(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년07월13일
(11) 등록번호 10-0969497
(24) 등록일자 2010년07월05일

(51) Int. Cl.
G06F 11/34 (2006.01) *G06F 15/00* (2006.01)
(21) 출원번호 10-2005-7005267
(22) 출원일자(국제출원일자) 2003년08월22일
심사청구일자 2008년07월22일
(85) 번역문제출일자 2005년03월25일
(65) 공개번호 10-2005-0071515
(43) 공개일자 2005년07월07일
(86) 국제출원번호 PCT/US2003/027429
(87) 국제공개번호 WO 2004/029809
국제공개일자 2004년04월08일
(30) 우선권주장
10/255,427 2002년09월26일 미국(US)
(56) 선행기술조사문헌
US04872121 A1
US05937437 A1
전체 청구항 수 : 총 5 항

(73) 특허권자
프리스케일 세미컨덕터, 인크.
미국 텍사스 (우편번호 78735) 오스틴 윌리암 캐논 드라이브 웨스트 6501
(72) 발명자
카브랄, 카로스, 제이.
미국, 텍사스 78727, 오스틴, 랜치 로드 620 엔. 샵 428, 13425
누네즈, 조세, 엠.
미국, 텍사스 78727, 오스틴, 블랙 앤저스 드라이브 12107
(74) 대리인
이범래, 장훈

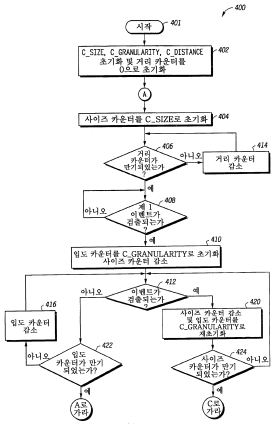
심사관 : 이경홍

(54) 성능 모니터와 그를 위한 방법

(57) 요 약

본 발명의 실시예들은 일반적으로 사이즈 문턱 값(206,216), 입도 문턱 값(208,218), 및 거리 문턱 시간(210,220)에 기초하여 선택된 이벤트 종류의 다수의 적격 클러스터들을 카운트하는 방법 및 모니터링 유닛에 관한 것이다. 일 실시예에서, 적격 클러스터는 사이즈 문턱 값 및 입도 문턱 값을 만족하는 클러스터이다. 일 실시예에서, 적격 클러스터가 검출되고, 이전에 카운트된 적격 클러스터 후 적어도 거리 문턱 시간에 발생한다면 클러스터 카운터에 의해 카운트된다. 예를 들어, 일 구현에서, 성능 모니터(104)는, 다음 적격 클러스터의 검출 전에 거리 문턱 시간이 만기될 때까지 기다린다. 대안적으로, 클러스터 카운터에 의해 카운트되었는지의 여부에 상관없이, 이전의 적격 클러스터 후 적어도 거리 문턱 시간에 발생하는 경우에만 클러스터 카운터가 적격 클러스터를 카운트하는 모든 적격 클러스터들이 검출된다.

대 표 도 - 도4



특허청구의 범위

청구항 1

성능 모니터(performance monitor)에 있어서,

이벤트의 표시를 수신하도록 동작하는 입력;

상기 입력에 결합된 제어 로직으로서, 사이즈 문턱 값 및 입도 문턱 값(granularity threshold)을 만족하는 이벤트의 적격 클러스터의 발생을 결정하도록 동작하는 상기 제어 로직; 및

상기 제어 로직에 결합된 카운터로서, 상기 카운터는 상기 이벤트의 다수의 적격 클러스터들을 표시하며, 상기 카운터에 의해 표시된 각각의 후속하는 적격 클러스터는 상기 카운터에 의해 표시된 이전의 적격 클러스터 후 적어도 거리 문턱 시간에 발생하는, 상기 카운터를 포함하는, 성능 모니터.

청구항 2

제 1 항에 있어서, 상기 카운터는 상기 이벤트의 다수의 적격 클러스터들을 표시하며, 상기 카운터에 의해 표시된 각각의 후속하는 적격 클러스터는 이전의 적격 클러스터 후 적어도 상기 거리 문턱 시간에 발생하는, 성능 모니터.

청구항 3

삭제

청구항 4

삭제

청구항 5

삭제

청구항 6

삭제

청구항 7

삭제

청구항 8

프로세서와 모니터링 유닛을 포함하는 데이터 처리 시스템에 있어서,

상기 모니터링 유닛은:

이벤트의 표시를 수신하도록 동작하는 입력;

상기 입력에 결합된 제어 로직으로서, 사이즈 문턱 값 및 입도 문턱 값을 만족하는 이벤트의 적격 클러스터의 발생을 결정하도록 동작하는 상기 제어 로직; 및

상기 제어 로직에 결합된 카운터로서, 상기 카운터는 상기 이벤트의 다수의 적격 클러스터들을 표시하며, 상기 카운터에 의해 표시된 각각의 후속하는 적격 클러스터는 상기 카운터에 의해 표시된 이전의 적격 클러스터 후 적어도 거리 문턱 시간에 발생하는, 상기 카운터를 포함하는, 데이터 처리 시스템.

청구항 9

삭제

청구항 10

삭제

청구항 11

삭제

청구항 12

삭제

청구항 13

삭제

청구항 14

전자 시스템의 이벤트의 다수의 클러스터들의 카운트를 제공하는 방법에 있어서,

이벤트의 제1 적격 클러스터의 발생을 결정하고, 카운트를 증가시키는 단계로서, 상기 결정 단계는 상기 이벤트의 사이즈 문턱 값이 각각의 입도 문턱 시간 내에 발생하는지 결정하는 단계를 포함하는, 상기 클러스터 발생 결정 단계; 및

제2 적격 클러스터가 상기 제1 적격 클러스터의 발생으로부터 적어도 거리 문턱 시간 후 발생하는 것으로 결정되면 상기 카운트를 증가시키는 단계를 포함하는, 카운트 제공 방법.

청구항 15

삭제

청구항 16

삭제

청구항 17

삭제

청구항 18

삭제

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

제 14 항에 있어서, 상기 이벤트는 버스 트랜잭션을 포함하는, 카운트 제공 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 성능 모니터들, 특히, 이벤트들의 클러스터를 모니터링하는 것에 관한 것이다.

배 경 기 술

[0002] 성능 모니터링(performance monitoring)은 일반적으로 데이터 처리 시스템들의 동작 특성을 나타내는데 사용된다. 예를 들어, 데이터 처리 시스템 내의 특정 이벤트들(메모리 액세스 등)은 성능 특성을 나타내기 위해 식별 및 모니터링될 수 있다. 하나의 공지된 성능 모니터는 메모리 액세스들과 같은 이벤트들을 모니터링하고, 클러스터 사이즈 및 클러스터 입도(granularity)에 기초하여 상기 이벤트들의 클러스터링 특성을 나타낼 수 있다. 상기 해결책에서, 클러스터 사이즈는 클러스터를 구성하는 이벤트 발생들의 최소 수를 지정하고, 클러스터 입도는 클러스터의 일부로 간주될 이벤트 발생들에 대해 개개의 이벤트 발생들 간의 사이클들의 최대 허용가능한 수를 지정한다. 그러므로, 성능 모니터는 메모리 액세스들의 클러스터링에 관한 데이터 처리 시스템의 동작 특성을 나타낼 수 있다. 그러나, 이 이벤트 발생들의 분배를 정량화하는 것은 어렵기 때문에, 이벤트들의 클러스터링 특성을 나타내기 위해 단지 클러스터 사이즈와 입도를 사용하는 것은 이벤트 클러스터링의 특성을 잘못 나타내게 한다. 그러므로, 데이터 처리 시스템 동작의 보다 나은 특성 표시와 보다 나은 이해를 달성하기 위해서, 개선된 이벤트 프로파일링에 대한 필요성이 존재한다.

발명의 상세한 설명

[0003] 본 발명은 예로써 도시되고, 유사한 참조들이 유사한 요소들을 나타내는 첨부 도면들에 의해 제한되지 않는다.

실 시 예

[0009] 도면들의 요소들은 간단함과 명료함을 위해 도시되고, 반드시 일정한 비례로 축소하여 그려진 것은 아니라는 점을 당업자는 알 수 있다. 예를 들어, 도면들에서 일부 요소들의 크기는 본 발명의 실시예들에 대한 이해를 향상시키는 것을 돕기 위해 다른 요소들에 비해 과장될 수 있다.

[0010] 여기서 사용된 바와 같이, 용어 "버스"는 데이터, 어드레스들, 제어, 또는 상태와 같은 하나 이상의 다양한 종류의 정보를 전송하는데 사용될 수 있는 복수의 신호들 또는 컨덕터들을 언급하는데 사용된다. 여기서 논의된 컨덕터들은 단일 컨덕터, 복수의 컨덕터들, 단방향 컨덕터들, 또는 양방향 컨덕터들에 관하여 설명 또는 기술될 수 있다. 그러나, 상이한 실시예들은 컨덕터들의 구현을 바꿀 수 있다. 예를 들어, 양방향 컨덕터들보다 별개의 단방향 컨덕터들이 사용될 수 있고, 그 역도 또한 같다. 또한, 복수의 컨덕터들은 다중 신호들을 직렬 또는 시간 다중화된 방식으로 전송하는 단일 컨덕터로 대체될 수 있다. 또한, 다중 신호들을 나르는 단일 컨덕터들은 이 신호들의 서브셋트들을 나르는 다양하고 상이한 컨덕터들로 분리될 수 있다. 그러므로, 신호들을 전송하기 위해 다수의 옵션들이 존재한다.

