Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 10 September 2019

Case Number: T 0818/16 - 3.5.07

Application Number: 07853813.9

Publication Number: 2074505

IPC: G06F17/30

Language of the proceedings: EN

Title of invention:
Time series search engine

Applicant:
Splunk Inc.

Headword:
Time series search engine/SPLUNK

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - (no)

Decisions cited:
G 0003/08, T 0641/00, T 0154/04

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0818/16 - 3.5.07

D E C I S I O N
of Technical Board of Appeal 3.5.07
of 10 September 2019

Appellant: (Applicant)	Splunk Inc. 270 Brannan Street San Francisco, CA 94107 (US)
Representative:	Dendorfer, Claus Dendorfer & Herrmann Patentanwälte Partnerschaft mbB Neuhauser Straße 47 80331 München (DE)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 24 November 2015 refusing European patent application No. 07853813.9 pursuant to Article 97(2) EPC**

Composition of the Board:

Chairman	R. Moufang
Members:	M. Jaedicke
	C. Barel-Faucheux

Summary of Facts and Submissions

I. The applicant (appellant) appealed against the decision of the Examining Division refusing European patent application No. 07853813.9, filed as international application PCT/US2007/080616 and published as WO 2008/043082 A2. The application claims a priority date of 5 October 2006.

II. The documents cited in the contested decision included:
D11: US 2005/114707 A1, published on 26 May 2005

III. The Examining Division decided that the subject-matter of claim 1 of the sole request on file lacked inventive step in view of document D11. The Examining Division considered that the differentiating features did not make any technical contribution to the prior art.

IV. In its statement of grounds of appeal, the appellant requested that the decision be set aside and that a patent be granted on the basis of the main request or one of the three auxiliary requests, all submitted with the grounds of appeal.

V. In a communication under Article 15(1) RPBA accompanying the summons to oral proceedings, the Board expressed its provisional opinion *inter alia* that the subject-matter of claim 1 of the main request lacked inventive step in view of document D11 and the common general knowledge of the skilled person. As evidence of selected aspects of common general knowledge, the Board cited the following documents:

D12: G. Graefe, "Query Evaluation Techniques for Large Databases", ACM Computing Surveys, vol. 25, no. 2, pp. 73-170, June 1993

D13: S. Chaudhuri, "An Overview of Query Optimization in Relational Systems", Proceedings of the 1998 ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 98), Seattle, Washington, USA, 1-4 June, 1998, pp. 34-43, published on 1 June 1998

VI. In a letter of 11 August 2019, the appellant submitted arguments together with a new main request and new first to third auxiliary requests and, as a matter of precaution, maintained its previous requests as new fourth to seventh auxiliary requests.

VII. Oral proceedings were held as scheduled and the appellant was heard on relevant issues. In the course of the oral proceedings, the appellant filed a new sole request (labelled "New Main Request") which replaced all previously pending requests. At the end of the oral proceedings, the chairman pronounced the Board's decision.

VIII. The appellant's final request was that the decision under appeal be set aside and that a patent be granted on the basis of the sole request filed in the oral proceedings.

IX. Claim 1 of the sole request reads as follows:

"A computer-implemented method for searching data in a time series search engine, comprising:

receiving streams of different types of machine data generated by different types of machines, wherein different types of machine data are in different formats;

executing a time stamping process on a stream of machine data by:

classifying a collection of machine data from the stream into a domain, the domain being indicative of a source of machine data;

applying aggregation rules corresponding to the domain for the classified machine data to organize the classified machine data into a plurality of events by detecting the beginning and ending boundaries for each event;

determining, for each event of the plurality of events, a time stamp based on the domain by iterating over potential time stamp format patterns from an ordered list; and

time stamping each event of the plurality of events with its determined time stamp to create a plurality of time stamped events, wherein each time stamped event of the plurality of time stamped events includes a respective portion of the machine data, and each time stamp is normalized to a common offset;

executing an indexing process to create time bucketed indices based on the time stamps, wherein each time bucket is defined to correspond to a certain time period according to a bucketing policy, and indexing the time stamped events includes assigning each time stamped event to a time bucket from amongst a plurality of time buckets instantiated in random access computer memory, wherein the assignment is based on the time stamp for the time stamped event, and wherein the bucketing policy enforces that buckets (i) do not overlap and (ii) cover all possible incoming time stamps; and

upon receiving a time series search request that requires search results to be sorted in reverse chronological order, generating sub-searches targeted at individual time buckets, querying time buckets until a number of results specified in the search request are

retrieved, and merging the results of the sub-searches into a result set organized by time according to the reverse chronological sort order for the result set, wherein the sub-search for the most recent time bucket is issued first."

