요소가 상기 미리 정해진 순서에 기초하여 소팅될 수 있다. 병합이 출력 데이터 요소의 생성과 실질적으로 동시에 일어날 수 있다. 프록시 데이터 요소는 제2 버퍼로부터의 제거를 위한 가장 높은 우선순위를 갖는 제2 버퍼로부터의 요소의 복사본일 수 있다. 제거된 하나 이상의 데이터 요소는 제1 버퍼 내의 데이터 요소 중 가장 작은 값을 나타낼 수 있다. 제거된 하나 이상의 데이터 요소는 제1 버퍼 내의 데이터 요소 중 가장 큰 값을 나타낼 수 있다. 상기 방법은 경계 조건을 제1 버퍼에 저장된 데이터 요소에 대해 적용하는 단계 및 경계 조건에 기초하여, 미리 정해진 우선순위 순서에 따라 소팅된 데이터 요소들의 출력 스트림으로서 하나 이상의 데이터 요소를 제1 버퍼로부터 방출하는 단계를 더 포함할 수 있다.

[0007] 각각의 데이터 요소는 데이터 레코드 생성 시간을 나타낼 수 있다. 경계 조건을 제1 버퍼에 저장된 데이터 요소에 대해 적용하는 단계는 제1 버퍼에 저장된 데이터 요소의 가장 작은 값을 나타내는 데이터 요소를 식별하는 단계 및 가장 작은 값을 나타내는 데이터 요소와 제1 버퍼에 입력하기 위해 수신된 데이터 요소 간의 차이를 판단하는 단계를 포함할 수 있다. 경계 조건에 기초하여 제1 버퍼로부터 하나 이상의 데이터 요소를 방출하는 단계는 하나 이상의 데이터 요소가 미리 정해진 값을 초과하는지를 판단하는 단계 및 하나 이상의 데이터 요소가 미리 정해진 값을 초과하면, 하나 이상의 데이터 요소를 제1 버퍼로부터 방출하는 단계를 포함할 수 있다. 제1 버퍼와 제2 버퍼 중 적어도 하나는 스킵 리스트 데이터 구조(skip list data structure)에 의해 구현될 수 있다. 제1 버퍼와 제2 버퍼 중 적어도 하나는 트립 데이터 구조(treap data structure)에 의해 구현될 수 있다. 제1 버퍼 및 제2 버퍼 내의 개개의 데이터 요소의 하나 이상은 그래프 기반 연산 시스템(graph-based computation system)의 컴포넌트로 입력되는 데이터 레코드로부터 도출될 수 있다.

[0008] 다른 측면에서, 일반적으로, 데이터 요소를 소팅하기 위한 컴퓨터 프로그램을 저장하는 컴퓨터 판독가능 저장 매체는 컴퓨팅 시스템으로 하여금, 제1 버퍼의 오버플로우 상태 검출에 응답하여 제1 버퍼에 있는 데이터 요소의 일부분을 제2 버퍼에 저장하도록 하고 제2 버퍼에 저장된 데이터 요소의 일부분을 나타내기 위하여 제1 버퍼에 프록시 데이터 요소를 삽입하도록 하기 위한 명령을 포함할 수 있으며, 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅될 수 있다.

[0009] 다른 측면에서, 일반적으로, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템은 제1 버퍼에 있는 데이터 요소의 일부분을 제2 버퍼에 저장하기 위한 데이터 저장 시스템 및 데이터 저장 시스템에 연결되고, 수신된 데이터 요소 스트림을 프로세싱하도록 구성되는 적어도 하나의 프로세서를 포함할 수 있고 상기 프로세싱은 제1 버퍼의 오버플로우 상태 검출에 응답하여 제1 버퍼에 있는 데이터 요소의 일부분을 제2 버퍼에 저장하는 과정 및 제2 버퍼에 저장된 데이터 요소의 일부분을 나타내기 위하여 제1 버퍼에 프록시(proxy) 데이터 요소를 삽입하는 과정을 포함할 수 있으며, 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅될 수 있다.

[0010] 다른 측면에서, 일반적으로, 데이터 요소를 소팅하기 위한 컴퓨팅 시스템은 제1 버퍼에 있는 데이터 요소의 일부분을 제2 버퍼에 저장하기 위한 수단 및 수신된 데이터 요소 스트림을 프로세싱하기 위한 수단을 포함할 수 있고, 상기 프로세싱은 제1 버퍼의 오버플로우 상태 검출에 응답하여 제1 버퍼에 있는 데이터 요소의 일부분을 제2 버퍼에 저장하는 과정 및 제2 버퍼에 저장된 데이터 요소의 일부분을 나타내기 위하여 제1 버퍼에 프록시(proxy) 데이터 요소를 삽입하는 과정을 포함할 수 있으며, 제1 버퍼 내의 데이터 요소는 미리 정해진 순서에 따라 소팅될 수 있다.

[0011] 이와 같은 본 발명은 하나 이상의 다음의 이점을 포함한다.

[0012] 몇몇 예에서, 제1 버퍼 및 그 내용은 고속 메모리, 예컨대, 랜덤 액세스 메모리(RAM)에 저장될 수 있다. 동작하는 동안, 주 버퍼의 특정 부분을 저속 메모리(예컨대, 하드 디스크)에 저장하는 것이 더 바람직할 수 있다. 예를 들어, 주 버퍼 내의 특정 요소가 즉시 필요하지 않을 수 있고, 따라서 저속 메모리에 저장될 수 있는 보조 버퍼에서 제거될 수 있다. 하나 이상의 보조 버퍼가 즉시 필요하지 않은 요소들을 임시로 저장하기 위해서 채택될 수 있다. 주 버퍼에 있는 실질적으로 동일한 수의 요소가 더 이상 존재하지 않고 이에 따라 요소를 위한 적절한 공간을 제공하게 되면, 이러한 요소는 주 버퍼로 복원될 수 있다.

[0013] 또한, 대부분 소팅되어 있는, 수신된 데이터 요소들의 스트림을 수반하는 몇 가지 구현예에서, 요소들은 그 요소들의 대다수에 의한 적절한 순서로 되어 있지 않은 소수의 요소를 제외하고는, 대부분이 소팅 순서로 되어 있을 수 있다. 스트림의 각각의 개별 데이터 요소를 수신한 후에 경계 조건을 주 버퍼에 저장된 데이터 요소들에 적용함으로써, 미리 정해진 순서에 따라 소팅된, 데이터 요소의 출력 스트림이 생성될 수 있다.

[0014] 본 발명의 다른 특징 및 이점이 다음의 설명과 청구항으로부터 명백해질 것이다.

도면의 간단한 설명

[0015] 도 1은 버퍼 관리 시스템의 블록도이다.

도 2는 일련의 관리 동작 동안의 버퍼를 도시한 도면이다.

