BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS

BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE

CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN DES BREVETS

Internal distribution code:

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

Datasheet for the decision of 2 April 2019

Case Number:	T 1125/17 - 3.5.06
Application Number:	04777236.3
Publication Number:	1636699
IPC:	G06F9/44
Language of the proceedings:	EN

Title of invention:
COMPUTER-AIDED PARALLELIZING OF COMPUTATION GRAPHS

Applicant:
Ab Initio Technology LLC

Headword:
Parallelizing computation graphs/AB INITIO

Relevant legal provisions:
EPC 1973 Art. 56, 83, 84
EPC Art. 123(2)

Keyword:
Claims clarity - main and auxiliary requests 1-3 (no)
Sufficiency of disclosure main and auxiliary requests 1-3 (no)
Amendments added subject-matter - auxiliary requests 3 and 4 (yes)
Inventive step auxiliary request 5 (no)

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Decisions cited:

G 0002/10, T 0288/84, T 0689/90, T 1173/97, T 0641/00, T 1415/07

Catchword:

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 1125/17 - 3.5.06

DECISION
of Technical Board of Appeal 3.5.06
of 2 April 2019

Appellant: (Applicant)	Ab Initio Technology LLC 201 Spring Street Lexington, MA 02421 (US)
Representative:	Lloyd, Patrick Alexander Desmond Reddie & Grose LLP The White Chapel Building 10 Whitechapel High Street London E1 8QS (GB)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 13 December 2016 refusing European patent application No. 04777236.3 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	W. Sekretaruk
Members:	M. Müller
	S. Krischer

Summary of Facts and Submissions

I. The appeal is against the decision of the examining division dated 13 November 2016 to refuse European patent application No. 04 777 236 for lack of clarity, Article 84 EPC 1973, and inventive step, Article 56 EPC 1973, as the mere computer implementation of a purely mathematical method.

II. A notice of appeal was filed on 13 February 2017, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 21 April 2017. The appellant requested that the decision be set aside and a patent be granted on the basis of claims 1-21 according to a main or an auxiliary request as filed with the grounds of appeal.

III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claimed invention was deficient under Articles 83, 84 and 56 EPC 1973.

IV. In response to the summons, by letter dated 4 March 2019, the appellant filed new sets of claims 1-17, 1-17, 1-15, 1-15, 1-14 according to a main request and auxiliary requests 1-4, respectively.

V. During the oral proceedings held on 2 April 2019, the claims of the main request were again replaced by amended claims 1-15 and claims 1-3 according to an additional auxiliary request 5.

VI. Claim 1 of the main request reads as follows:

"A computer-implemented method for automated generation, from a user specified computation graph, of a parallel computation graph that specifies the distribution of the computations of the user specified computation graph for parallel execution on a number of separate processors, the parallel computation graph including:

at least a first parallel data processing element representing a plurality of instances of a first data processing element arranged in parallel;

at least a second parallel data processing element representing a plurality of instances of a second data processing element arranged in parallel; and

at least a parallelized inter-component link joining the first parallel data processing element and the second parallel data processing element,

whereby, during execution of the parallel computation graph, each instance of the data processing elements may be hosted on a different processor;

the method including:

accepting, by a computer system, a specification of the user specified computation graph in which data processing elements are joined by linking elements, each of the linking elements being associated with a data flow from an associated upstream one of the data processing elements to an associated downstream one of the data processing elements;

for each of one or more of the linking elements (205) of the computation graph, that joins an upstream data processing element (210) to a downstream data processing element (240) and for which parallel characteristics are undetermined:

determining, by the computer system, data processing characteristics, including a degree of parallelism, of a parallelized inter-component link representing the linking element (205) and configured to process data records with a degree of parallelism, the determining being according to characteristics of the upstream and the downstream data processing elements (210, 240) associated with the linking element (205), including

determining data processing characteristics of the parallelized inter-component link based on a data characteristic for data records provided on an output port by the upstream data processing element and for data records received on an input port by the downstream data processing element; and

the method further comprising inserting the parallelized inter-component link as a parallel partition element, interconnection network, and a parallel gather element on a data path joining parallel components of the upstream and the downstream data processing elements associated with the linking element."