In view of the outcome of the appeal, the text of the other claims need not be given.

X. The appellant's arguments where relevant to the decision are discussed in detail below.

Reasons for the Decision

1. *Admissibility of appeal*

The appeal complies with the provisions referred to in Rule 101 EPC and is therefore admissible.

The invention

2. The application relates to time series data organisation, search and retrieval. Time series data are sequences of time-stamped records occurring in one or more usually continuous streams, representing some type of activity made up of discrete events such as information processing logs, market transactions and sensor data from real-time monitors (supply chains, military operation networks or security systems). The ability to index, search and present relevant search results is important for understanding and working with systems emitting large quantities of time series data (see the description as published, paragraphs [0002] and [0003]).

According to the application, existing large scale search engines (e.g. Google and Yahoo web search) are designed to address the needs of less time-sensitive types of data and are built on the assumption that data only needs to be stored in one state in the index repository, for example URLs in a web search index, records in a customer database, or documents as part of a file system. Searches for information are generally based on keywords (paragraph [0004]).

Compared to full text search engines, which organise their indices so that retrieving documents with the highest relevance scores is most efficient, an engine for searching time series data preferably would organise the index so that access to various time ranges, including less recent time ranges, is efficient. Indexing time series data is further complicated because the data can be collected from multiple, different sources asynchronously and out of order (paragraphs [0006] and [0007]).

The application proposes a time series search engine (TSSE) for the indexing, searching and retrieval of time series data. One aspect of such TSSEs is the use of time as a primary mechanism for indexing, searching and/or presenting of search results (paragraph [0014]).

A computer-implemented method for time searching data includes the following steps (paragraphs [0016] to [0018]): time series data streams are received. One example of time series data streams includes server logs and other types of machine data (i.e. data generated by machines). The time series data streams are time stamped to create time-stamped events. Time stamping the time series data streams includes

aggregating the time series data streams into events, classifying the events by domain and time stamping the events. The time-stamped events are time indexed to create time bucketed indices by assigning the time-stamped events to time buckets according to their time stamps. The creation of time bucket indices facilitates the execution of time series searches. In one approach, a time series search request is divided into different sub-searches for the affected time buckets, with each sub-search executed across the corresponding time bucket index (paragraphs [0018] to [0019]).

Admission

3. Since the current set of claims was a response to rather formal objections raised by the Board in the oral proceedings and could be dealt with without adjourning the oral proceedings, the Board admitted it into the appeal proceedings.

The appellant's request

4. Claim 1 relates to a "computer-implemented method for searching data in a time series search engine". The method comprises the following steps.

A receiving streams of different types of machine data generated by different types of machines, wherein different types of machine data are in different formats

B executing a time stamping process on a stream of machine data by:

B1 classifying a collection of machine data from the stream into a domain, the domain being indicative of a source of machine data

B2 applying aggregation rules corresponding to the domain for the classified machine data to organise the classified machine data into a plurality of events by detecting the beginning and ending boundaries for each event

B3 determining, for each event of the plurality of events, a time stamp based on the domain by iterating over potential time stamp format patterns from an ordered list

B4 time stamping each event of the plurality of events with its determined time stamp to create a plurality of time-stamped events, wherein each time-stamped event of the plurality of time-stamped events includes a respective portion of the machine data, and each time stamp is normalised to a common offset

C executing an indexing process to create time bucketed indices based on the time stamps, wherein each time bucket is defined to correspond to a certain time period according to a bucketing policy, and indexing the time-stamped events includes assigning each time-stamped event to a time bucket from amongst a plurality of time buckets instantiated in random access computer memory, wherein the assignment is based on the time stamp for the time-stamped event, and wherein the bucketing policy enforces that buckets (i) do not overlap and (ii) cover all possible incoming time stamps

D upon receiving a time series search request that requires search results to be sorted in reverse chronological order, generating sub-searches targeted at individual time buckets, querying time buckets until a number of results specified in the search request are retrieved, and merging the results of the sub-searches into a result set

organised by time according to the reverse chronological sort order for the result set, wherein the sub-search for the most recent time bucket is issued first

Inventive step

5. The Examining Division considered document D11 as a suitable starting point for assessing inventive step and this has not been contested by the appellant (see statement of grounds of appeal, section A-V.1).

5.1 Document D11 discloses in paragraph [0019] that raw log data is received by a raw log server, stored in complete form in a database and sent to a networked log data analyzer for parsing, summarising and routine reporting. The raw log data may be received from the log-producing network devices on the same local area network as the raw log server and from a log data analyzer at a remote location on a different network over a wide area network (D11, Figure 1).