도 3은 버퍼 관리 프로세스의 순서흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0016] 도 1을 참조하면, 예시적인 버퍼 관리 시스템(buffer management system)(100)은 데이터 요소들(data elements)(106a-106n)(전체로서 106)을 저장 장치(108)에 저장하는 컴퓨터 시스템(104)(예컨대, 하나 이상의 클라이언트 시스템을 서비스하는 웹 서버와 같은 서버) 상에서 실행되는 어플리케이션(102)(예컨대, 소팅(sorting) 어플리케이션)을 포함한다. 저장 장치(108)는, 예를 들어, 하드 드라이브 메모리(hard drive memory) 또는 다수의 버퍼(두 개의 버퍼(110, 112)로 도시되어 있음)를 갖는 다른 이동식 저장 매체(removable storage media)일 수 있고, 컴퓨터 시스템(104)의 일부분으로 또는 분리된 독립형 저장 장치로 구현될 수 있다. 몇 가지 예에서, 제1 버퍼(110)는 제1 메모리, 예컨대, 랜덤 액세스 메모리(random access memory: RAM)에 위치할 수 있고, 제2 버퍼(112)는 하드 디스크에 위치할 수 있다.

[0017] 요소(106)들은, 예를 들어, 소팅 어플리케이션(102)에 의해 생성된 데이터 레코드(도시하지 않음)의 하나 이상의 속성(attribute)에 할당된 값과 같은, 여러 가지 유형의 정보를 나타낼 수 있다. 몇 가지 구현예에서, 요소(106)들은 하나 이상의 데이터 레코드에 대해 수행된 연산(computation)으로부터 기인하는 값을 나타낼 수 있다. 요소(106)에 의해 나타낼 수 있는 다른 정보는 타임스탬프(timestamp), 예컨대 대응하는 데이터 레코드가 소팅 어플리케이션(102)에 의해 생성된 각각의 시간을 포함할 수 있다.

[0018] 소팅 어플리케이션(102)이 여러 가지의 상이한 기능을 제공할 수 있지만, 일 예에서는, 소팅 어플리케이션(102)이 심사중인 미국 특허 출원 제13/031,056호(이 문헌의 내용이 원용에 의해 본 명세서에 포함됨)에서 상세하게 설명되는 윈도우 소팅 어플리케이션(window sorting application)일 수 있다. 몇 가지 예에서, 소팅 어플리케이션(102)이 데이터 요소(106)를 만들어낼 수 없지만, 소팅을 위한 제2의 상이한 어플리케이션(114)(예를 들어, 컴퓨터 시스템(116) 상에서 실행됨)으로부터 데이터 요소를 선택적으로 수신할 수 있다.

[0019] 본 명세서에서 소팅은 수신된 스트림의 요소(106)들을 특정 순서로 나열하는 프로세스를 나타낸다. 예를 들어, 소팅 어플리케이션(102)은 요소를, 그 숫자 값(numeric value)의 오름 차순으로 출력할 수 있다. 동작(operation)에 있어서, 소팅 어플리케이션(102)은 소팅되지 않은 요소(예컨대, 데이터 소스 어플리케이션(114)

으로부터 도착하는 요소들)의 입력 스트림(incoming stream)을 주 버퍼(primary buffer)라 불리는 버퍼(110)에 저장한다. 요소들은 소팅된 요소(118a, 118b)(전체로서 118)와 같이 실질적으로 소팅된 방식으로 저장될 수 있다. 다만, 입력 스트림에 비교적 많은 요소(예컨대, 백만개)가 있을 수 있는데, 이러한 입력 스트림은 명백하게 주 버퍼(110)로 하여금 제공되는 요소 모두를 저장할 수 없도록 할 수 있다. 예를 들어, 주 버퍼(110)는 그것의 최대 물리적 메모리 사이즈에 기초한 용량을 채울 수 있다. 이러한 상황의 결과로서, 주 버퍼(110)는 오버플로우(overflow)할 수 있고, 컴퓨터 시스템(104)의 동작에 장애가 발생하거나 동작이 잠재적으로 중단되도록 할 수 있다. 이러한 문제점을 해결하기 위하여, 하나 이상의 "스필 오버 베이슨(spill-over basin)" 또는 버퍼(112)와 같은 보조 버퍼(secondary buffer)가 채택되어, 도 1에 나타낸 요소(120a, 120b)(전체로서 120)와 같은 오버플로우하는 요소들을 임시로 저장할 수 있다.

[0020] 몇 가지 예에서, 주 버퍼 및 그 내용(contents)은 RAM과 같은 고속 메모리에 저장될 수 있다. 동작하는 동안, 주 버퍼의 특정 부분을 저속 메모리(예컨대, 하드 디스크)에 저장하는 것이 더 바람직할 수 있다. 예를 들어, 주 버퍼 내의 특정 요소가 즉시 필요하지 않을 수 있고, 따라서 저속 메모리에 저장될 수 있는 보조 버퍼에서 제거될 수 있다. 이러한 상황을 해결하기 위하여, 버퍼(112)와 같은 하나 이상의 보조 버퍼가, 요소(120a, 120b)와 같이 즉시 필요하지 않은 요소들을 임시로 저장하기 위해서 채택될 수 있다.

[0021] 주 버퍼(110)에 있는 실질적으로 동일한 수의 요소가 더 이상 존재하지 않고(예컨대, 방출된(ejected) 요소(122a, 122b)) 이에 따라 요소(120)를 위한 적절한 공간을 제공하게 되면, 이러한 요소(120)는 주 버퍼(110)로 복원(예컨대, 주 버퍼 내로 다시 병합됨)될 수 있다.

[0022] 소팅된 출력을 제공하기 위하여, 소팅 어플리케이션(102)은 방출된 요소(122)들이 실질적으로 소팅된 순서로 되는 것을 보장하기 위한 동작을 수행한다. 예를 들어, 더 작은(원하는 출력에 따라, 또는 더 큰) 값의 요소가 먼저 방출되고 그 다음 더 큰(더 작은) 값의 요소가 방출되도록, 요소(122)들이 방출된다. 요소가 타임스탬프 값을 나타내는 예에서, 가장 최근의(원하는 출력에 따라, 또는 가장 오래된) 타임스탬프와 관련된 요소가 버퍼에서 방출될 수 있다. 방출된 요소(122)는, 도시된 바와 같이, 다른 컴퓨터 시스템(126)에 의해 실행되는 수신 어플리케이션(receiving application)(124)에 대한 추가 프로세싱을 위해 제공될 수 있다. 수신 어플리케이션(124)은 방출된 요소(122)를 수신하고 프로세싱할 수 있는 소프트웨어 어플리케이션의 대표적인 유형이다.