Claim 1 of auxiliary request 1 differs from that of the main request in that the data characteristics are further specified by this phrase:

"... the characteristics including one or more input requirements of the downstream data processing element (240) and one or more output requirements of the upstream data processing element (210), ..."

and that the last paragraph reads as follows:

"... inserting the parallelized inter-component link as at least a data processing element on a data path joining the upstream and the downstream data processing element associated with the linking element, including configuring the parallelized inter-component link according to the data processing characteristics of the linking element."

Claim 1 of auxiliary request 2 differs from that of auxiliary request 1 in that what is laid out as the penultimate paragraph of claim 1 above reads as follows:

"... the determining including determining one or more sort requirements for the data elements processed by the parallelized inter-component link and one or more partitioning requirements for the data elements processed by the parallelized inter-component link based on a data characteristic for data records provided on an output port by the upstream data processing element according to the one or more output requirements of the upstream data processing element and for data records received on an input port by the downstream data processing element according to the one or more input requirements of the downstream data processing element; ..."

Claim of auxiliary request 3 differs from that of auxiliary request 2 in that the following phrase is added to its end:

"... and configuring the parallelized inter-component link for parallel execution on at least two separate processors".

Claim 1 of auxiliary request 4 differs from that of auxiliary 3 in that the inserting step is amended and further steps were added as follows:

"... inserting the parallelized inter-component link, as at least a parallel partition element (221), a parallel gather element (231) and an interconnection network (225), on a data path joining m instances of the upstream data processing element and a *n* instances of the downstream data processing element associated with the linking element, wherein the parallel partition element (221) is implemented by m instances of a partition element (220), the parallel gather element (231) is implemented by *n* instances of a gather element (230), and the interconnection network (225) is implemented as a cross-connection of serial links in which every instance of partition element (220) is connected to every instance of gather element (230), the method further including configuring the parallelized inter-component link according to the data processing characteristics of the linking element and configuring the parallelized inter-component link for parallel execution on at least two separate processors;

the method further comprising executing, on a computer system comprising at least two separate processors, the parallel computation graph by:

preparing the graph for execution by performing, on the computer system, graph transformation steps until the graph is in an executable form, and each serial link is associated with at least one communication method compatible with the access methods of the instances of the first and second data processing elements connected by the serial link;

launching each serial link by creating, by means of the computer system, a combination of communication channels and/or data stores, as appropriate to the serial link's communication method; and

launching each data processing element by invoking execution of the process on the computer system."

The beginning of claim 1 of auxiliary request 5 is identical to claim 1 of the main request. From the determining step onwards, it reads as follows:

"... determining, by the computer system, data processing characteristics of a parallelized inter-component link (205) representing the linking element and configured to process data records with a degree of parallelism, the parallelized inter-component link comprising a parallel partition element (221), a parallel gather element (231) and an interconnection network (225), the determining being based on:

the degree of parallelism of the upstream data processing element (m) and the downstream data processing element (n) associated with the linking element;

the characteristics of the output flow from the upstream data processing element (210) associated with the linking element;

the requirements of the input flow of the downstream data processing element associated with the linking elements;

the determining being based on:

metadata associated with the processing elements, the metadata comprising data indicators that indicate whether: if partitioned, the input flow of the processing element must be partitioned according to a particular key or keys; each instance of the component must receive copies of all work elements on its input; and the input flow must be sorted, and the key or keys that define the sort order; and

processing element specific mapping functions that accept the characteristics of each of the input data flows of the components, and produces characteristics for each of the output flows;

wherein:

if m=n, and the input to the downstream data processing element does not need to be partitioned or sorted according to any particular key, and the input flow does not need a copy of each work element, corresponding instances of the upstream and downstream components are connected directly by serial links;

if m is not equal to n and the input flow to the downstream component does not need to be partitioned according to any particular key, and the input flow does not need a copy of each work element, then the partition element of the inter-component link is defined to perform a round-robin distribution;

if the input flow to the downstream component requires the work elements to be partitioned according to a set of keys that is different than the partitioning of the output flow of the upstream component, the partitioning element performs a hash partition according to the required key values; and

if the input flow requires a copy of each work element, then the partition element of the inter-component link is defined to perform a broadcast function;

and

inserting the parallelized inter-component link on a data path joining m instances of the upstream data processing element and a *n* instances of the downstream data processing element associated with the linking element, wherein the parallel partition element (221) is implemented by m instances of a partition element (220), the parallel gather element (231) is implemented by *n* instances of a gather element (230), and the interconnection network (225) is implemented as a cross-connection of serial links in which every instance of partition element (220) is connected to every instance of gather element (230),

the method further comprising executing, on a computer system comprising at least two separate processors, the parallel computation graph by distributing the parallel computation graph and performing parallel execution thereof on a number of separate processors."