[0011] 용어들 "어설트(assert)" 및 "니게이트(negate)"(또는 "디어설트(deassert)")는, 신호, 상태 비트, 또는 유사한 장치를 각각 논리적으로 참이거나 논리적으로 거짓 상태로 랜더링하는 것을 언급할 때 사용된다. 논리적으로 참 상태가 논리적 레벨 1이면, 논리적으로 거짓 상태는 논리적 레벨 0이다. 그리고, 논리적으로 참 상태가 논리적 레벨 0이면, 논리적으로 거짓 상태는 논리적 레벨 1이다. 또한, 여기서 사용된 바와 같이, 만기된 카운터 또는 만기되는 카운터는 0에 도달한 카운터를 나타낸다.

[0012] 이벤트 클러스터링의 특성 표시는 데이터 처리 시스템이나 임의 종류의 전자 시스템의 동작을 프로파일링할 수 있게 한다. 이 특성 표시는, 액세스들이 액세스들의 클러스터들 간에 비균일한 거리로 시간에 따라 전개되는 임의의 장치 또는 메모리에 대한 액세스들의 동작을 이해하는데 사용될 수 있다. 이 특성 표시는 주어진 인터페이스의 최대 대역폭 요구, 이 피크 대역폭 요구의 기간, 및 이 피크들 간의 거리를 결정하는데 사용될 수 있다. 그 결과들은 시스템의 전체 성능을 증가시키도록, 소프트웨어를 튜닝하고 이 피크들의 특성들을 변경하기 위해, 소프트웨어 프로그래머에 의해 사용될 수 있다. 이 특성 표시를 통해 획득된 정보의 또다른 사용은, 최대 시스템 대역폭 요구와 이 피크 요구들의 분배에 대한 아이디어를 하드웨어 설계자들에게 제공하는 것이다. 하드웨어 설계자들은 디자인 교환(trade-off)을 결정하고 이 요구를 최적의 방식으로 조정하도록 상기 정보를 사용할 수 있다.

[0013] 이벤트는 데이터 처리 시스템이나 전자 시스템 내의 임의의 이벤트를 언급할 수 있다는 점에 유의하라. 일반적으로, 이벤트는 예를 들어, 논리 신호의 어설션 또는 디어설션에 의해 검출될 수 있는 임의의 활동이다. 예를 들어, 일 실시예에서, 이벤트들은 메모리 액세스들(즉, 요청들을 판독하고 요청들을 기록함), 이더넷 인터페이스를 통해 이더넷 제어기로부터 수신된 프레임들, 주변 장치 액세스들, 통신 버스 상에 유효 데이터의 표시들,

또는 임의 종류의 버스 트랜잭션들을 포함한다. 대안적으로, 임의의 다른 이벤트 종류가 규정될 수 있다.

[0014] 도 1은 프로세서(102), 메모리(106), 다른 주변 인터페이스들(108), 이더넷 제어기(110) 및 시스템 버스(112)를 갖는 데이터 처리 시스템(100)을 블록 다이어그램 형태로 도시한다. 프로세서(102)는 성능 모니터(104)를 포함하고, 양방향 컨덕터(114)를 통해 시스템 버스(112)에 결합된다. 메모리(106)는 양방향 컨덕터들들(116)을 통해 시스템 버스(112)에 결합되고, 이더넷 제어기는 양방향 컨덕터들(120)을 통해 시스템 버스(112)에 결합되며, 다른 주변 인터페이스들(108)은 양방향 컨덕터들(118)을 통해 시스템 버스(112)에 결합된다. 다른 주변 인터페이스들(108)은 양방향 컨덕터들(112)을 통해 다른 주변 장치들에 결합되고, 이더넷 제어기(110)는 양방향 컨덕터들(124)을 통해 통신 네트워크에 결합된다. 일 실시예에서, 데이터 처리 시스템(100)은 당업계에 공지된 바와 같이, 마이크로프로세서, 마이크로제어기, 디지털 신호 처리기 등이다. 메모리(106)는 예를 들어, RAM(random access memory), SRAM(static RAM), DRAM(dynamic RAM), ROM(read only memory), 플래시, EEPROM(electrically erasable and programmable ROM), EPROM(erasable and programmable ROM) 등과 같은 휘발성 또는 비휘발성 메모리들을 포함하는 임의 종류의 메모리일 수 있다. 이더넷 제어기(110)는 데이터 처리 시스템(100) 내에 위치될 수 있는 주변 장치의 한 종류이나, 데이터 처리 시스템(100)은 또한 예를 들어, 입/출력(I/O) 장치들, UART(universal asynchronous receiver transmitter), RTC(real time clock) 등과 같은 주변 장치들에 인터페이스하는 다른 주변 인터페이스들(108)을 포함할 수 있다. 다른 주변 인터페이스들(108)은 또한 PCI(Peripheral Component Interface) 인터페이스 및 RapidIO 인터페이스와 같은 인터페이스들을 포함할 수 있다. 또한, 데이터 처리 시스템(100)은 도 1에 도시된 것 대신에 또는 그에 부가하여, 시스템 버스(112)에 결합된 다수의 메모리들과 다수의 주변 장치들을 포함할 수 있다.

[0015] 일 실시예에서, 데이터 처리 시스템(100) 모두는 예를 들어, SoC(system on chip) 솔루션과 같은 단일 칩에 통합된다. 그러나, 대안적인 실시예들에서, 데이터 처리 시스템의 부분들은 예를 들어, 칩 셋트 내와 같이 상이한 칩들에 위치된 별개의 집적 회로들 상에 있을 수 있다. 예를 들어, 메모리(106)는 프로세서(102)와 별개의 집적 회로 상에 위치될 수 있다. 또한, 성능 모니터(104)는 프로세서(102) 내에 통합되거나(도 1에 도시됨), 프로세서(102) 외부의 별개의 집적 회로 상에 위치될 수 있다. 일 실시예에서, 성능 모니터(104)는 독립형 ASIC(stand alone Application Specific Integrated Circuit)이거나, 예를 들어, 논리 분석기와 같이 데이터 처리 시스템(100)과 별개의 전자 시스템에 위치될 수 있다. 동작 중에, 프로세서(102)는 당업계에 공지된 바와 같이, 메모리(106)나 다른 주변 장치들로부터 수신된 명령들을 처리하고, 메모리(106), 다른 주변 장치들, 및 이더넷 제어기(110)와 데이터를 송수신할 수 있다.

[0016] 도 2는 도 1의 성능 모니터(104)의 일 실시예를 도시한다. (성능 모니터(104)는 또한 모니터링 유닛으로 언급될 수 있다는 점에 유의하라.) 성능 모니터(104)는 카운터 제어 레지스터 1(200)부터 카운터 제어 레지스터 N(202), 제어 회로(234), 및 클러스터 카운터 1(230)부터 클러스터 카운터 N(232)를 포함한다. 카운터 제어 레지스터 1(200), 카운터 제어 레지스터 N(202), 클러스터 카운터 1(230), 및 클러스터 카운터 N(232)는 각각 제어 회로(234)에 양방향으로 결합된다. 카운터 제어 레지스터 1(200)은 클러스터 사이즈 문턱 값 필드(c_size(206)), 클러스터 입도 문턱 값 필드(c_granularity(208)), 클러스터 거리 문턱 시간 필드(c_distance(210)), 이벤트 선택 필드(event_select(212)), 및 그 외 필드들(204)을 포함한다. 유사하게, 카운터 제어 레지스터 N(202)는 클러스터 사이즈 문턱 값 필드(c_size(216)), 클러스터 입도 문턱 값 필드(c_granularity(218)), 클러스터 거리 문턱 시간 필드(c_distance(220)), 이벤트 선택 필드(event_select(222)), 및 그 외 필드들(214)을 포함한다. 제어 회로(234)는 이벤트 선택 회로(236)와 내부 카운터들(238)을 포함하고, 프로세서(102)의 다양한 부분들로부터 이벤트 신호들(229)을 수신한다. 그러므로, 성능 모니터(104)는 필요에 따라, N개의 제어 레지스터들과 N개의 클러스터 카운터들을 포함할 수 있고, 여기서 N은 임의의 수라는 점에 유의하라. 또한, 성능 모니터(104)의 제어 레지스터들은 여러가지 상이한 방법들로 구성될 수 있다는 점에 유의하라. 예를 들어, 일 실시예에서, 별개의 제어 레지스터들은 옵션들을 클러스터링하는데 사용될 수 있거나, 카운터 제어 레지스터들은 필요에 따라 상이한 다수의 필드들을 갖는 보다 큰 레지스터들에 결합될 수 있다.