[0023] 도 2는 버퍼 오버플로우 관리를 위한 동작을 나타내는, 한 쌍의 버퍼(210, 212)(예컨대, 도 1에 도시한 주 버퍼(110)와 보조 버퍼(112))의 상태(202-208)를 도식적으로 나타낸 것이다. 입력 요소의 순서는, 예컨대 완전히 랜덤한 순서이거나 또는 어느 정도 결정되어(deterministic) 있는 것(예를 들어, 부분적으로 소팅되거나, 대부분 소팅된 경우 등)과 같이, 여러 가지의 순서로 제공될 수 있다. 대부분 소팅된 시나리오에서, 소수의 아웃라이어(outlier)(대다수(majority)의 요소와 비교하여)를 제외하고는, 요소들의 대부분이 소팅된 순서로 되어 있을 수 있다. 이러한 아웃라이어를 수량화하자면, 비교적 많은 요소(예컨대, 백만개)를 갖는 스트림에 대하여, 요소의 약 0.1-10%가 순서대로 되어 있지 않을 수 있다. 이러한 아웃라이어 요소들을 많은 요소를 갖는 스트림 내의 적절한 위치로 효율적으로 삽입하기 위해서, 소팅 어플리케이션은 요소의 대다수가 적절한 순서로 도착한다는 사실을 이용할 수 있다. 예를 들어, 아웃라이어를 포함하는 입력 요소들은 소팅된 방식으로 주 버퍼(210)에 저장될 수 있다(예컨대, 소팅 어플리케이션(102)에 의해). 도착한 요소들의 대다수가 이미 소팅되어 있기 때문에, 버퍼(210)에 요소들을 저장하는 것은 비교적 효율적인 프로세스이다. 그러나, 아웃라이어가 버퍼에 도착하여 주 버퍼(210) 내의 적절한 위치에 삽입될 필요가 있는 경우, 추가 시간이 필요할 수 있다.

[0024] 일반적으로, 저장된 요소들은 하나 이상의 규칙에 기초하여 주 버퍼(210)로부터 방출된다. 예를 들어, 각각의 저장된 요소와 관련된 값(예를 들어, 숫자 값)은 대응하는 요소가 방출되어야 하는지를 판단하는 데 사용될 수 있다. 요소들이 저장되는 순서 또한, 하나 이상의 요소를 주 버퍼(210)로부터 방출할지 여부에 대한 판단의 한 요인이 될 수 있다. 적절한 시간에, 소팅 어플리케이션(102)은 소팅된 순서(예를 들어, 오름 차순, 내림 차순(descending order) 등)에 따라 우선순위를 매기는 방식으로 요소들을 주 버퍼(210)로부터 방출한다. 또한, 저장된 요소들에 부과된 조건(예컨대, 임의의 두 개의 저장된 요소 간의 허용가능한 최대 차이)이 위반되는 경우에 요소가 방출될 수 있다. 주 버퍼(210)로부터 방출되면, 그 요소들은 소팅된, 요소들의 출력 스트림을 형성한다.

[0025] 요소들이 숫자 값의 오름 차순에 따라 소팅되면, 더 작은 요소들이 먼저 방출되고 더 큰 값의 요소들이 나중에 방출될 수 있다. 유사하게, 내림 차순에 따라 소팅된 요소들의 경우, 더 큰 요소들이 먼저 방출되고 더 작은 값의 요소들이 후에 방출될 수 있다. 각 요소와 관련된 이러한 숫자 값은 하나 이상의 양(quantity)을 나타낼 수 있다.

[0026] 버퍼(210, 212)는 데이터 요소들을 저장하고 그 요소들을 특정의 소팅된 순서로 방출하기 위한 하나 이상의 데이터 구조(data structure) 및 데이터 아키텍처(data architecture)에 의해 구현될 수 있다(예컨대, 우선순위 큐 데이터 구조(priority queue data structure)). 일 구현예에서, 소팅 어플리케이션(102)은 데이터 구조를 이용하여 데이터 요소들을 소팅하기 위한 여러 가지의 기능을 제공할 수 있다. 예를 들어, 어떤 방식에서, 주 버퍼와 보조 버퍼를 규정하기 위하여 선택된 데이터 구조에 대해 세 개의 기능이 주로 사용될 수 있다.

[0027] 상술한 세 개의 기능 중 하나에 있어서, 소팅 어플리케이션(102)은, 데이터 구조에 저장된 요소들의 주된 소팅 순서에 의해 판단되는 바와 같이, 요소를 데이터 구조의 적절한 위치 내에 삽입할 수 있다. 예를 들어, 숫자 값의 오름 차순으로 배열되어 있는, 저장된 요소들을 고려한다. 소팅 어플리케이션(102)은 입력 요소의 숫자 값에 기초하여 데이터 구조 내의 적절한 위치에 새로운 입력 요소를 삽입할 수 있다. 두 번째 기능으로서, 소팅 어플리케이션은, 방출 우선순위 순서(예를 들어, 버퍼 내의 가장 작은 또는 가장 큰 데이터 요소를 방출함)에 기초하여 데이터 구조로부터 방출될 하나 이상의 요소를 식별할 수 있다. 세 번째로, 소팅 어플리케이션은 식별된 요소 또는 데이터 구조에 있는 요소들의 방출을 시작하는 기능을 제공할 수 있다.

[0028] 어떤 방식에서, 주 버퍼(210)와 보조 버퍼(212)는 각각 균형 이진 트리 데이터 구조(balanced binary tree data structure)(예컨대, 힙 이진 트리 데이터 구조(heap binary tree data structure))일 수 있다. 일반적으로, 힙 이진 트리 데이터 구조(보통 힙으로만 불림)는 두 가지 조건을 만족한다. 하나의 조건은 트리 구조에 포함된 자식 노드(children node)의 우선순위가 적어도 부모 노드의 우선순위만큼 크다는 것일 수 있다. 두 번째 조건은 힙이 "완전 트리(complete tree)", 즉 모든 행(row)이 채워진 트리일 수 있다는 것이며, 이는 바닥 행(bottom row)를 제외하고는, 리프 노드(leaf node) 간에 갭(gap)(즉, 채워지지 않은 노드)이 없다는 것을 나타낸다.

[0029] 몇몇 예에서, 버퍼(210, 212)는 스킵 리스트 데이터 구조(skip list data structure)에 의해 구현될 수 있다. 일반적으로, 스킵 리스트 데이터 구조 및 그와 관련된 알고리즘은 연결 선형 리스트(linked linear list)의 변형된 형태이고 키잉된(keyed) 데이터 요소의 빈번한 삽입을 요구하는 상황에서 소팅 성능을 향상시킬 수 있다. 스킵 리스트 데이터 구조는 데이터 요소(또는 노드)의 정렬된, 연결 선형 리스트를 포함할 수 있고, 몇몇 요소는 중간(intermediate) 데이터 요소를 스킵하는 추가 포인터(pointer)를 가짐으로써 데이터 요소 탐색의 속도 및 효율을 증가시킬 수 있다.

