

Nicolas Vargas

Search history

Search PN=(WO2016164678) (Results 1)

1:

2: [SP]: 1. a system comprising: a processor coupled to a memory, the processor configured to execute computer-executable instructions stored in the memory that when executed perform acts comprising: receiving an event-processing query, as part of a subscription, for continuous evaluation until the subscription is canceled andrewriting the event-processing query automatically in a pageable form, wherein the pageable form enables swapping of at least a portion of the event-processing query into and out of memory based on one or more injected triggers specified in terms of at least one characteristic of one or more of the event-processing query or an event sequence over which the event-processing query operates. 2. the system of claim 1 further comprises injecting a page-in trigger into the pageable form that initiates loading of the at least the portion of the event-processing query into the memory. 3. the system of claim 2, the page-in trigger is generated based on historical data related to execution of the event-processing query. 4. the system of claim 1 further comprises injecting a page-out trigger into the pageable form that initiates removal of the at least the portion of the event-processing query from the memory. 5. the system of claim 4, the page-out trigger is generated based on historical data related to execution of the event-processing query. 6. the system of claim 1 further comprises gathering data regarding execution of the event-processing query. 7. a method comprising: employing at least one processor configured to execute computer-executable instructions stored in a memory to perform the following acts: receiving an event-processing query, as part of a subscription, for continuous evaluation until the subscription is canceled andgenerating a pageable query automatically based on the event-processing query, wherein the pageable query supports swapping of at least a portion of the event-processing query into and out of the memory based on at least one characteristic of one or more of the event-processing query or an event sequence over which the event-processing query operates. 8. the method of claim 7 further comprises injecting a page-in trigger into the pageable query that initiates loading of the at least the portion of the event-processing query into the memory. 9. the method of claim 7 further comprises injecting a page-out trigger into the pageable query that initiates removing the at least the portion of the event-processing query from the memory. 10. the method of claim 7 further comprises inserting a mechanism to defer loading of the at least the portion of the event-processing query into memory until triggered. 11. the method of claim 7 further comprises inserting a mechanism to collect data related to execution of the event-processing query. 12. the method of claim 11 further comprises employing machine learning with respect to the data to infer at least one of a page-in trigger or page-out trigger. 13. the method of claim 7 further comprises inserting a mechanism to save state of the at least the portion of the event-processing query in connection with swapping the event-processing query out of the memory. 14. the method of claim 7 further comprises inserting a mechanism to recover state of the at least the portion of the event-processing query in connection with swapping the event-processing query into the memory. 15. the method of claim 7 further comprises inserting a mechanism to swap out of the memory a subset of the at least the portion of the event-processing query. 16. a computer-readable storage medium having instructions stored thereon that enable at least one processor to perform a method upon execution of the instructions, the method comprising: detecting a page-in trigger of an event-processing query specified in a pageable form that enables swapping of at least a portion of the event-processing query into and out of memory, wherein the page-in trigger is based on at least one characteristic of one or more of the event-processing query or an event sequence over which the event-processing query operates andloading at least a portion of an event-processing query located outside of memory into the memory in response to detecting the page-in trigger. 17. the computer-readable storage medium of claim 16, the method of loading the event-processing query comprises: locating a data representation of the at least the portion of the event-processing query based on an identifier andcompiling the data representation to executable code. 18. the computer-readable storage medium of claim 16, the method further comprises: detecting a page-out trigger with respect to the event sequence andunloading the at least the portion of the event-processing query from memory in response to detecting the page-out trigger. 19. the computer-readable storage medium of claim 18, the method of unloading the event-processing query comprises initiating a state capture operation that captures current state of the event-processing query. 20. the computer-readable storage medium of claim 18, the method of unloading the event-processing query comprises unloading solely a subset of the at least the portion of the event-processing query. (Results 0)