[0017] 동작 중에, 성능 모니터는 각각의 카운터 제어 레지스터에 대응하는 이벤트 카운터를 포함한다. 예를 들어, 클러스터 카운터 1(230)는 카운터 제어 레지스터 1(200)에 대응하는 이벤트 카운터이다. 필드 event_select(212)는 어떤 이벤트 종류가 모니터링될지를 식별하는데 사용된다. 예를 들어, 메모리 액세스들(예를 들어, 시스템 버스(112)를 통해 프로세서(102) 또는 다른 주변 장치와 메모리(106) 간에 발생하는 판독 또는 기록 또는 양자 모두)이 모니터링되거나 이더넷 인터페이스를 통해 수신된 프레임들이 모니터링되도록 event_select(212)가 설정될 수 있다. 그러므로, event_select(212)는 하나의 이벤트 종류를 식별하도록 설정될 수 있는 한편,

event_select(222)는 동일하거나 또다른 이벤트 종류를 식별하도록 선택될 수 있다. 이벤트 클러스터링이 카운터 제어 레지스터 1(200)(c_size(206)는 0으로 설정됨)에 대해 인에이블되지 않는다고 가정하면, 클러스터 카운터 1(230)은 단순히 event_select(212)에 의해 표시된 종류의 이벤트들의 수를 카운트하는 이벤트 카운터로서 동작하고, 여기서 그 외 필드들(204)은 또한 옵션들을 규정하는데 사용될 수 있다. 예를 들어, 이벤트 신호들(228)은, 메모리 기록들과 메모리 판독들이 발생하는 때를 나타내는 논리 신호들을 포함할 수 있다. even_select(212)는 오직 클러스터 카운터 1(230)(이벤트 카운터로서 동작함)에 의한 메모리 기록들의 모니터링을 선택하도록 설정될 수 있다. 그러므로, 이벤트 선택 회로(236)는 메모리 기록들의 발생을 나타내는 이벤트 신호들(228) 중 적절한 인입 이벤트 신호를 선택하도록 event_select(212)를 사용하고, 적절한 인입 이벤트 신호의 어설션을 수신하면, 대응하는 이벤트 카운터(클러스터 카운터 1(230))가 증가된다.

[0018] 이벤트 클러스터링이 인에이블되는 경우(c_size(206)가 0이 아님), 클러스터 카운터 1(230)는 c_size(206), c_granularity(208), 및 c_distance(210)에 의해 설정된 문턱 값들을 만족하는 선택된 이벤트 종류의 클러스터들(event_select(212)에 의해 선택됨)을 카운터하도록 동작한다. 그러므로, 클러스터 카운터 1(230)이 선택된 이벤트 종류 중 단일 이벤트들의 발생들을 더이상 카운트하지 않고, 선택된 이벤트 종류의 클러스터들의 발생을 카운트하기 위해, 이벤트 클러스터링이 인에이블될 때, 필드들 c_size(206), c_granularity(208), 및 c_distance(210)이 사용된다. (클러스터 제어 레지스터 N(202)과 클러스터 카운터 N(232)은 클러스터 제어 레지스터 1(200)과 클러스터 카운터 1(230)과 유사하게 동작한다는 점에 유의하라.) 여기서 논의된 실시예에서, c_size(206)와 c_granularity(208)은 적격 클러스터들(qualified clusters)을 규정한다. 즉, 적격 클러스터(도 3을 참조하여 더 기술될 것임)는 각각 c_size(206)와 c_granularity(208)에 의해 설정된 사이즈 문턱 값과 입도 문턱 값들 모두를 만족하는 것이다. 사이즈 문턱 값은 이벤트들의 클러스터를 구성하는 이벤트 발생들의 최소 수를 지정한다. 일반적으로, 클러스터는 동일한 이벤트 종류의 그룹화로서 규정되나, 대안적으로, 클러스터는 하나 이상의 이벤트 종류의 그룹화로 언급할 수 있다. 입도 문턱 값은 동일한 클러스터의 일부로 간주될 이벤트 발생들에 대해 개개의 이벤트 발생들 간의 시간 또는 사이클들(예컨대, 클럭 사이클들)의 최대 허용가능한 수를 지정한다.

[0019] c_distance(210)에 의해 설정된 거리 문턱 시간은, 어떤 적격 클러스터들이 클러스터 카운터 1(230)에 의해 카운트되는지를 결정하도록 사용된다. 후속하는 적격 클러스터가 클러스터 카운터 1(230)에 의해 개개의 클러스터로서 카운트되기 위해, 거리 문턱 시간은 이전의 적격 클러스터의 미리결정된 점과 후속하는 적격 클러스터의 미리결정된 점 간의 시간 또는 사이클들(예컨대, 클럭 사이클들)의 수용가능한 수를 지정한다. 예를 들어, 일 실시예에서, 거리 문턱 시간은 이전의 적격 클러스터의 끝과 후속하는 적격 클러스터의 시작 간의 시간 또는 사이클들의 최소 수용가능한 수를 지정할 수 있다. 대안적인 실시예에서, 거리 문턱 시간은 c_size개의 이벤트 발생들이 이미 후속하는 적격 클러스터 내에서 검출되었던 점과 이전의 적격 클러스터의 끝 간의 시간 또는 사이클들의 최소 수용가능한 수를 지정할 수 있다. 대안적으로, 거리 문턱 시간은 이전 클러스터의 시작점 또는 끝점과, 후속하는 클러스터의 시작점 또는 끝점 간의 시간 또는 사이클들의 최소 수용가능한 수를 지정할 수 있다. 대안적으로, c_size개의 이벤트 발생들이 이미 이전의 적격 클러스터 내에서 검출되었던 점과, c_size개의 이벤트 발생들이 이미 후속하는 적격 클러스터 내에서 검출되었던 점 간의 시간 또는 사이클들의 최소 수용가능한 수를 지정할 수 있다. 또한, 또다른 실시예에서, 거리 문턱 시간은 카운트되었던 이전의 적격 클러스터의 미리결정된 점과, 후속하는 적격 클러스터의 미리결정된 점 간의 시간 또는 사이클들의 최소 수용가능한 수를 지정한다. 그러므로, 거리 문턱 시간은 여러가지 상이한 방식으로 규정될 수 있다.

[0020] 도 2를 다시 참조하여, c_size(206)가 0보다 큰 값을 갖는다면, 클러스터 카운트 1(230)이 c_size(206), c_granularity(208), 및 c_distance(210)를 만족하는 이벤트 종류(event_select(212)에 의해 선택됨)의 적격 클러스터들의 수를 카운트하도록 클러스터링이 인에이블된다. 즉, 전술된 바와 같이, 클러스터 카운터 1(230)이 선택된 이벤트 종류의 클러스터들을 카운트하도록, 이벤트 선택 회로(236)는 event_select(212)에 따라 어느 이벤트 신호들(228)을 모니터링할지를 선택한다. 유사하게, 클러스터링이 또한 카운터 제어 레지스터 N(202)(0보다 큰 c_size(216)를 가짐)에서 인에이블된다면, 클러스터 카운터 N(232)이 선택된 이벤트 종류의 클러스터들을 카운트하도록, 이벤트 선택 회로(236)는 event_select(222)에 따라 어느 이벤트 신호들(228)을 모니터링할지를 선택한다. 도 4 및 도 5의 흐름도와 관련하여 보다 자세하게 기술되는 바와 같이, c_size(206), c_granularity(208), 및 c_distance(210)에 의해 설정된 문턱 값들이 만족되는지를 결정하도록 내부 카운터들(238)이 사용된다. (카운터 제어 레지스터 1(200)과 클러스터 카운터 1(230)에 관하여 여기서 제공된 설명들은 또한 카운터 제어 레지스터 N(202)과 클러스터 카운터 N(232)에 적용한다는 점에 유의하라.)

[0021] 도 3은 사이즈 문턱 값, 입도 문턱 값, 및 거리 문턱 시간의 이해를 돕는 이벤트 클러스터링의 예(300)를 도시

한다. 도 3은 두개의 클러스터들(302,304)을 도시한다. 클러스터(302)를 참조하여, 다수의 이벤트 발생들이 도시된다. 이벤트 발생들의 수가 사이즈 문턱 값 요구(이 경우, 5)를 만족하고, 클러스터(302) 내 임의의 두개의 이벤트들 간의 거리가 입도 문턱 값(c_granularity)보다 작기(또는 동일) 때문에, 클러스터(302)는 적격 클러스터로 간주된다. 적격 클러스터(302)에 대해 설명된 것과 동일한 이유로, 클러스터(304)도 사이즈 문턱 값보다 큰(또는 동일) 다수의 이벤트 발생들을 포함하고, 임의의 두개의 이벤트들 간의 거리가 입도 문턱 값보다 작기(또는 동일) 때문에 적격 클러스터이다. 도 3의 예(300)에서, 클러스터 거리(즉, 두개의 클러스터들 간의 시간 또는 클럭 사이클의 양)는, T1으로 표시된 이전의 적격 클러스터(302)의 끝에서부터 T3으로 표시된 후속하는 적격 클러스터(304)의 시작까지로 규정된다. 예(300)에서, T2는 문턱 거리 시간(c_distance)이 만기되는 시간을 나타낸다. 즉, T1과 T2 간의 시간은 문턱 거리 시간(c_distance)을 나타낸다. 그러므로, 클러스터 거리가 c_distance보다 크기 때문에, 적격 클러스터(304)가 클러스터 카운터에 의해 카운트된다. 그러나, 적격 클러스터들(304,302) 간의 클러스터 거리가 거리 문턱 시간을 만족하지 않았다면, 적격 클러스터(304)는 카운트되지 않을 것이다. 즉, T2 후의 이벤트 발생들의 수가 문턱 거리 요구를 만족하지 않도록 T3가 T2 이전에 발생했다면, 적격 클러스터(304)는 카운트되지 않을 것이다. 그러므로, 도시된 실시예에서, 도 4 및 도 5와 관련하여 이하에서 보다 자세하게 기술되는 바와 같이, 제1 적격 클러스터의 발생 후, 성능 모니터(104)는 다음의 적격 클러스터를 검출하기 전에 문턱 거리 시간이 만기되기를 기다린다. 즉, 도시된 실시예에서, 각각의 적격 클러스터가 카운트된 후, 문턱 거리 시간은 후속하는 적격 클러스터를 카운트하기 전에 만기될 것이다.