[0030] 버퍼(210, 212)를 스킵 리스트 데이터 구조로 구현함으로써, 가장 작은 데이터 요소를 검색하기 위하여 검색 시간은 O(1) 시간이 될 수 있다. 데이터 요소를 적절하게 삽입하는 데 O(1) 시간의 삽입 시간이 필요할 수 있다(예컨대, 요소가 가장 최근에 삽입된 요소에 인접하게 삽입되는 경우). 삽입 위치가 가장 최근에 삽입된 요소에 인접하지 않으면, 삽입 시간은 O(log N) 시간이 될 수 있다. "k" 요소(예컨대, 가장 큰 값을 갖는 요소들)를 판독하기 위하여 방출 시간이 O(k) 시간이 될 수 있다. 스킵 리스트 데이터 구조는 Communications of the ACM. 1990년 6월, 668쪽에서 676쪽의, William Pugh의 "스킵 리스트: 평형 트리에 대한 확률적 대안(Skip lists: A probabilistic alternative to Balanced trees)"이라는 제목의 논문에 더욱 상세하게 설명되어 있으며, 상기 논문의 내용은 원용에 의해 본 명세서에 포함된다.

[0031] 다른 기술 또한 사용될 수 있는데, 예를 들어, 버퍼(210, 212)는 트립(treap) 데이터 구조에 의해 구현될 수 있다. 트립 데이터 구조는 키(key) 뿐만 아니라 각 노드가 랜덤하게 할당된 우선순위 속성을 갖는 이진 탐색 트리(binary search tree)이다. 노드들은 일반적인 이진 탐색 트리에서와 같이 그들의 키와 관련하여 순서대로(in order) 되어 있다. 즉, 키와 관련하여 노드의 좌측 서브트리(subtree)가 상기 노드의 키보다 작은 키를 갖는 노드들만을 포함하는 반면, 상기 노드의 우측 서브트리는 상기 노드의 키보다 크거나 같은 키를 갖는 노드들만을 포함한다. 또한, 노드들은 그들의 우선순위 속성과 관련하여 "힙 순서"로 되어 있어서, 각각의 자식 노드가 적어도 부모 노드만큼 큰 우선순위 속성을 갖는다. 트립 데이터 구조는 Proc. 30th Symp. Foundations of Computer Science (FOCS 1989), Aragon, Cecilia R., 및 Seidel, Raimund의, "랜덤화된 탐색 트리(Randomized Search Trees)"라는 제목의 논문에 더욱 상세하게 설명되어 있으며, 상기 논문의 내용은 원용에 의해 본 명세서에 포함된다.

[0032] 일 구현예에서, 소팅 어플리케이션(102)은, 주 버퍼로부터 어떤 요소가 언제 방출될 것인지를 판단하는 데 사용되는, 경계 조건(boundary condition)으로 하여금 주 버퍼(210)에 대해 적용되도록 할 수 있다. 경계 조건은 주 버퍼에 저장되도록 허용된 요소 값의 "윈도우(window)"의 형태를 취할 수 있다. 윈도우는 주 버퍼에 저장된 임의의 두 개의 데이터 요소 간의 최대 허용 차이(maximum allowable difference)를 나타낼 수 있다. 이에 따라, 윈도우의 "폭(width)"은 주 버퍼 내에서 허용가능한 값의 범위를 나타낸다. 예를 들어, 윈도우 폭은 주 버

퍼(210) 내의 가장 작은 요소와 가장 큰 요소 간의 차이를 산출함으로써 정해질 수 있다.

[0033] 경계 조건은, 그 조건이 충족되었는지 또는 충족되지 않았는지(예컨대, 위반)로 고려될 수 있는 점에서, 이진법으로(binary) 고려될 수 있다. 예를 들어, 경계 조건은 어떠한 입력 요소도 윈도우 폭이 초과되도록 하지 않는 한 충족된 것으로 간주될 수 있다. 반대로, 윈도우 폭이 초과되면, 경계 조건이 위반된 것으로 간주될 수 있다. 경계 조건이 위반된 상태에서 하나 이상의 동작이 실행될 수 있는데, 예를 들어, 주 버퍼(210) 내의 하나 이상의 요소가 방출 우선순위 순서에 따라 방출될 수 있다.

[0034] 윈도우와 관련된 파라미터들을 정하기 위하여 여러 가지의 기술이 사용될 수 있으며, 예를 들어, 윈도우 폭은 사용자에 의해서 특정되거나 미리 정해질 수 있다. 예를 들어, 특정 어플리케이션(예컨대, 전화 통화 종료 시간에 기초한 소팅)에 기초하여 윈도우 폭이 정해질 수 있다. 몇 가지 상황(context)에서, 윈도우 폭은 "out-of-orderness"의 기준(measure), 예컨대, 아웃라이어 요소가 허용될 수 있는 기준에 대응한다. 예를 들어, 타임스탬핑된 데이터 요소를 포함하는 상황을 고려한다. 윈도우 폭은 주 버퍼 내의 현재 타임스탬핑된 요소(예컨대, 가장 작은 또는 가장 큰 타임스탬핑된 요소)와 관련하여, 타임스탬핑된 요소가 주 버퍼에 얼마나 늦게 제공될 수 있는지를 나타낸다. 어떤 방식에서, 윈도우 폭(시간의 단위)이 4시간이면, 주 버퍼(210)는 4 시간 윈도우(주 버퍼 내의 가장 작은 타임스탬핑된 요소에 의해 정해짐) 내의 모든 타임스탬핑된 요소를 저장할 수 있다. 4시간을 벗어나서 도착하는 타임스탬핑된 요소는 폐기되거나, 별도로 처리될 수 있다.

[0035] 도 2에 나타낸 바와 같이, 주 버퍼(210)의 개개의 내용, 주 버퍼(210)의 윈도우 폭(라벨 W로 참조됨), 주 버퍼(210) 내의 현재 가장 작은 요소(라벨 S로 참조됨), 현재 출력 데이터 요소에 관한 정보가 각 상태(202-208)에 대해 제공된다. 이러한 예에서, 요소들은 오름 차순으로 저장되고, 입력 스트림은 오름 차순으로 대부분 소팅된 것으로 보인다. 소팅에 기초하여, 실질적으로 오름 차순으로 소팅된 출력 스트림이 제공된다. 설명을 위하여, 상기 예는 약 13개의 요소의 스트림에 대한 동작을 도시하지만, 프로세스는 더 많거나 더 적은 요소들을 갖는 스트림에 대해 동작할 수 있으며, 예를 들어, 요소들의 연속 스트림이 입력될 수 있고 프로세스에 의해 소팅될 수 있다.