Examples of log-producing devices are routers and firewalls (D11, paragraph [0027]). Upon receipt of the raw log data, the raw log server may insert the text string comprising the raw log data into a database together with identifying and/or indexing information. One example of a database is MySQL, a relational database management system. The log data may be stored together with the identity of the log-producing device and a date and time stamp. The time stamps may represent the local time and the time zone of the log-producing device (paragraph [0028]).

Raw log data streams from the local log-producing devices and the compressed, encrypted data streams from

remote locations are merged into a single, sequentially ordered database table (paragraph [0037]). Raw log data from one or more log-producing devices are collected and stored in a buffer, which may be in the random access memory of a processor-based system (paragraphs [0038] and [0043]).

In view of the above, the Board considers that D11 discloses a computer-implemented method for searching data in a time series search engine, the expression "time series" being interpreted broadly in view of the description, paragraph [0003]. As D11 discloses receiving streams of raw log data from routers and firewalls and other devices (see also D11, paragraphs [0003] and [0027]) and as it is implicit that these devices use different formats, D11 discloses step A of claim 1. Since the method disclosed in D11 stores log data together with time stamps (paragraphs [0028] and [0043]), it discloses step B of claim 1.

Moreover, D11 discloses that a header with a device identifier is added to the received log data (paragraphs [0028] and [0043]) and thus that a collection of machine data is classified into a domain indicative of a source of machine data (cf. feature B1).

D11 also discloses parsing and summarising log data (paragraph [0049]). According to document D11, paragraph [0050], the parser parses the received raw log data to extract fields based on a log data message type, and generates Structured Query Language (SQL) statements from the extracted fields.

5.2 The Board agrees with the appellant that D11 does not disclose features B2, B3, B4, C and D.

Features B2, B3 and B4 aim to provide event data with normalised time stamps. Features C and D interact to process time series search requests by means of the time bucket indices. The Board sees no synergistic effect of steps B2 to B4 with steps C and D, but rather considers that their overall result corresponds to an aggregation of these steps.

5.3 The provision of events as data that can be analysed is a non-technical requirement that reflects the information needed by a data analyst. According to the established case law of the boards of appeal, when assessing inventive step in accordance with the problem/solution approach, an aim to be achieved in a non-technical field may legitimately appear in the formulation of the problem as part of the framework of the technical problem to be solved as a constraint that has to be met (see decisions T 641/00, OJ EPO 2003, 352; T 154/04, OJ EPO 2008, 46). Hence, steps B2 to B4 solve the problem of how to implement the conversion of the classified machine data into event data that can be analysed with respect to time.

As to step B2, D11 discloses parsing machine data in paragraph [0050]. According to the description of the present application (paragraph [0046]), an example of an aggregation rule for detecting beginning and ending boundaries of events consists of detecting line breaks. The Board is aware that the wording of step B2 is rather broad and that the event boundaries may be defined based on non-technical considerations or at least not based on further technical considerations (see opinion G 3/08, OJ EPO 2011, 10, reasons 13.5.1). However, in any case, on the relevant date the skilled person would have extended the parser of D11 with rules

to detect event boundaries such as line breaks in the machine data without exercising inventive skill.

As to steps B3 and B4, D11 (see paragraphs [0028] and [0037]) discloses that the time stamps are received in different formats and that the data is stored in a sequentially ordered table. Moreover, D11 (paragraph [0046]) discloses that the system performance can be improved if the data is sorted (e.g. in chronological order) prior to insertion into the database. In view of this, it was obvious for a skilled person to store the database table in the sort order of the data to be inserted, i.e. in chronologically sorted order. Moreover, the skilled person would consider providing some kind of normalisation of the time stamps, such as normalisation to a common offset, as they are received in different formats. The application itself mentions the well-known Unix epoch as a common offset (description, paragraph [0049]). Hence, the skilled person would have considered using such a well-known common offset for normalisation.

5.4 In its statement of grounds of appeal, the appellant submitted that the effect of the aggregation rules was that events that ranged anywhere from one line to hundreds of lines were logically grouped together by aggregation rules (description, paragraph [0044]). However, as explained above, the Board considers that on the priority date the grouping of such events by detecting line breaks or other delimiters would have been a matter of routine for the skilled person.

Moreover, the appellant argued that the use of raw event data (i.e. data that has not been summarised or made to fit into a predefined schema) provided more query flexibility since fields not identified in a

database schema could be queried and important fields need not be identified before data is collected. The Board does however not see any basis for the alleged effect, as claim 1 does not specify that any raw data is stored or used for searching. Rather the received machine data is transformed into event data with normalised time stamps. Hence, the Board is not persuaded by the appellant's arguments.