[0022] 도 4 및 도 5는 본 발명의 일 실시예에 따라 성능 모니터(104)에 의해 사용될 수 있는 적격 클러스터들을 카운트하는 방법을 흐름도 형태로 도시한다. 도 4의 흐름도(400)는 본 발명의 일 실시예에 따라, 적격 클러스터를 검출하는 방법을 도시한다. 흐름도(400)는 시작(401)에서 시작하고, c_size, c_granularity, 및 c_distance가 초기화되는 블록(402)으로 진행한다. (도 4 및 도 5의 설명들에서, c_size, c_granularity, c_distance, 및 클러스터 카운터는 클러스터 제어 레지스터들 1(200) 내지 N(202)과 대응하는 클러스터 카운터들 1(230) 내지 N(232) 중 임의의 하나로 언급할 수 있다는 점에 유의하라.) 일 실시예에서, 사용자가 c_size, c_granularity, 및 c_distance에 대해 원하는 값들을 프로그래밍할 수 있도록, 클러스터 제어 레지스터들은 사용자 프로그래밍 가능하다. 이런 식으로, 사용자는 상이한 문턱 값들을 사용하여 데이터 처리 시스템(100)을 프로파일링할 수 있다. 대안적으로, 그들은 데이터 처리 시스템(100)으로 하드와이어될 수 있다. 또한, 블록(402)에서, 거리 카운터(성능 모니터(104)의 내부 카운터들(238) 중 하나)가 0으로 초기화된다. 흐름은 이후 포인트 A를 경유하여, 사이즈 카운터(성능 모니터(104)의 내부 카운터들(238) 중 또다른 하나)가 c_size의 값으로 초기화되는 블록(404)으로 진행한다. 흐름은 이 후, 결정 마름모(406)로 진행한다.

[0023] 결정 마름모(406)에서, 거리 카운터가 만기되었는지 결정된다. 만기되지 않았다면, 흐름은, 아직 0과 동일하지 않은 경우 거리 카운터가 감소되는 블록(414)으로 진행하고, 흐름은 이 후 결정 마름모(406)로 되돌아간다. 그러므로, 거리 카운터가 만기될 때까지 흐름은 결정 마름모(406)에서 결정 마름모(408)(제1 이벤트 발생에 대한 검색을 시작함)로 진행하지 않는다. 그러므로, 만기된 거리 카운터 전에 발생하는 이벤트들은 적격 클러스터를 검출하는데 있어 사용되지 않는다. 그러나, 거리 카운터가 결정 마름모(406)에서 만기되었다면, 흐름은 결정 마름모(408)로 진행한다.

[0024] 결정 마름모(408)에서, 제1 이벤트가 검출되었는지 결정된다. 검출되지 않았다면, 흐름은 결정 마름모(408)로 되돌아간다. 즉, 거리 카운터가 이미 만기되었기 때문에, 흐름(400)은 제1 이벤트가 검출될 때까지 결정 마름모(408)에 남아 있다. 일단 제1 이벤트가 검출되면, 흐름은 블록(410)으로 진행한다. 결정 마름모(408)에서 검출된 제1 이벤트는 적격 클러스터 내의 제1 이벤트일 수 있는 이벤트이다. 즉, 적격 클러스터를 식별하는 처리는, 거리 카운터가 만기되고 제1 이벤트가 검출된 후까지 시작하지 않는다. 예를 들어, 이 제1 이벤트는 성능 모니터(104)를 시작하면서 검출된 제1 이벤트일 수 있다. 이 제1 이벤트는 또한 이전의 적격 클러스터의 끝 후에, 또는 검출된 제1 이벤트 전에 입도 문턱 값에 의해 허용되었던 것보다 큰 어느 정도의 시간에 발생되었던 이전의 이벤트 후에 검출된 제1 이벤트일 수 있다. 예를 들어, 거리 카운터가 클러스터 또는 적격 클러스터의 발생 중에 만기된다면, 이 제1 이벤트는 또한 클러스터 또는 적격 클러스터 동안 발생하는 이벤트를 언급할 수 있다. 블록(410)에서, 적격 클러스터의 가능한 시작은 제1 이벤트를 검출함으로써 검출되었기 때문에, 입도 카운터(성능 모니터(104)의 내부 카운터들(238) 중 또다른 하나)가 c_granularity로 초기화된다. 또한, 제1 이벤트가 검출되었기 때문에, 사이즈 카운터가 감소된다.

[0025] 흐름은 이 후, 다음 이벤트가 검출되는지를 결정하는(예컨대, 데이터 처리 시스템(100)의 다음 클럭에서, 이벤트가 검출되는지를 결정하도록) 결정 마름모(412)로 진행한다. 다음 이벤트가 검출되지 않으면, 흐름은, 입도 카운터가 만기되었는지 결정되는 결정 마름모(422)로 진행한다. 다음 이벤트가 검출된다면, 입도 카운터가 만

기되었기 때문에 새로운 제1 이벤트가 검출될 수 있도록 흐름은 포인트 A로 되돌아가고, 제1 이벤트와 현재 시간(예컨대, 이벤트가 검출되지 않았던 현재 클럭 사이클) 간의 시간은 입도 문턱 값보다 커서, 제1 이벤트는 적격 클러스터의 시작일 수 없다. 그러므로, 포인트 A에서, 새로운 제1 이벤트에 대한 검색이 시작한다. (포인트 A로 진행하면서, 거리 카운터는 이미 만기되어, 흐름은 새로운 제1 이벤트를 검색하도록 결정 마름모(408)로 직접 하향 진행한다는 점에 유의하라.) 그러나, 결정 마름모(422)에서, 입도 카운터가 만기되지 않았다면, 흐름은, 입도 카운터가 감소되는 블록(416)으로 진행하고, 이 후 결정 마름모(412)로 되돌아간다.

[0026] 결정 마름모(412)에서, 이벤트가 검출된다면, 흐름은 블록(420)으로 진행한다. 결정 마름모(412)에서 검출된 이벤트는, 제1 이벤트(또는 이전의 이벤트)와 현재 이벤트 간의 시간이 입도 문턱 값 내인 것을 나타낸다. 그러므로, 블록(420)에서, 적격 클러스터의 일부일 수 있는 또다른 이벤트가 검출되었다는 것을 나타내도록 사이즈 카운터가 감소된다. 또다른 이벤트가 검출되었기 때문에, 입도 카운터는 또한 블록(420)에서 재초기화되고, 흐름은 결정 마름모(424)로 진행한다. 결정 마름모(424)에서, 사이즈 카운터가 만기되었는지 결정된다. 사이즈 카운터가 아직 만기되지 않았다면, 사이즈 문턱 값은 아직 만족되지 않았고, 흐름은 다음 클럭에서 또다른 이벤트가 검출되는지 결정되는 결정 마름모(412)로 되돌아간다. c_granularity개의 클럭들 내에서 어떤 이벤트도 검출되지 않았다면, 흐름은 전술된 바와 같이, 포인트 A를 경유하여 결정 마름모(422)에서부터 블록(404)로 되돌아갈 것이다. 그러나, 입도 카운터가 만기되기 전에 다음 이벤트가 검출된다면, 흐름은 블록(420)을 통해 결정 마름모(424)로 다시 되돌아갈 것이다.

[0027] 결정 마름모(424)에서, 사이즈 카운터가 만기되었다면, 입도 문턱 값을 만족하는 c_size개의 이벤트들이 검출되고, 따라서 적격 클러스터의 검출을 나타낸다. 즉, 검출되었던 c_size개의 이벤트들 중 임의의 두개의 인접 이벤트들 간의 시간은 기껏해야 입도 문턱 값이다. 또한, 거리 카운터가 만기되기를 기다리는 블록(414)과 결정 마름모(406)에 의해 규정된 루프 때문에, 적격 클러스터의 검출은 성능 모니터(104)를 시작하면서 맨처음 검출된 적격 클러스터이거나, 이전에 검출된 적격 클러스터 이 후 적어도 c_distance에서 발생했던 적격 클러스터였다. 사이즈 문턱 값, 입도 문턱 값, 및 거리 문턱 시간이 만족되기 때문에, 적격 클러스터가 표시되고, 흐름은 도 5의 포인트 C로 진행한다. (본 실시예에서, 결정 마름모(408)에서 검출된 "제1 이벤트"는 클러스터 또는 적격 클러스터의 발생 중에 발생할 수 있기 때문에, 검출된 적격 클러스터는 실제로 보다 큰 클러스터 또는 적격 클러스터의 일부일 수 있다는 점에 유의하라.)