VII. At the end of the oral proceedings, the chairman announced the decision of the board.

Reasons for the Decision

The invention

1. The application generally relates to computation graphs in which the vertices and the links define, respectively, "data processing elements" and the data flow between them (see paragraphs 3, 4 and 7).

1.1 In this context, the application relates to the transformation of "serial computation graphs" into "parallel computation graphs".

1.2 Figure 1A depicts a simple "serial" computation graph in which two components A and B are connected by a single "serial link" (see paragraph 4). It is observed that data may sometimes be processed in parallel by different "instances of individual components" A and B, the possible "degree of parallelism" depending on the "characteristics" of A, B and the connection between them (see paragraphs 5 and 30 but also figure 3, no. 330). Making this parallelism explicit in the computation graph yields a "parallel[ized] computation graph" (see figure 1B, 2A, 2B; paragraphs 4-6, 29, 36 and 37), which can also be represented in "serial form" (see paragraphs 39 and 54).

1.3 The data flow in parallel computation graphs is represented using data flow elements referred to as "partition", "gather" and "interconnection" (see figures 1B and 1C, nos. 115 and 135; figures 2B and 2C, no. 225). Several types of each of these elements are anticipated, depending on the type of computation (see paragraphs 30-32, 35, 49).

1.4 The invention means to propose a procedure for the transformation of serial computation graphs into equivalent parallel ones (see figure 3). The procedure may be able to rely on user annotations, indications or other metadata (see paragraphs 43-49 and 51-52) but may also have to make do without them (paragraphs 64-66).

1.5 The procedure according to figure 3 cycles through the "links" in the given serial graph and determines the required partition, interconnection and gather elements (see paragraph 53).

Clarity, Article 84 EPC 1973, and claim construction

2. The claimed method works by iteration over "linking elements" and "determining" various "data processing characteristics" which were "undetermined" before, in particular, "a degree of parallelism", "input requirements" and "sort requirements".

2.1 Claim 1 of the main request and auxiliary requests 1-4 leaves open how these characteristics are determined. Claim 1 of auxiliary request 5 specifies that the determining is based on "metadata" which, according to the description, may be provided manually by the user, i.e. the programmer (*loc. cit.*). Although the appellant insisted during oral proceedings that the determination could also rely on sources of information other than user annotations, no further explanations were given on how that was meant to be done. Neither does the description which supports the appellant's argument disclose details in this regard (see paragraphs 64 to 66). Either way, no amended claim 1 was filed containing a pertinent amendment. Therefore, claim 1 of all requests is at least consistent with the assumption that the "determining" step essentially retrieves user-provided metadata from the user-specified graph.

2.2 Claim 1 of the main request and auxiliary requests 1-4 does not define the characteristics to be determined. For instance, the "degree of parallelism" is only defined in claim 1 of auxiliary request 5 which states it to specify the desired number - m and n, respectively - of upstream and downstream processing elements.

2.3 Moreover, claim 1 of the main request and auxiliary requests 1-3 fails to specify what happens with the determined characteristics. The step of "inserting the parallelized inter-component link" is merely said to be "configur[ed] according to the data processing characteristics", but how this is done is not stated.

2.4 This constitutes a lack of essential features and renders claim 1 of the main request and auxiliary requests 1-3 unclear, Article 84 EPC 1973. Moreover, because neither the data processing characteristics nor their use in configuring the parallelized inter-component link are defined, the skilled person would not have been able to carry out the invention in its full breadth, in violation of Article 83 EPC 1973.

Article 123(2) EPC

3. Claim 1 of auxiliary request 3 contains the phrase "configuring the parallelized inter-component link for parallel execution on at least two separate processors". As its basis, the appellant refers to paragraph 20 of the application as filed.

3.1 Apart from the fact that a graph "configur[ed] for parallel execution" is not necessarily executed in parallel, paragraph 20 does not specifically disclose the "parallelized inter-component link" to be distributed but, more generally, the "computations specified by an initial serial [or parallel] computation graph".