It follows that the skilled person could and would arrive at steps B2 to B4 of claim 1 without exercising inventive skill.

According to step C of the method of claim 1, the time stamps are used to assign the events to time buckets instantiated in random access computer memory. The time buckets divide the time period covering all possible incoming time stamps into non-overlapping time periods. In subsequent step D, the time buckets are then used to process search requests, wherein a search request specifies the number of results to be retrieved and the search results are required to be sorted in reverse chronological order. For processing such search requests, the search engine generates sub-searches which are each targeted at an individual time bucket (see description, paragraph [0076]) and issues the sub-search for the most recent time bucket first.

With regard to step C, document D11 discloses storing the machine data in a sorted table in a relational database management system, but it does not mention any indexing of the stored data. However, the use of indexes for querying was well-known in relational database management systems and thus indexing cannot be the basis for acknowledging inventive step. Document D11 does not explain how the sorted table is actually

stored, but it was usual to store such a table not in a single storage area, but in several storage areas (in the main memory or secondary storage). As the data table is sorted in chronological order, different parts of this table, which are stored in different storage areas, correspond to non-overlapping time buckets as claimed. The Board is aware that generally a further difference could be that with a sorted table the events stored within a particular part of the table are stored in sorted order, whereas the events assigned to an individual time bucket may be stored unordered. However, as steps C and D of claim 1 do not specify whether or not the data within an individual time bucket is sorted, there is no further difference which the Board needs to take into account.

As to the search requests received in step D of claim 1, D11 discloses receiving time series search requests and generating a result set organised by time (see D11, paragraphs [0053] and [0058]). The Board considers that the limitation to a specified number of results represents a non-technical requirement as the desired number of results depends on the user. Moreover, it was usual to limit the number of results. This also applies to the requirement that the results are sorted in reverse chronological order. As it was well-known in the field of relational database management systems to exploit an existing order of stored data for query processing, the skilled person faced with the task of implementing, in the method known from document D11, search requests requiring a limited number of results in reverse chronological order would consider processing the sorted table in reverse chronological order. Consequently, it was obvious to start the processing with the most recent data.

As to the generation of sub-searches targeted at individual time buckets, the Board fails to see any effect that can be derived from the features of the claim. The description, paragraph [0076], discloses the following with respect to the sub-searches:

> "Time buckets are queried in the order that is most advantageous to pruning given the sort order for the results. For example, if search results are sorted in reverse chronological order, then the sub-search for the most recent time bucket will be issued first. This allows the search execution engine 620 to examine the results 635 of the sub-search before proceeding with additional (expensive) sub-searches 625. For example, if a particular sub-search returns enough results 635, then it is not necessary to proceed with additional sub-searches 625."

Hence, the description supports a sequential execution of sub-searches. In the oral proceedings, the appellant argued that the sub-searches could be performed in parallel or concurrently, but did not refer to a passage supporting its argument. Moreover, the claim is not limited to the parallel or concurrent processing of sub-searches. As sequential processing of the time buckets in temporal order corresponds to sequential processing of the sorted table, the Board does not see any effect of the use of sub-searches.

In its statement of grounds of appeal, the appellant submitted that time-based searches according to the claimed method were more efficient because of the time bucketed indexing of events, e.g. because only buckets corresponding to the time range of the search needed to be accessed via sub-searches. However, claim 1 does not

refer to a time range of the search or to accessing only time buckets corresponding to a particular time range.

Furthermore, the appellant argued that another effect of the claimed method was that a chronologically ordered result set could be returned without sorting by searching buckets in the required sort order. However, the Board considers that sorting is still necessary if the data within individual time buckets is not stored in sort order and claim 1 does not specify a sort order for events within a bucket. Moreover, according to D11, the data is stored in a chronologically sorted table so that it was obvious to process the data in reverse chronological order to answer a search request demanding results in this order. As a consequence, the results are already available in the required sort order and no sorting is necessary. Hence, the Board is not convinced by the appellant's arguments.

In view of the above, the Board considers that, on the relevant date, the skilled person would arrive at steps C and D in an obvious manner.

5.5 Consequently, the subject-matter of claim 1 of the sole request lacks inventive step in view of document D11 and the common general knowledge of the skilled person (Article 56 EPC).

Conclusion

6. As the appellant's sole request cannot form the basis for the grant of a patent, the appeal is to be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

I. Aperribay R. Moufang

Decision electronically authenticated