[0028] 도 5는, 적격 클러스터가 검출되는 것을 나타내는 포인트 C로 시작하는 흐름(450)을 도시한다. 그러나, 도시된 실시예에서, 현재 적격 클러스터의 끝이 검출될 때까지 클러스터 카운터는 증가되지 않는다. (대안적인 실시예들에서, 클러스터 카운터는 적격 클러스터의 끝 전에 증가될 수 있다는 점에 유의하라.) 그러므로, 결국 현재 적격 클러스터의 끝을 검출하기 위해, 흐름은 포인트 C를 경유하여 결정 마름모(440)로 진행한다.

[0029] 결정 마름모(440)에서, 이벤트가 검출되는지(다음 클럭에서) 결정된다. 이벤트가 검출된다면, 새로운 이벤트의 검출은 입도 카운터를 재설정해야 하기 때문에, 흐름은, 입도 카운터가 c_granularity로 재초기화되는 블록(442)으로 진행한다. 흐름은 이 후, 결정 마름모(440)로 되돌아간다. 결정 마름모(440)에서, 어떤 이벤트도 다음 클럭에서 검출되지 않는다면, 흐름은, 입도 카운터가 만기되었는지 결정되는 결정 마름모(444)로 진행한다. 만기되지 않았다면, 흐름은, 입도 카운터가 감소되는 블록(446)으로 진행하고, 이 후 흐름은, 다음 클럭에서 이벤트가 검출되는지를 결정하는 결정 마름모(440)로 진행한다. 그러므로, 이벤트가 검출되거나 입도 카운터가 만기되지 않는 한, 흐름은 계속해서 결정 마름모(440)로 되돌아간다. 결정 마름모(444)에서, 입도 카운터가 만기되었다면, 임의의 후속하는 이벤트는 입도 문턱 값을 초과할 것이고, 더이상 현재 적격 클러스터의 일부로 간주될 수 없다. 그러므로, 입도 카운터가 결정 마름모(444)에서 만기되었을 때, 흐름은, 현재 적격 클러스터가 종료하였고, 따라서 카운트되어야 하는 것을 나타내는 블록(448)으로 진행한다. 블록(448)에서, 클러스터 카운터가 증가되고, 거리 카운터가 c_distance로 재초기화된다. 흐름은 이 후, 거리 문턱 시간이 발생하기를 기다린 후(즉, 거리 카운터가 만기됨), 새로운 제1 이벤트(즉, 다음 적격 클러스터) 검색이 시작되는, 포인트 A를 경유한 도 4의 블록(404)으로 진행한다.

[0030] 도시된 실시예에서, 현재 적격 클러스터가 카운트된 후, 거리 카운터는 c_distance로 재초기화된다(블록(448)에서)는 점에 유의하라. 그러므로, 도시된 실시예에서, 클러스터 카운터는 매번 증가되고, 사이즈 카운터, 입도 카운터, 및 거리 카운터는 0이며, 이 클럭에서는 어떤 이벤트도 발생하지 않는다.

[0031] 도 4 및 도 5의 방법들은 소프트웨어, 하드웨어, 펌웨어, 또는 그의 임의 조합으로 구현될 수 있다. 예를 들어, 일 실시예에서, 도 2의 제어 회로(234)는 내부 카운터들과, 내부 카운터들(238) 및 클러스터 카운터들 1 내지 N을 적절하게 증가, 감소, 및 재설정하도록 요구된 임의 로직을 포함한다. 제어 회로(234)는 또한, 카운

터 제어 레지스터 1 내지 N으로 기록 및 상기로부터 판독할 필요가 있는 임의 로직을 포함할 수 있다. 또한, 상이한 실시예들은 내부 카운터들을 상이하게 구현할 수 있다. 예를 들어, 0으로 카운트 다운하기 보다, 대안적인 실시예는 문턱 값이 만족될 때까지 카운터를 증가시킬 수 있다.

[0032] 도 6은 도 4 및 도 5의 방법들의 보다 나은 이해를 돕는 클러스터링 예(600)를 도시한다. 도 6의 예는 각 클럭에서 주어진 입도 카운터, 사이즈 카운터, 거리 카운터, 및 클러스터 카운터의 값들을 갖는 일련의 클럭들(0 내지 42로 번호 매겨짐)로서 도시된다. 클럭 0에서, 모든 값들이 초기화된다. 입도 카운터, 거리 카운터, 및 클러스터 카운터는 모두 0으로 설정되고, 사이즈 카운터는 c_size로 설정되는데, 도시된 예에서, c_distance는 6개의 클럭 사이클들이고, c_size는 4개의 이벤트들이며, c_granularity는 2개의 클럭 사이클들이다. 처음 몇개의 사이클들에 대해, 클럭 사이클 3까지, 모든 값들이 유지된다(도 4의 결정 마름모(408)의 루프에 대응함). (또한, 거리 카운터는 도 4의 결정 마름모(406)와 블록(414)의 루프를 바이패싱하여 클럭 0에서 0으로 초기화되었다는 점에 유의하라.) 제1 이벤트(602)가 클럭 3에서 검출된다(결정 마름모(408)에 대응함). 그러므로, 입도 카운터가 c_granularity로 초기화되고, 사이즈 카운터가 감소된다(도 4의 블록(410)에 대응함). 예(600)에서 입도 및 사이즈 카운터들의 신규 값들은 제1 이벤트(602) 후 다음 클럭에서 효력을 나타내고, 그러므로 클럭 4에서 입도 카운터는 2로 초기화되고, 사이즈 카운터는 3이라는 점에 유의하라. 상기 다음 클럭(클럭 4)에서, 어떤 이벤트도 검출되지 않고, 따라서 입도 카운터는 각 클럭에서 다시 감소된다(도 4의 결정 마름모(412), 결정 마름모(422), 및 블록(416)에 대응함). 클럭 5에서, 입도 카운터는 만기되지 않았고, 그러므로 다시 감소된다는 점에 유의하라(블록 416). 감소된 값은 다음 클럭(클럭 6)에서 나타난다.

[0033] 그러나, 클럭 6에서, 이벤트(603)가 검출된다(결정 마름모(412)에서부터 블록(420)에 대응함). 이벤트(603)가 검출되었고, 이벤트들(602,603) 간의 거리가 기껏해야 2개의 클럭 사이클들(즉, c_granularity 클럭 사이클들)이었기 때문에, 이벤트(603)는 이벤트(602)와 동일한 클러스터로 간주될 수 있다. 이벤트(603)가 검출되기 때문에, 입도 카운터는 c_granularity로 재초기화되고, 사이즈 카운터가 감소된다(도 4의 블록(420)에 대응함). 그러므로, 다음 클럭(클럭 7)에서, 이 신규 값들이 나타난다. 즉, 클럭 7에서, 사이즈 카운터는 2이고, 입도 카운터는 다시 2이다. 이것은 검출되는 제1 클러스터이기 때문에, 제1 적격 클러스터의 검출 후 거리 카운터가 재설정될 때까지 0으로 남아있다. 이것은 제1 적격 클러스터가 항상 카운트되는 것을 보증한다.

[0034] 클럭 9에서, 제4 이벤트가 검출된다. 사이즈 카운터가 다시 감소되고, 입도 카운터가 c_granularity로 다시 재초기화된다. 다음 클럭(클럭 10)에서, 적격 클러스터(604)가 검출된 것을 나타내면서, 따라서 사이즈 카운터가 만기(0과 동일함)되었음을 알 수 있다. 도 6에서 알 수 있는 바와 같이, 임의의 두개의 인접한 이벤트들 간의 시간은 기껏해야 2개의 클럭 사이클들이기 때문에, 적격 클러스터(604)의 처음 4개의 이벤트들은 사이즈 문턱 값 및 입도 문턱 값을 만족한다. 거리 카운터도 만기되기 때문에(이것은 제1 검출된 적격 클러스터이기 때문임), 적격 클러스터(604)가 카운트될 것이지만, 적격 클러스터(604)의 끝이 검출될 때까지는 카운트되지 않을 것이다. 그러므로, 현재 적격 클러스터의 끝이 검출될 때까지, 흐름은 결정 마름모들(440,444)과 블록(442,446)을 통해 진행한다. 클럭 21에서, 도 6의 빗금친 막대(606)로 표시되는 바와 같이, 현재 적격 클러스터의 끝을 나타내면서 따라서 입도 카운터가 만기된다. 이 지점에서, 입도 카운터, 사이즈 카운터, 및 거리 카운터는 모두 만기된다(즉, 0과 동일함)는 점에 유의하라. 또한, 이 지점에서, 도 6의 클럭 22에서 도시된 바와 같이, 거리 카운터는 c_distance로 재초기화되고, 사이즈 카운터는 c_size로 재초기화되며, 클러스터 카운터는 증가된다(블록(448)에 대응함)(그 값들은 다음 클럭에서 효력을 나타낼 수 있기 때문임). 그 다음 신규의 적격 클러스터에 대한 검색이 시작된다(도 4의 포인트 A에 대응함).