Search [SP]: event processing system paging a pageable query can be generated based on an event-processing query. the pageable query is a form of the event-processing query that supports swapping the event-processing query into and out of memory. for instance, page-in and page-out triggers can be inserted. after detection of a page-in trigger, the event-processing query can be loaded into the memory, and after detection of a page-out trigger, the event-processing query can be unloaded from memory. (Results 26)

Search 3 and paging (Results 6)

4:

Search 4 and PD<20150410 (Results 5)

5:

Search 5 and quer* (Results 2)

6:

Search SFT=(paging and quer*) and PD<20150410 (Results 10352)

7:

Search SFT=(paging and quer* and rewrit*) and PD<20150410 (Results 848)

8:

Search (Results)

9:

Search (Results)

10:

Search SFT=(paging and quer* and rewrit* and processor and memory and event*) and PD<20150410 (Results 668)

11:

Search STAC=(paging and quer* and rewrit* and processor and memory and event*) and PD<20150410 (Results 0)

12:

Search STAC=(paging and quer* and rewrit* and event*) and PD<20150410 (Results 0)

13:

Search STAC=(paging and quer* and rewrit*) and PD<20150410 (Results 1)

14:

Search STAC=(paging and quer*) and PD<20150410 (Results 564)

15:

Search 15 and rewrit* (Results 25)

16:

Search 16 and event* (Results 17)

17:

Search 17 and processor* and memory (Results 16)

18:

Search 18 and query (Results 15)

19:

Search 19 and data (Results 15)

20:

Search 15 and (rewrit* w10 query) (Results 1)

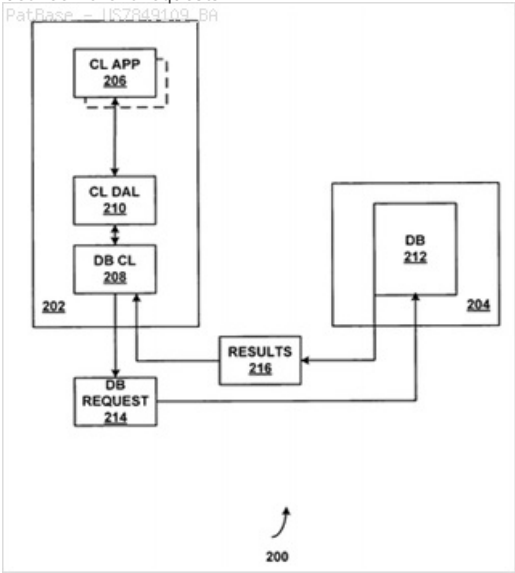
21:

Title: Use of server processing of temporary results for stateless **paging** over arbitrary user **queries**

Abstract: A **query** processor's capabilities may be employed to identify a range rows resulting from a user **query** to be returned. A server cursor may be declared, opened, fetched from and closed in a single batched request in order to avoid keeping state on a database server between requests. A single block fetch may be used within the batch to retrieve a requisite number of rows from a specified ordinal position. Where server cursors are not available, an intermediate table may be used to hold a subset of the rows resulting from the user **query**, and row numbers may be added to facilitate selecting the range of rows from that intermediate table. Intelligence within a **query** processor may be leveraged to support identifying the range of rows to be returned without maintaining state between client requests.

Classifications:

International (IPC 8-9): G06F17/30 (Advanced/Invention)
CPC: G06F17/30501
European: G06F17/30S4P4P7
US: 707/809



Family:

Publication number	Publication date	Application number	Application date
US7849109 BA	20101207	US20040909093	20040730

Priority:

US20040909093 20040730

Probable Assignee: MICROSOFT TECHNOLOGY LICENSING LLC
Assignee(s): (std): MICROSOFT CORP
Assignee(s): MICROSOFT TECHNOLOGY LICENSING LLC
Inventor(s): (std): DIVRINGI LALE R ; PIZZO MICHAEL J
Agent(s): WOODCOCK WASHBURN LLP