[0036] 초기 상태(202)에서, 윈도우 폭 값이 20(예컨대, 주 버퍼(210) 내의 가장 작은 요소와 가장 큰 요소 간의 차이가 20 이하임)으로 미리 결정된다. 도시된 바와 같이, 주 버퍼(210)가 채워져 있고, 따라서 더 이상의 요소들을 저장할 수 없다. 주 버퍼(210) 내에서 현재 가장 작은 값을 가진 요소는 0의 값을 갖는다(또한 상태(202)에서 S=0으로 나타냄). 어떠한 요소도 방출되지 않았으므로, 현재의 출력은 "NIL", 즉, 출력 데이터 요소가 없는 것으로 나타난다. 또한, 보조 버퍼(212)는 현재 비어 있다.

[0037] 상태(204)에서, 16의 값을 갖는 다음의 입력 요소가 주 버퍼(210)에 삽입된다. 그러나, 값이 16인 요소에 대한 적절한 위치가 값이 17인 요소와 값이 14인 요소 사이에 있지만, 주 버퍼(210)는 현재, 값이 16인 요소를 위한 공간을 만들기 위해서 값이 17인 요소가 이동될 수 있는, 값이 16인 요소를 삽입하기 위해 필요한 공간이 없다. 공간을 만들기 위해서, 주 버퍼(210)에 저장된 요소들의 미리 결정된 일부분(예컨대, 절반(half))이 제거되어 보조 버퍼(212)에 제공될 수 있고, 값이 16인 요소는 주 버퍼(210) 내의 새로운 이용가능한 위치에 저장될 수 있다. 예를 들어, 저장된 요소들의 더 큰 절반(즉, 17, 14, 11, 10, 9의 값을 갖는 요소들)이 보조 버퍼(212)로 이동될 수 있다. 몇 가지 예에서, 미리 결정된 일부분은 미리 정해진 규칙에 기초하여 선택된다.

[0038] 저장된 요소들의 더 큰 절반을 제거하는 것의 하나의 이점은 요소들을 주 버퍼(210) 내로 다시 병합하기 위해 필요한 시간을 최대화하는 것이다. 요소들을 제거하고 보조 버퍼(212)에 기록하기 위해 필요한 시간은 약 O(k) 시간이고, 여기서 k는 제거된 요소들의 개수이다. 제거된 요소들은 또한 소팅 순서로 보조 버퍼(212)에 저장될 수 있다. 또한, 프록시 요소들(proxy element)(예컨대, 주 버퍼 내에 S9로 표시됨)가 주 버퍼(210) 내의 적절한 위치에 삽입될 수 있다. 프록시 요소들은, 제거되어 보조 버퍼(212)에 저장된 요소들을 (주 버퍼(210) 내에) 나타내기 위하여 사용된다. 프록시 요소들은 여러 가지 방식으로 나타내어질 수 있는데, 예를 들어, 프록시 요소를 나타내는 데이터 요소에 첨부된 부울 필드(Boolean field)(플래그(flag))를 변경함으로써 나타내어질 수 있다. 프록시 요소의 위치는 주 버퍼(210)에 남아 있는 요소들의 주된 순서에 의해 설정될 수 있다. 예를 들어, 이러한 설명에서, 가장 작은 값의 제거된 요소가 9의 값을 가지므로, 프록시 요소(S9)는 새롭게 삽입된 값이 16인 요소와 값이 8인 요소 사이에 삽입될 수 있다. 상태(204)에 대하여, 주 버퍼(210) 내의 가장 작은 요소와 가장 큰 요소 간의 차이가 16이고, 이러한 차이가 윈도우 폭, 20보다 여전히 작기 때문에, 경계 조건이 위반된 것으로 고려되지 않고 따라서 어떠한 데이터 요소도 방출될 필요가 없다.

[0039] 상태(206)에서, 도시된 바와 같이, 요소들을 저장하기 위해서 수개의 위치가 주 버퍼(210) 내에서 이용가능하다. 값이 20인 다음의 입력 요소가 주 버퍼 내의 적절한 위치, 즉, 값이 16인 요소의 다음 위치로

삽입된다. 이러한 삽입에 기초하여, 주 버퍼(210) 내의 가장 작은 요소와 가장 큰 요소 간의 차이가 20이 되는데, 이러한 차이는 윈도우 폭, 20과 동일하고, 따라서 경계 조건을 위반하게 된다. 이에 따라, 경계 조건이 위반되는 동안은, 요소들이 그들의 방출 우선순위의 순서(예컨대, 오름 차순)로 주 버퍼(210)로부터 방출된다. 값이 0인 주 버퍼(210) 내의 가장 작은 요소가 방출된다. 이제 다음으로 가장 작은 요소는 값이 1이다. 이 경우에 주 버퍼(210) 내의 가장 작은 요소와 가장 큰 요소 간의 차이가 19이므로, 윈도우 폭, 20보다 작게 되어 경계 조건이 위반되지 않는다. 따라서, 더 이상의 요소가 주 버퍼(210)로부터 방출될 필요가 없다.

[0040] 상태(208)에서, 값이 29인 요소가 주 버퍼(210) 내로 삽입되면, 가장 작은 요소와 가장 큰 요소 간의 차이가 28(즉, 29-1)이 되어서 윈도우 폭, 20보다 더 커진다. 경계 조건이 위반되었으므로 값이 1, 6, 7, 그리고 8인 요소들이 주 버퍼(210)로부터 방출된다. 29에서 9를 빼면 20이므로, 프록시 요소(S9) 또한 이제 경계 조건을 위반하게 된다.

[0041] 이러한 상황에서, 보조 버퍼(212)로부터의 요소들이 주 버퍼(210) 내의 요소들과 다시 병합됨으로써 결합된 요소들이 순서대로 저장될 수 있도록 한다. 독자의 편의를 위하여, 주 버퍼(210)에 새롭게 병합된 요소들은 밑줄 그은 굵은 폰트로 도시되어 있다. 일반적으로, 스킵 리스트 데이터 구조에 의해 구현되는 주 버퍼(210)의 경우, 요소들을 주 버퍼(210) 내로 다시 병합하는데 필요한 시간은 약 O(k log(n/k)) 시간이고, 여기서 "n"은 주 버퍼(210) 내의 요소들의 개수이고, "k"는 병합되는 요소들의 개수이다. 병합이 완료된 후에, 값이 9인 요소는 버퍼로부터 방출될 수 있는데, 왜냐하면 이 요소가 경계 조건이 위반되도록 했기 때문이다(29-9=20). 따라서, 경계 조건을 위반하지 않는, 버퍼 내의 가장 작은 요소는 이제 값이 10인 요소이다. 결국, 더 이상의 요소가 주 버퍼로부터 방출되지 않는다.