[0035] 클럭 23에서, 이벤트(605)가 발생하더라도, 거리 카운터가 아직 만기되지 않았기 때문에, 제1 이벤트로서 검출되지 않는다. 도시된 실시예에서, 거리 카운터가 만기될 때까지, 어떤 이벤트들도 제1 이벤트로서 검출되지 않는다(도 4의 결정 마름모(406)과 블록(414)의 루프에 대응함). 그러므로, 클럭 28에서의 이벤트(606)까지 제1 이벤트는 검출(결정 마름모(408)에 의해)되지 않는다. 이벤트(605)로 시작하는 이벤트들의 클러스터 동안 이벤트(606)가 발생하더라도, 성능 모니터(104)는 다음 적격 클러스터를 검색하기 전 거리 카운터가 만기될 때까지 기다리기기 때문에, 여전히 제1 이벤트로서 검출된다는 점에 유의하라. 또한, 거리 카운터가 만기 후까지, 사이즈 카운터가 감소되지 않았고, 입도 카운터가 설정되지 않는다는 점에 유의하라.

[0036] 그러므로, 클럭 28에서, 신규의 제1 이벤트(606)가 검출된다. 전술된 바와 같이, 흐름은 클럭 32에서 발생하는 사이즈 카운터가 만기될 때까지, 결정 마름모(412)를 통해 진행한다. 이 지점에서, 또다른 적격 클러스터(608)가 검출되고, 흐름은 적격 클러스터(608)의 끝이 검출되는 도 5의 포인트 C로 진행한다(결정 마름모들(440,444)와 블록들(442,446)에 대응함). 그러므로, 도 6의 빗금친 막대(616)로 표시되는 바와 같이, 클럭 34에서, 사이즈, 입도, 및 거리 카운터들 모두가 만기되고, 어떤 이벤트도 발생하지 않을 때, 적격 클러스터(60

8)의 끝이 검출된다. 또한, 도 6의 클럭 35(그 값들은 다음 클럭에서 효력을 나타내기 때문임)에서 알 수 있는 바와 같이, 이 지점에서, 거리 카운터가 c_distance로 재초기화되고, 사이즈 카운터가 c_size로 재초기화되며, 클러스터 카운터가 증가(블록(448)에 대응함)된다. 이 후, 신규의 적격 클러스터에 대한 검색이 시작된다(도 4의 포인트 A에 대응함).

[0037] c_distance의 만기 후까지 신규의 제1 이벤트가 검출되지 않는다. 그러므로, 빗금친 막대(616) 후, 이벤트(610)로 시작하는 네개의 이벤트들 중 어느 것도 신규의 제1 이벤트로서 검출되지 않는다. 거리 카운터는 계속해서 감소될 것이고, 사이즈 카운터와 입도 카운터들은 그 값들을 유지할 것이다. 그러나, 클럭 41 또는 그 후에서 발생하는 다음 이벤트(도시되지 않음)는 신규의 제1 이벤트로 간주될 것이고, 흐름은 이 후 전술된 바와 같이 진행할 것이다.

[0038] 도 6의 예에서, 적격 클러스터(604)와 적격 클러스터(608)인 두개의 적격 클러스터들이 카운트되었다. 그러므로, 사이즈 문턱 값(c_size)와 입도 문턱 값(c_granularity)에 부가하여 거리 문턱 시간(c_distance)의 사용을 통해, 예(600)에서 오직 두개의 적격 클러스터가 일련의 이벤트들로부터 카운트되었다. 도 6에서 알 수 있는 바와 같이, 사이즈 카운터, 입도 카운터, 및 거리 카운터가 모두 0과 동일하고 이벤트가 발생하지 않을 때(클럭들 21 및 34에서 발생함), 클러스터 카운터가 증가된다. 그러므로, 사용자가 원하는 대로 설정할 수 있는 다양한 문턱 값들의 사용을 통해, 사용자가 프로파일링하고자 하는 분배에 따라, 카운트되는 적격 클러스터들의 수가 변한다. 또한, 데이터 처리 시스템의 성능에 대한 보다 나은 특성 표시를 획득하기 위해서, 부가적인 거리 문턱 시간의 사용은 개선된 프로파일링을 허용한다는 것을 알 수 있다.

[0039] 도시된 실시예에서, 거리 카운터의 만기 후까지, 후속하는 적격 클러스터의 검출이 시작되지 않았다는 점에 유의하라. 즉, 거리 문턱 시간의 만료 후에만 적격 클러스터가 검색되었다. 그러나, 대안적인 방법에서, 거리 카운터가 감소되는 시간 동안에 적격 클러스터들이 검출될 수 있다. 즉, 거리 카운터가 만기되기를 기다리는 동안 발생하는 이벤트들을 무시하기 보다, 결정 마름모(408)의 "검출된 제1 이벤트"로서 검출될 수 있다. 그러나, 적격 클러스터를 검출하는 이 대안적인 방법에서, 클러스터 카운터에 의해 카운트되기 위해, 그것이 이전의 적격 클러스터로부터 적어도 거리 문턱 시간 만큼 떨어져 발생되는지를 결정할 필요가 있다. 즉, 이 대안적인 방법을 사용하여 결정된 모든 적격 클러스터들은 클러스터 카운터에 의해 카운트될 수 없다. 이 대안적인 방법을 사용하는 일 실시예에서, 적어도 거리 문턱 시간이 이전 카운트된 적격 클러스터 이후에 발생되었다면, 적격 클러스터들이 카운트된다. 이 대안적인 방법을 사용하는 대안적인 실시예에서, 클러스터 카운터에 의해 카운트되었는지의 여부에 상관 없이, 적어도 거리 문턱 시간이 이전 적격 클러스터 이후에 발생되었다면, 적격 클러스터들이 카운트된다. 그러므로, 당업자는, 데이터 처리 시스템의 성능을 원하는 대로 프로파일링하기 위해 여기서 기술된 다양한 문턱 값들(즉, 사이즈 문턱 값, 입도 문턱 값, 및 거리 문턱 시간)을 사용하는 다수의 상이한 방법들이 존재한다는 것을 알 수 있다.

[0040] 예를 들어, 도 6의 예(600)에 대한 이전 단락에서 기술된 방법을 사용하여, 이전 적격 클러스터 후 적어도 거리 문턱 시간에 발생하는 적격 클러스터들만이 카운트되었는지의 여부에 상관없이 상이한 프로파일링 결과를 유발할 것이다. 도 6을 다시 참조하여, 이 대안적인 방법을 사용하여, 적격 클러스터(604), 적격 클러스터(618), 및 적격 클러스터(620)인 세개의 적격 클러스터들이 검출될 것이다. 이 클러스터들 각각은 사이즈 및 입도 문턱 값들 모두를 만족하기 때문에, 적격 클러스터들로서 검출된다. 또한, 이전 클러스터의 끝과 후속하는 클러스터들의 시작 간의 거리가 입도 문턱 값보다 크고, 따라서 동일한 적격 클러스터의 일부로 간주될 수 없기 때문에, 이 클러스터들 각각은 별개의 적격 클러스터들로 간주된다. 예를 들어, 클럭 18(적격 클러스터(604)의 마지막 이벤트)과 클럭 23(적격 클러스터(618)의 시작) 간의 시간이 2개의 클럭 사이클들보다 크다. 그러나, 이 세개의 검출된 적격 클러스터들 중, 적격 클러스터들 중 오직 하나(적격 클러스터(604))가 카운트된다. 적격 클러스터(604,618)간의 거리가 거리 시간 문턱 값을 만족하지 않기 때문에, 적격 클러스터(618)는 카운트되지 않는다. 유사하게, 적격 클러스터(618)(카운트되지 않았더라도 이전의 적격 클러스터)과 적격 클러스터(620)간의 거리가 거리 시간 문턱 값을 만족하지 않기 때문에, 적격 클러스터(620)가 카운트되지 않는다. (이 대안적인 방법의 실시예에서, 그것이 카운트되는지의 여부에 상관없이, 각각의 적격 클러스터의 끝을 검출한 후, 거리 카운터가 c_distance로 재초기화된다는 점에 유의하라.) 그러므로, 도 4 및 도 5의 방법에 비해, 이 대안적인 방법은 상이한 적격 클러스터들을 검출하고, 이 검출된 적격 클러스터들의 서브셋트를 단지 카운트할 수 있다. 도 4 및 도 5의 방법은 항상 다음의 적격 클러스터를 검출하기 전에, 만기되는 거리 카운터를 기다려야 하기 때문에, 도 4 및 도 5의 방법은 단지 실제로 카운트될 적격 클러스터들을 검출한다. 또한, 이 대안적인 방법의 실시예는 적격 클러스터(618)를 검출하지만, 도 4 및 도 5의 방법이, 제1 이벤트가 적격 클러스터의 중간에서 발생하게 하기 때문에, 도 4 및 도 5의 방법은 단지 적격 클러스터(618)(즉, 적격 클러스터(608))의

일부를 검출한다.