3.2 The amendments made to claim 1 of auxiliary request 1 thus do not comply with Article 123(2) EPC.

4. Claim 1 of auxiliary request 4 was amended by incorporation of "features relating specifically to the execution of the resulting parallel computation graph[,] based upon material found in reference document US 5966072" (see page 7 of the letter of 4 March 2019, section entitled "Amendments", paragraph 3). This document is assigned to the present appellant and is mentioned in paragraph 3 of the present application documents as originally filed by the brief statement that "A system that implements such graph-based computation is described in U.S. Patent 5,966,072 EXECUTING COMPUTATIONS EXPRESSED AS GRAPHS".

4.1 It is established jurisprudence of the boards of appeal that an amendment in conformance with Article 123(2) may only be made within the limits of what a skilled person would derive directly and unambiguously, using common general knowledge, and seen objectively and relative to the date of filing, from the whole of these documents as filed" (G 2/10, point 4.3 of the reasons).

4.2 When protection is sought for features which were not expressly disclosed in the application documents as originally filed but were taken from a cross-referenced document, it has further been established to use the criteria defined in T 689/90 (catchword 2). This decision is also mentioned in the appellant's letter of 4 March 2019 (bottom of page 7). During the oral proceedings, the appellant made further reference to T 1415/07 and T 288/84 to support its case.

4.3 In accordance with T 689/90, an amendment based on a feature only described in a cross-referenced document is only allowable "if the description of the invention as filed leaves no doubt to a skilled reader", *inter alia*, "(a) that protection is or may be sought for such features" and "(c) that such features implicitly clearly belong to the description of the invention contained in the application as filed [...] and thus to the content of the application as filed".

4.4 The application states specifically that the contents of the U.S. Provisional Application of which it claims the benefit is "incorporated [...] by reference" (paragraph 1). This statement means to make the content of the provisional application part of the content of the present one. While it is open to argument which features specifically, if any, are (or can) be made part of the content of an application by way of a generic reference like this, this statement is notably missing from paragraph 3. That is, US 5 966 072 is not stated to be incorporated by reference.

4.5 In compliance with the strict standard regarding added matter (see point 3.1 above), requirement (a) is very stringent. It requires that the description "leaves no doubt to a skilled reader" that protection is or may be sought for the pertinent features. In the words of T 1415/07 (point 19 of the reasons), "it must be unambiguously derivable to the skilled person which features of the application are to be taken from the referenced document". A mere mention of a prior art document as an "example" is normally insufficient to identify which features or groups of features are of particular relevance for the citing application, especially if, as in the case at hand, the cited documents contains 32 pages. Also, absent any express indication in the citing application, the disclosure of the abstract and the summary from which the pertinent features were taken cannot be understood to form part of the content of the citing application (requirement (c)).

4.6 Thus, the brief mention of the U.S. patent 5 966 072 in the application at hand is insufficient to make any specific features disclosed in the mentioned patent part of the content of the application as originally filed.

4.7 The situation underlying T 288/84 does not, in the board's judgment, withstand this finding. In that case, the insertion of a single word into the claims was allowed under Article 123(2) EPC because the underlying concept was disclosed in all examples of the application, albeit without expressly mentioning it by name (see point 6.4 of the reasons). In other words, the board in case T 288/84 essentially decided that it was allowable to insert the name of a particular concept that was, without being named, clearly disclosed.

4.8 In the present case, paragraph 20 discloses that "the computation specified by a serial computation graph can be distributed for parallel execution on a number of separate processors". This statement as such is very broad and does not imply which components of the graph are or might be deployed on separate processors. Furthermore, the general reference in paragraph 3 of the present application to "a system that implements [...] graph-based computation" does not imply, without doubt, that the relevant features in the U.S. patent relate to how a parallel computation graph is distributed for parallel computation.

4.9 As a consequence, in view of the fact that the amendments made to claim 1 of auxiliary request 4 crucially rely on that U.S. patent and with no other basis for the added claim language being given, claim 1 of auxiliary request 4 does not comply with Article 123(2) EPC.