[0042] 일반적으로, 병합이 발생하면, 주 버퍼로부터 방출되는 요소의 개수는 보조 버퍼에 있는 요소의 개수와 동일하다. 예를 들어, 보조 버퍼로부터의 네 개의 요소를 위한 공간을 제공하기 위하여 네 개의 요소가 주 버퍼로부터 방출된다(병합 후에 값이 9인 다섯 번째 요소가 방출됨).

[0043] 예컨대 개수와 같은, 버퍼 구조의 변형은 몇 가지 구현예에서 달성될 수 있다. 예를 들어, 하나의 주 버퍼와 둘 이상의 보조 버퍼가 사용될 수 있다. 다수의 버퍼와 유사하게, 요소들을 저장하는 다수의 보조 버퍼에 각각 대응하여 다수의 프록시 요소들이 주 버퍼에 저장될 수 있다. 프록시 요소가 주 버퍼 내에서 가장 높은 방출 우선순위 요소가 되면, 대응하는 보조 버퍼로부터의 요소가 주 버퍼에 병합된다.

[0044] 몇 가지 예에서, 보조 버퍼는 용량이 다 채워질 수 있고, 다른 상이한 버퍼(예컨대, 제3의 버퍼)가 오버플로우 요소를 저장하는 데 사용될 수 있다. 제3의 버퍼에 저장된 오버플로우 요소를 나타내기 위하여 프록시 요소가 보조 버퍼 내의 적절한 위치에 삽입될 수 있다.

[0045] 보조 버퍼들의 길이(그리고 따라서 이러한 버퍼들에 대응하는 파일 크기)는, 오버플로우 상태 발생 시에 주 버퍼로부터 제거된 요소들의 수에 의존한다. 예를 들어, 최적의 파일 크기는, 주 버퍼의 효율적인 이용을 위해 허용되는 만큼의 요소들을(예를 들어, 요소들의 개수의 적어도 절반까지) 주 버퍼로부터 제거하는 것을 수반한다. 추가적으로 고려하여야 할 사항은 제거 및 병합 동작의 수를 최소화하는 것이다.

[0046] 도 3을 참조하면, 버퍼의 오버플로우 상태를 관리하기 위한 예시적인 버퍼 오버플로 관리 프로세스(300)의 동작을 나타내는 순서흐름도가 도시된다. 구현예에서, 이하에서 설명되는 단계들은 컴퓨터 시스템상에서 실행되는 소프트웨어 어플리케이션에 의해 수행될 수 있다. 예비 문제(preliminary matter)로서, 주 버퍼 및 보조 버퍼는 하나 이상의 기술에 의해 초기화된다(단계 302). 예를 들어, 초기화는 입력 요소들을 저장하기 위한 메모리 영역, 예컨대, 복수의 인접한 메모리 주소들을 특정하는 과정을 포함할 수 있다. 초기화의 일부로서, 경계 조건이 주 버퍼에 대해 특정될 수 있다. 예를 들어, 주 버퍼에 대한 윈도우 폭에 대응하는 값을 유지하기 위하여 변수가 정해질 수 있다. 윈도우 폭은 주 버퍼에 저장된 임의의 두 개의 데이터 요소 간의 최대 허용 차이를 나타낼 수 있다.

[0047] 주 버퍼에 대한 경계 조건이 정해진 후에, 주 버퍼는 데이터 요소들의 입력 스트림을 수신할 준비가 되어 있다(단계 304). 그러나, 주 버퍼의 용량이 다 차면, 오버플로우 상태가 발생한다(단계 306). 더 이상의 요소가 주 버퍼에 수용될 수 없으면 오버플로우 상태가 발생한다. 이러한 상황에서, 주 버퍼에 저장된 요소들 중 일부분(예컨대, 절반)이 제거되어 보조 버퍼에 저장될 수 있다(단계 308). 보조 버퍼(예컨대, 제2 버퍼)에 저장된 데이터 요소의 일부분을 나타내기 위하여 프록시 데이터 요소를 주 버퍼(예컨대, 제1 버퍼)에 삽입하는 데 하나 이상의 기술이 채택될 수 있다. 예를 들어, 프록시 요소는 제거된 요소들을 나타내기 위하여 주 버퍼 내의 적절한 위치에 삽입될 수 있는 센티넬(sentinel)에 의해 제공될 수 있다. 보조 버퍼에 저장된 요소가 소팅 순서

로 주 버퍼 내로 병합되는 경우, 프록시 요소가 버퍼로부터 방출될 필요가 있다(단계 310). 프로세스(300)는 입력 요소가 존재하는 한, 모든 입력 요소에 대해 반복될 수 있다.

[0048] 본 명세서에 설명된 기술들은 컴퓨터상에서 실행되는 소프트웨어를 이용하여 구현될 수 있다. 예를 들어, 소프트웨어는, 적어도 하나의 프로세서, 적어도 하나의 데이터 저장 시스템(휘발성 및 비휘발성 메모리 및/또는 저장 요소를 포함), 적어도 하나의 입력 장치 또는 포트, 및 적어도 하나의 출력 장치 또는 포트를 각각 포함하는, 하나 이상의 프로그램된 또는 프로그램가능한 컴퓨터 시스템(분산형, 클라이언트/서버형, 또는 그리드형 등의 다양한 구조가 될 수 있음)에서 실행하는 하나 이상의 컴퓨터 프로그램에서 절차를 구성한다. 소프트웨어는, 예컨대, 연산 그래프의 설계 및 구성과 관련된 다른 서비스를 제공하는 더 큰 프로그램의 하나 이상의 모듈을 형성할 수 있다. 그래프의 노드 및 요소는 컴퓨터로 판독가능한 매체에 저장된 데이터 구조 또는 데이터 레포지토리에 저장된 데이터 모델에 부합하는 다른 조직화된 데이터로서 구현될 수 있다.