[0041] 모든 적격 클러스터들이 검출되지만 적어도 거리 문턱 시간이 이전에 카운트된 적격 클러스터 이후에 발생하는 경우에만 카운트되는 대안적인 방법을 사용하는 또다른 예에서, 상이한 프로파일링 결과가 발생할 수 있다. 예를 들어, 도 6을 다시 참조하여, 적격 클러스터(604), 적격 클러스터(618), 및 적격 클러스터(620)인 동일한 세 개의 적격 클러스터들이 검출될 것이다. 그러나, 이 예에서, 적격 클러스터들 중 두개인 적격 클러스터(604)와 적격 클러스터(620)가 카운트된다. 각각의 적격 클러스터가 거리 문턱 시간을 만족하는지에 대한 결정은 상이하게 결정되기 때문에, 두개의 적격 클러스터들이 카운트된다. 예를 들어, 이전 두 단락들에서 기술된 대안적인 방법으로, 적격 클러스터(604)는 제1 검출된 적격 클러스터이기 때문에 카운트된다. 또한, 적격 클러스터(618)가 카운트되는지를 결정하기 위해, 카운트된 적격 클러스터(604)와 검출된 적격 클러스터(618) 간의 클러스터 거리가 거리 문턱 시간을 만족하는지가 결정된다. 그러나, 도 6의 예에서는 만족하지 않고, 따라서 카운트되지 않는다. 그러나, 이전 두 단락들에서 기술된 대안적인 방법과 달리, 적격 클러스터(620)가 카운트되는지를 결정하기 위해, 검출된 적격 클러스터(620)와 카운트된 적격 클러스터(604)간의 클러스터 거리가 거리 문턱 시간을 만족하는지가 결정된다. 도 6의 예에서는 만족하기 때문에, 그것은 또한 카운트된 적격 클러스터이다. 이전 실시예는 카운트되지 않았던 적격 클러스터(618)와 검출된 적격 클러스터(620) 간의 클러스터 거리를 사용하기 때문에, 이것은 이전 두 단락들에서 기술된 대안적인 방법의 이전 실시예들과 상이하다. (이 대안적인 방법의 대안적인 실시예에서, 카운트되지 않은 적격 클러스터의 끝을 검출한 후, 거리 카운터가 c_distance로 재초기화되지 않는다는 점에 유의하라. 대신에, 카운트된 적격 클러스터의 끝을 검출한 후에만, c_distance로 재초기화된다.)

[0042] 당업자는 거리 카운터를 사용하여 적격 클러스터들을 카운팅하는 이 대안적인 방법들을 적응시키도록 도 4 및 도 5의 흐름들을 변경할 수 있다는 점에 유의하라. 또한, 당업자는 거리 문턱 시간의 다양한 사용들을 통해, 프로파일링하고자 하는 분배에 따라, 카운트되는 적격 클러스터들의 수가 어떻게 바뀔 수 있는지 알 수 있다. 그러므로, 부가적인 거리 문턱 시간의 사용은 데이터 처리 시스템의 성능에 대한 개선된 특성 표시를 획득하는 데 있어서 증가된 유연성을 허용한다는 점을 알 수 있다.

[0043] 전술된 명세서에서, 본 발명은 특정한 실시예들을 참조하여 기술되었다. 그러나, 당업자는 이하의 청구범위에서 설명된 바와 같은 본 발명의 범위를 벗어나지 않고 다양한 변경들과 변화들이 형성될 수 있다는 점을 알 수 있다. 예를 들어, 여기서 기술된 방법들은 소프트웨어, 하드웨어, 펌웨어, 또는 그 임의 조합으로 구현될 수 있다. 예를 들어, 여기서 가르쳐진 방법들이나 그 방법들 일부는 하나 이상의 컴퓨터 하드 디스크들, 플로피 디스크들, 3.5" 디스크들, 컴퓨터 저장 테잎들, 자기 드럼들, SRAM(static random access memory) 셀들, DRAM(dynamic random access memory) 셀들, 전기적 삭제가능(EEPROM, EPROM, 플래쉬) 셀들, 비휘발성 셀들, 강유전 또는 강자성 메모리, 컴팩트 디스크들(CD), 레이저 디스크들, 광학 디스크들, 및 임의의 유사한 컴퓨터 판독가능한 매체 상에 소프트웨어로서 구현될 수 있다. 또한, 블록 다이어그램들은 도시된 것과 상이한 블록들을 포함할 수 있고, 몇개의 블록들을 가지거나 상이하게 배열될 수 있다. 또한, 흐름도들도 상이하게 배열된 몇개의 단계들을 포함하여 상이하게 배열될 수 있거나, 다수의 단계들로 분할될 수 있는 단계들이나 서로 동시에 수행될 수 있는 단계들을 가질 수 있다. 따라서, 명세서 및 도면들은 제한적이기 보다는 예시적인 것으로 간주되고, 모든 변경들은 본 발명의 범위 내로 포함되도록 의도된다.

[0044] 이점, 다른 장점들, 및 문제점들에 대한 해결책들은 특정한 실시예들에 관해 전술되었다. 이점, 장점들, 문제들에 대한 해결책들, 및 임의의 이점, 장점, 또는 해결책을 발생 또는 더 표명되게 할 수 있는 요소들은 중요하거나 요구되거나 필수적인 임의 또는 모든 청구항들의 특징이나 요소로서 해석되지 않는다. 여기서 사용된 바와 같이, 용어 "포함하다", "포함하는", 또는 그 임의의 다른 변형은, 요소들의 리스트를 포함하는 처리, 방법, 물건, 또는 장치가 그 요소들만을 포함하지 않고 상기 처리, 방법, 물건, 또는 장치에 대해 명백히 리스트되지 않거나 고유한 다른 요소들을 포함할 수 있도록, 배타적이지 않은 포함을 커버하도록 의도된다.

도면의 간단한 설명

[0004] 도 1은 본 발명의 일 실시예에 따른 데이터 처리 시스템을 블록 다이어그램 형태로 도시하는 도면.

[0005] 도 2는 본 발명의 일 실시예에 따라 도 1의 데이터 처리 시스템의 성능 모니터를 블록 다이어그램 형태로 도시하는 도면.

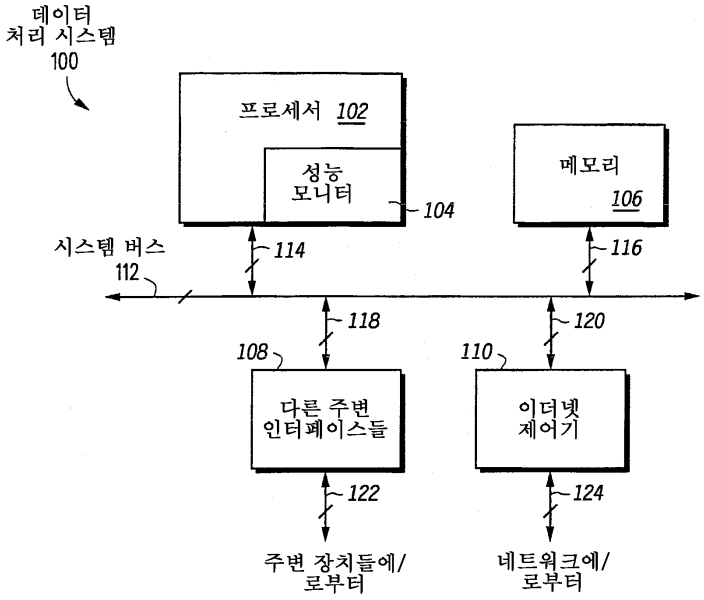
[0006] 도 3은 본 발명의 일 실시예에 따른 이벤트 클러스터링의 예를 도시하는 도면.

[0007] 도 4 및 도 5는 본 발명의 일 실시예에 따라 도 2의 성능 모니터의 동작을 도시하는 도면.

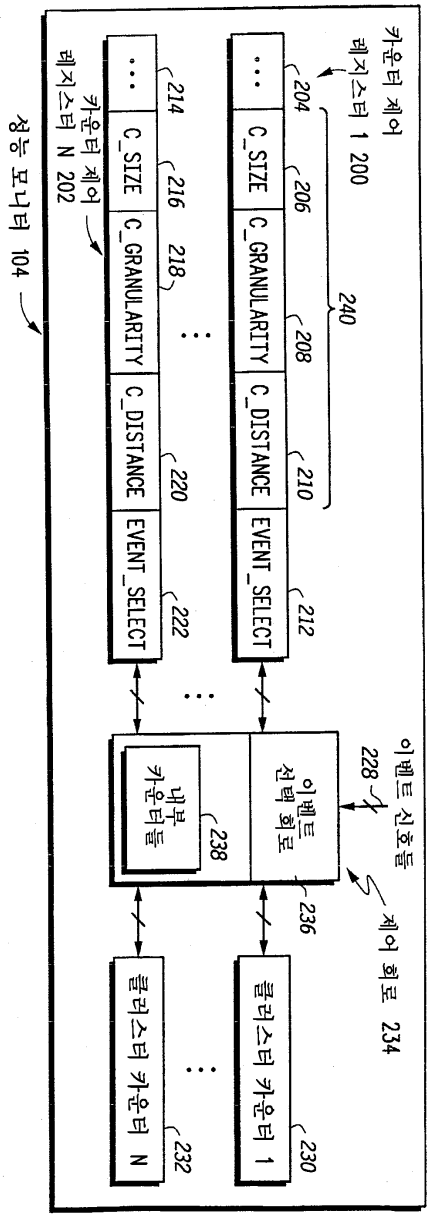
[0008] 도 6은 본 발명의 일 실시예에 따른 이벤트 클러스터링의 또다른 예를 도시하는 도면.