Inventive step, Article 56 EPC 1973

5. The amendments made to claim 1 of auxiliary request 5 specify that it is metadata provided in or with the user-specified computation graph which indicates how many upstream or downstream data processing elements there are (i.e., in the terms of figure 1C, how many As and Bs), whether the input flow to any data processing element must be partitioned or sorted and whether every "instance" of a component must receive a copy of each work element. According to the requirements expressed in the metadata, the sorted-merge element is generated. For instance, if the downstream components expect the

input to be partitioned differently to how the output of the upstream component is provided, a "hash partition" element is provided which is disclosed to redistribute the work elements according to the required partitions (see paragraph 32).

5.1 This subject-matter specifies (or, rather, is consistent with the assumption) that all information relevant for the automated generation of the parallel computation graph from the user-specified one is already contained explicitly, in the form of user-provided metadata, in the user-specific graph. In other words, the parallel computation graph contains the same amount of information as the user-specified graph and, at best, represents it in an explicit, expanded form. This interpretation was presented to the appellant during oral proceedings and was not challenged.

5.2 The board takes the view that any actual parallel execution of the parallel computation graph is, in principle, also available for the user-specified graph. Hence, whatever speed-up effect can in principle be achieved by parallelizing one graph is not due to the form of the graph but to the details of the user-specified graphs including its metadata.

5.3 Moreover, it is not evident that the expanded form of the graph, the "parallel" computation graph, is, in general, substantially easier to deploy on parallel hardware than the user-specified graph. Such an argument cannot be used for claim 1 as it stands because specifically how the parallel execution graph is deployed on a given hardware is not claimed.

5.4 Thus, the mapping of one graph to another, which is equivalent to expressing the same computation, is, essentially, a mathematical operation on computation graphs, which does not contribute to the effect of parallelization. The transformation itself therefore does not contribute towards inventive step in accordance with the established jurisprudence of the boards of appeal, see T 641/00 (catchword 1).

5.5 Beyond that, the general idea that a computation graph may be executed on some parallel hardware is known in the art (e.g. from the U.S. patent mentioned in paragraph 3 of the application) and therefore obvious as such.

5.6 The subject-matter of claim 1 of auxiliary request 5 thus lacks inventive step, Article 56 EPC 1973.

Further remark

6. During the oral proceedings, in the context of auxiliary request 4, it was discussed whether the transformation of one computation graph into another, without an explicit mention of the deployment of parallel hardware or the execution on parallel hardware, would have to be accepted as a technical effect on the assumption that the generated computation graph was established as lending itself more easily to parallelization than the given one. Although this question turned out not to be decisive for the case at hand, the appellant encouraged the board to comment on this question in its decision.

6.1 The board stresses that, as argued above, it is not convinced of the argument that the generated parallel computation graph is actually easier to parallelize than the given user-specified graph. For this reason alone, the following comment is strictly obiter.

6.2 A computation graph meant to be executed is, essentially, a computer program.

6.3 Even a program written in a programming language with parallelization instructions or with some express potential for parallel execution such as array processing may be executed on parallel hardware or not. Such a program can also be executed on a single-core processor. In this case, a speed-up by parallelization is not achieved. In many cases, a parallel program could be expected to execute more slowly on the single core than an equivalent serial program since any parallelization overhead is not compensated by the speed-up of parallel computation. This is to say that the speed-up of parallelization is not achieved by the form of the parallel program alone and not before the program is actually deployed and executed on parallel hardware.

6.4 In T 1173/97, it was decided that a computer program is not excluded from patentability if, when it is run on a computer, it produces a further technical effect which goes beyond the "normal" physical interactions between program (software) and computer (hardware) (see catchword and point 13 of the reasons).

6.5 This board takes the view that T 1173/97 meant to make this statement only if the mentioned further technical effect was produced *whenever* the program was run, i.e. on *any* suitable hardware or runtime environment. For if that effect was produced on a particular execution platform but not on another, the effect could not be attributed to the program itself - unless maybe there was an argument to the effect that the required execution platform was implicit in the program claim. In all other situations, it would seem that the execution platform required to achieve the effect would have to be claimed as an essential feature.

6.6 This would appear to mean that if, as in the present case, an inventive-step argument is to rely upon a speed-up by parallelization, a parallel execution platform must be claimed. Consequently, the mere potential for a speed-up by parallelization would not seem to be sufficient as a "further" technical effect because this effect is not achieved irrespective of how the program is executed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts ✦ Boards of Appeal of the European Patent Office ✦ Chambres de recours de l'Office européen des brevets ✦

N. Schneider W. Sekretaruk

Decision electronically authenticated