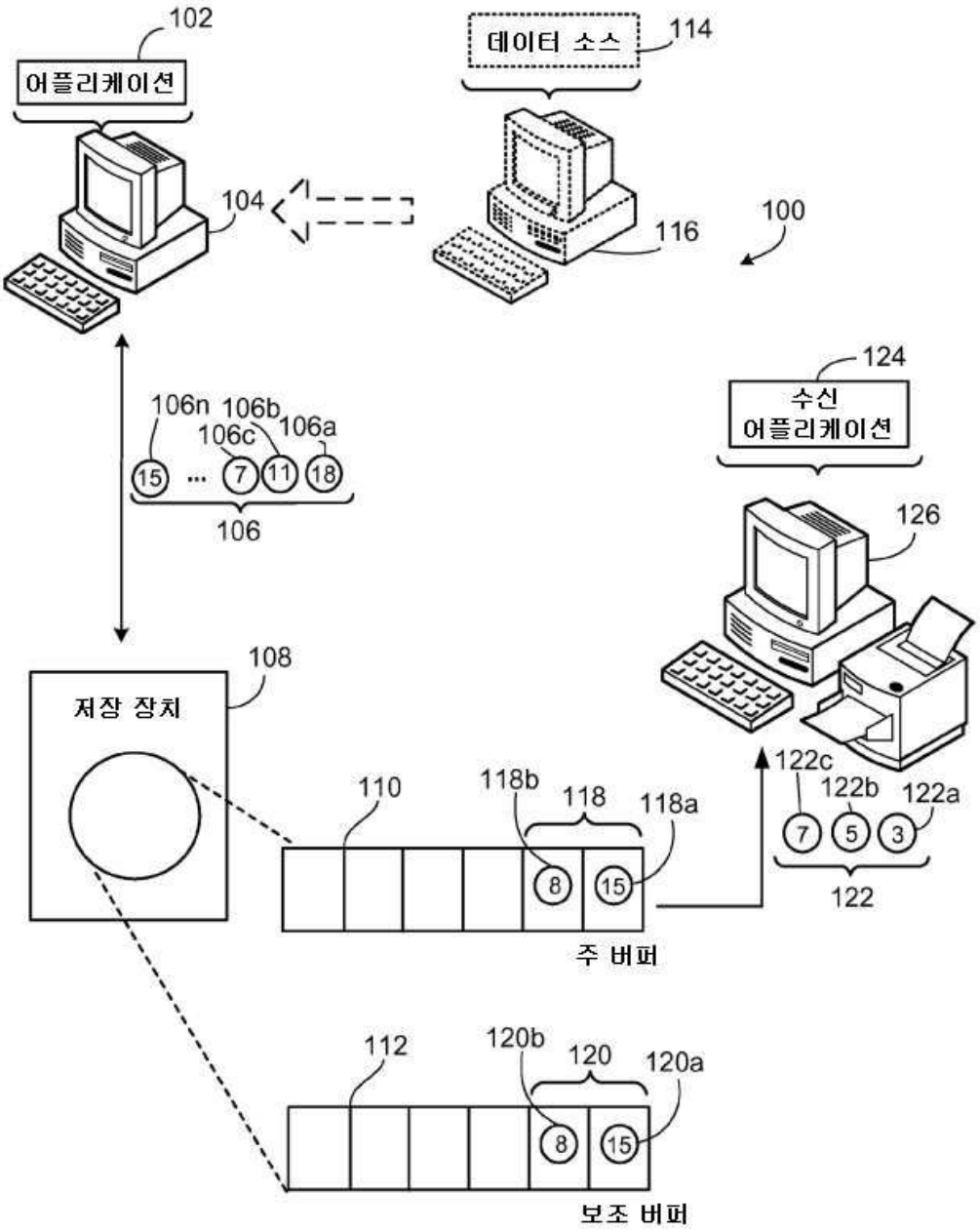
[0049] 소프트웨어는 범용 또는 전용의 프로그램가능한 컴퓨터로 판독가능한 CD-ROM과 같은 저장 매체에 제공되거나, 네트워크와 같은 통신 매체를 통해 실행가능한 컴퓨터로 전달(전파 신호로 부호화되는)될 수 있다. 모든 기능은, 전용의 컴퓨터상에서, 또는 코프로세서(coprocessor)와 같은 전용의 하드웨어를 사용해서 수행될 수 있다. 소프트웨어는, 해당 소프트웨어에 의해 특정된 연산의 상이한 부분이 여러 컴퓨터에서 수행되는 분산 방식으로 구현되어도 된다. 이러한 각각의 컴퓨터 프로그램은, 본 명세서에서 설명된 절차를 수행하도록 저장 매체 또는 디바이스가 컴퓨터 시스템에 의해 판독될 때에 컴퓨터를 구성 및 운용하기 위한, 범용 또는 전용의 프로그램가능한 컴퓨터에 의해 판독가능한 저장 매체 또는 디바이스(예를 들어, 고체 메모리 또는 매체, 또는 자기 또는 광학 매체)에 저장되거나 다운로드되도록 하는 것이 바람직하다. 본 발명의 시스템은 컴퓨터 프로그램에 맞게 구성된, 컴퓨터로 판독가능한 저장 매체로서 구현될 수도 있는데, 이러한 저장 매체는 컴퓨터 시스템으로 하여금, 본 명세서에서 설명한 기능의 수행을 위해 특정되고 미리 정해진 방식으로 동작할 수 있도록 구성된다.

[0050] 본 발명의 다수의 실시예를 설명하였다. 그러나, 본 발명의 범위를 벗어남이 없이 다양한 변형이 가능하다는 것을 알 수 있을 것이다. 예를 들어, 상기 설명한 단계들 중 몇몇은 반드시 그 순서대로 수행되지 않아도 되며, 설명된 것과 다른 순서대로 수행되어도 된다.