도면

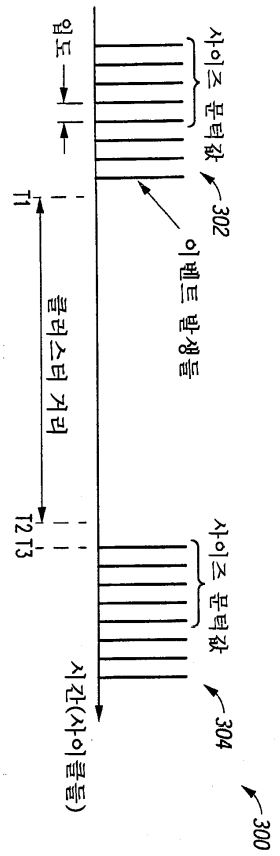
도면1



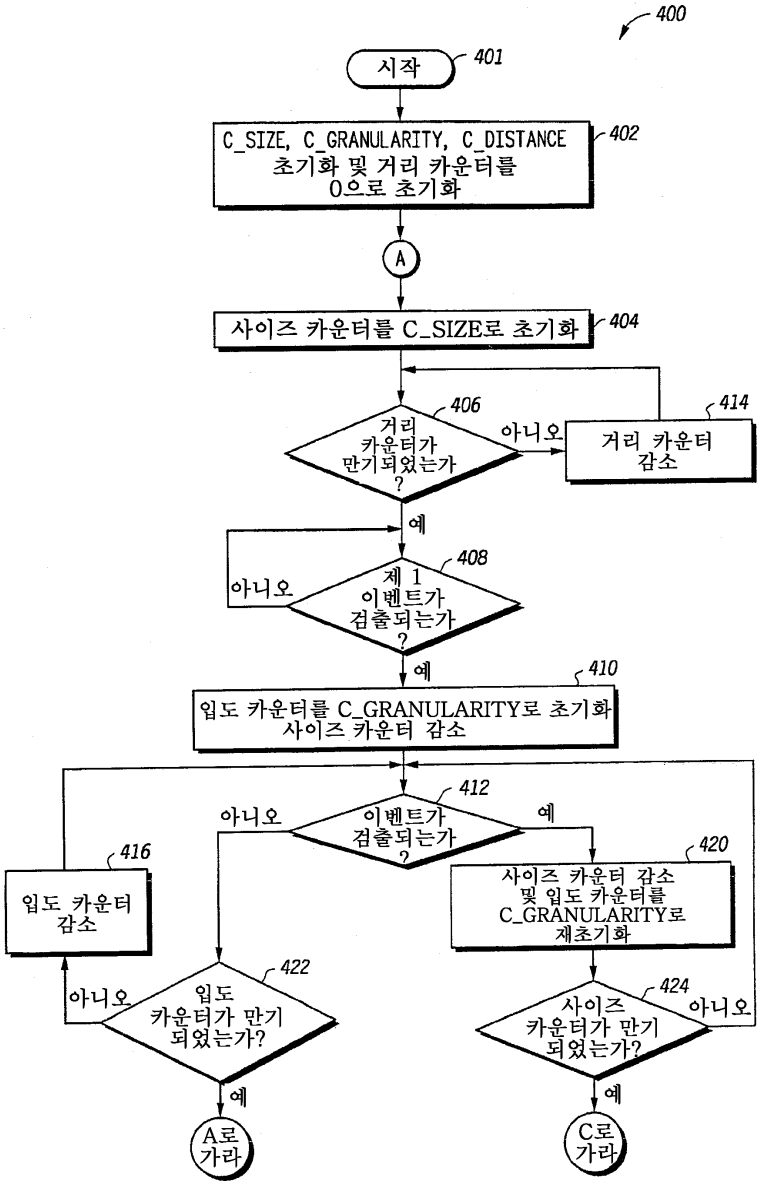
도면2


성능 모니터 104
카운터 제어 레지스터 1 200
240
204
206
208
210
212
C_SIZE
C_GRANULARITY
C_DISTANCE
EVENT_SELECT
214
216
218
220
222
카운터 제어 레지스터 N 202
C_SIZE
C_GRANULARITY
C_DISTANCE
EVENT_SELECT
이벤트 신호들
228
이벤트 선택 회로
236
제어 회로 234
내부 카운터들
238
클러스터 카운터 1
230
클러스터 카운터 N
232

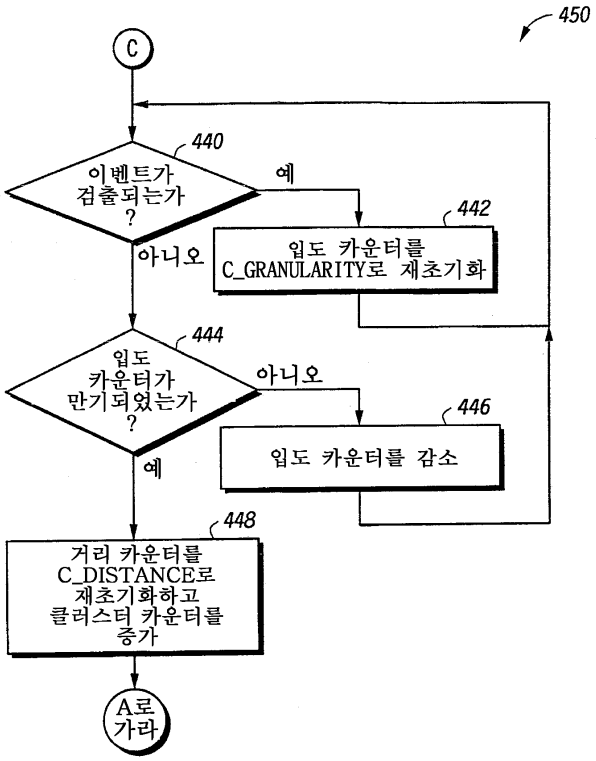
도면3



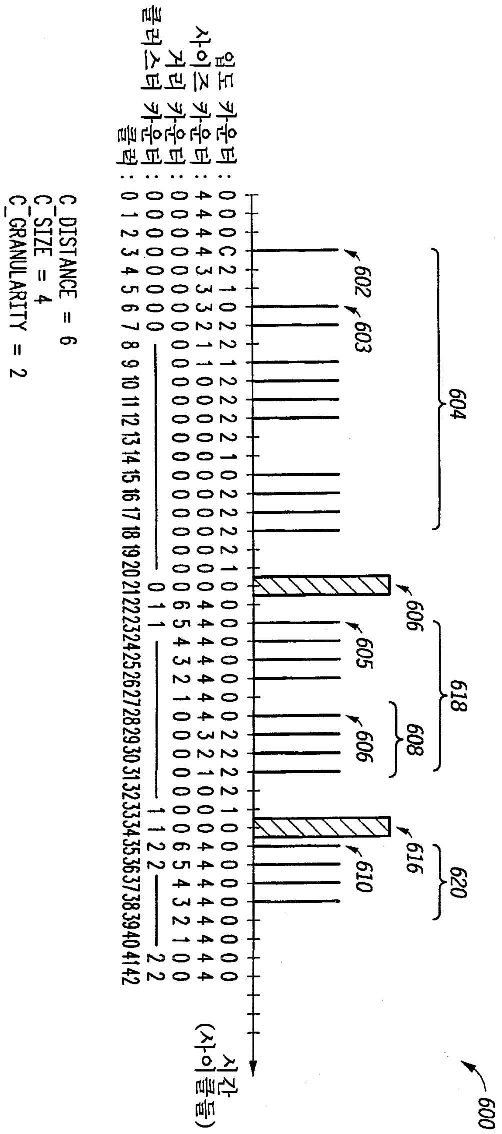
도면4


400
시작
401
C_SIZE, C_GRANULARITY, C_DISTANCE
초기화 및 거리 카운터를
0으로 초기화
402
A
사이즈 카운터를 C_SIZE로 초기화
404
거리
카운터가
만기되었는가
?
406
아니오
거리 카운터
감소
414
예
제 1
이벤트가
검출되는가
?
408
아니오
예
입도 카운터를 C_GRANULARITY로 초기화
사이즈 카운터 감소
410
이벤트가
검출되는가
?
412
아니오
예
입도 카운터
감소
416
사이즈 카운터 감소
및 입도 카운터를
C_GRANULARITY로
재초기화
420
입도
카운터가 만기
되었는가?
422
아니오
사이즈
카운터가 만기
되었는가?
424
아니오
예
예
A로
가라
C로
가라

도면5


450
C
440
이벤트가 검출되는가 ?
예
442
입도 카운터를 C_GRANULARITY로 재초기화
아니오
444
입도 카운터가 만기되었는가 ?
아니오
446
입도 카운터를 감소
예
448
거리 카운터를 C_DISTANCE로 재초기화하고 클러스터 카운터를 증가
A로 가라

도면6



밀도 카운터:
사이즈 카운터:
거리 카운터:
클러스터 카운터:
클럭:
C_DISTANCE = 6
C_SIZE = 4
C_GRANULARITY = 2
시간 (사이클들)
602
603
604
605
606
608
610
616
618
620
600