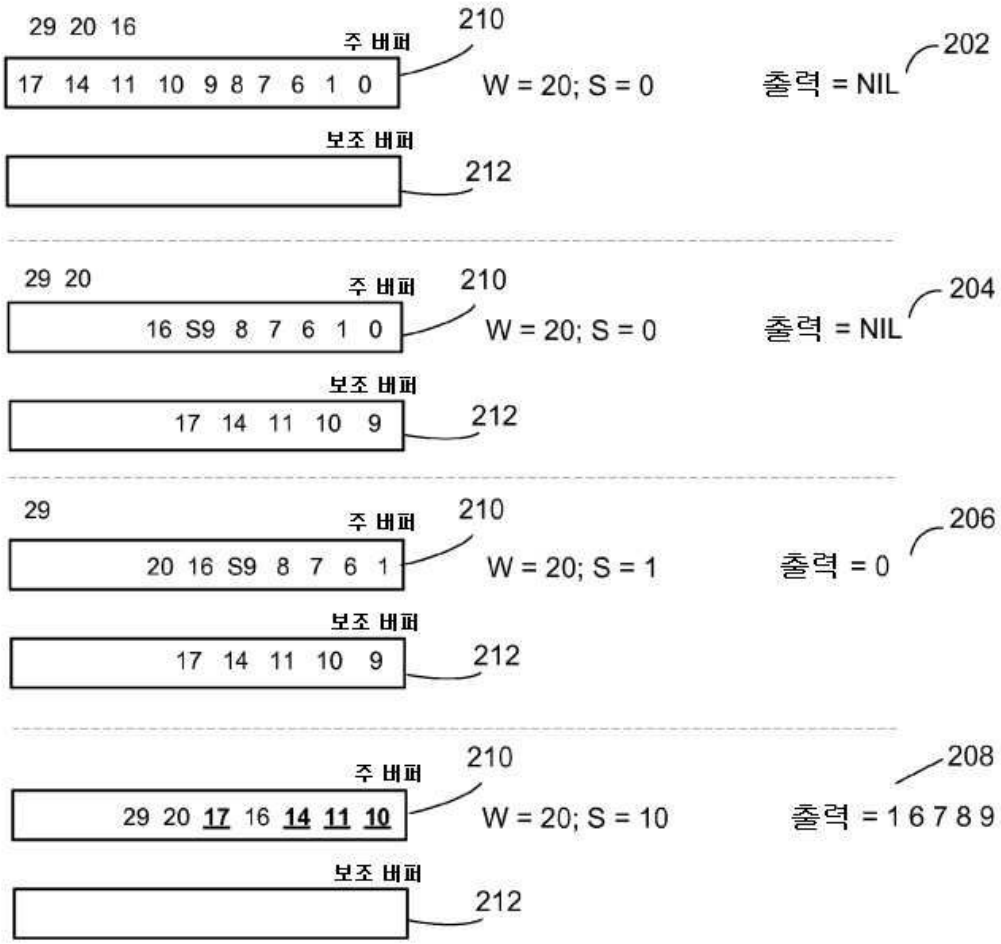
[0051] 이상의 설명은 청구항의 범위에 의해 정해지는 본 발명의 범위를 제한하기 위한 것이 아니고 예시일 뿐이다. 예를 들어, 앞서 설명한 많은 기능적 단계들은 전체적인 프로세싱에 실질적인 영향을 미치지 않으면서, 다른 순서로 수행되어도 된다. 다른 실시예들은 이하의 청구항의 범위에 포함된다.

도면

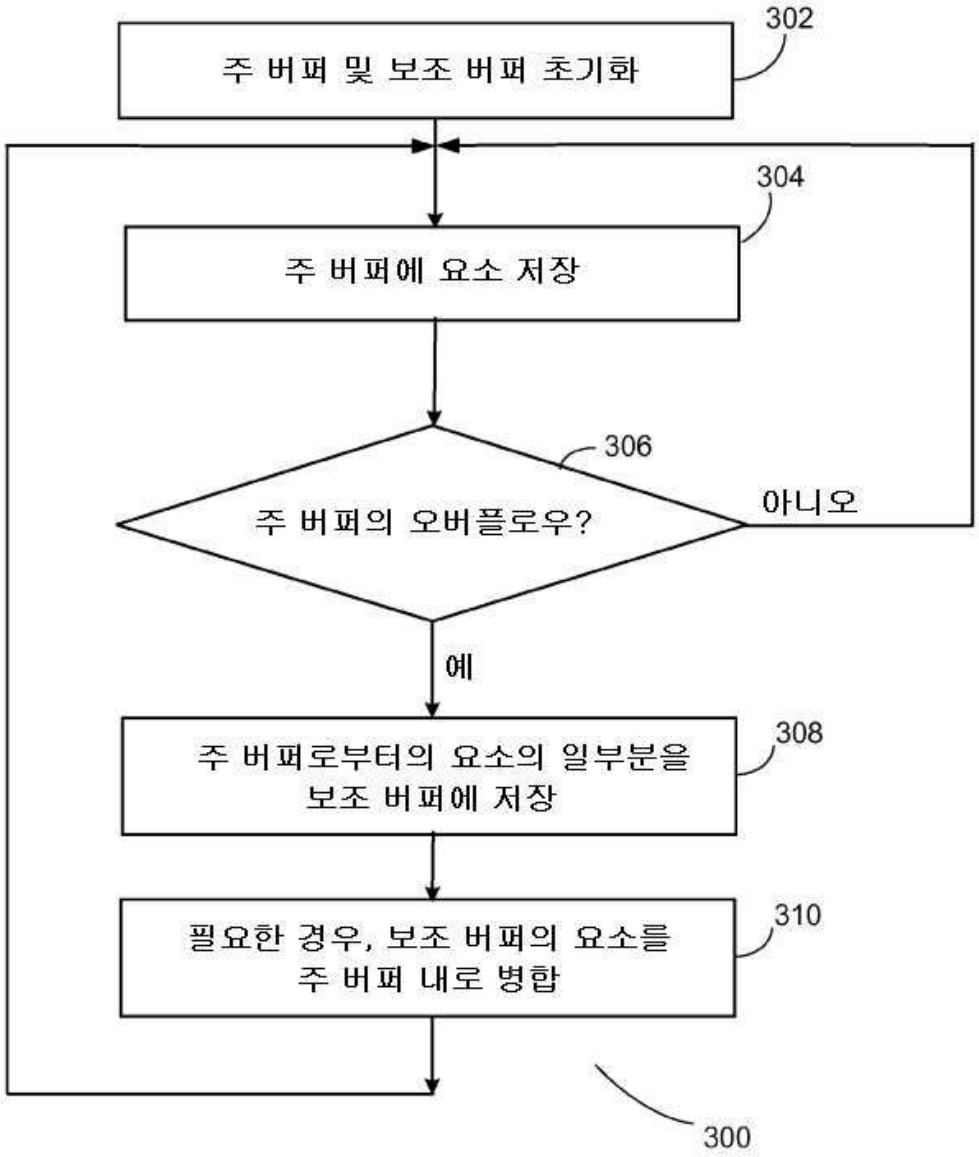
도면1


102
어플리케이션
104
데이터 소스
114
116
100
124
수신
어플리케이션
126
106n
106b
106c
106a
15
...
7
11
18
106
108
저장 장치
110
118b
118
118a
8
15
주 버퍼
122c
122b
122a
7
5
3
122
112
120b
120
120a
8
15
보조 버퍼

도면2


29 20 16
주 버퍼
17 14 11 10 9 8 7 6 1 0
210
W = 20; S = 0
출력 = NIL
202
보조 버퍼
212
29 20
주 버퍼
16 S9 8 7 6 1 0
210
W = 20; S = 0
출력 = NIL
204
보조 버퍼
17 14 11 10 9
212
29
주 버퍼
20 16 S9 8 7 6 1
210
W = 20; S = 1
출력 = 0
206
보조 버퍼
17 14 11 10 9
212
주 버퍼
29 20 17 16 14 11 10
210
W = 20; S = 10
출력 = 1 6 7 8 9
208
보조 버퍼
212

도면3


주 버퍼 및 보조 버퍼 초기화
302
주 버퍼에 요소 저장
304
주 버퍼의 오버플로우?
306
아니오
예
주 버퍼로부터의 요소의 일부분을 보조 버퍼에 저장
308
필요한 경우, 보조 버퍼의 요소를 주 버퍼 내로 병합
310
300