

Internal distribution code:

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

**Datasheet for the decision
of 18 January 2017**

Case Number: T 1166/11 - 3.5.06

Application Number: 06789282.8

Publication Number: 1910925

IPC: G06F9/46

Language of the proceedings: EN

Title of invention:

GENERAL INTERFACE WITH ARBITRARY JOB MANAGERS

Applicant:

The MathWorks, Inc.

Headword:

Job scheduling/THE MATHWORKS

Relevant legal provisions:

EPC 1973 Art. 56

Keyword:

Inventive step - (no)

Decisions cited:

Catchword:



Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 1166/11 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 18 January 2017

Appellant: The MathWorks, Inc.
(Applicant) 3 Apple Hill Drive
Natick, MA 01760 (US)

Representative: Rupprecht, Kay
Meissner Bolte Patentanwälte
Rechtsanwälte Partnerschaft mbB
Widenmayerstraße 47
80538 München (DE)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted on 28 December
2010 refusing European patent application No.
06789282.8 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 lies against the decision of the examining division, with reasons of 28 December 2010, to refuse European patent application No. 06 789 282.8 for lack of inventive step over document
- D1: US 2003/187983 A1.
- II. A notice of appeal was received on 25 February 2011, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 29 April 2011. The appellant requested that the decision be set aside and that a patent be granted on the basis of claims 1-24 as filed with the grounds of appeal, in combination with the description and the drawings on file.
- III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claimed subject-matter lacked inventive step over D1, Article 56 EPC 1973. A number of clarity objections were also raised, Article 84 EPC 1973.
- IV. In response to the summons, with a letter of 19 December 2016, the appellant filed three sets of amended claims 1-17, 1-14 and 1-6 according to a main request and two auxiliary requests.
- V. Claim 1 of the main request reads as follows:
- "A method of executing a job in a distributed computing environment comprising a client device (121), a scheduler (131), storage (271) and a plurality of remote workers (141, 151, 161), wherein the client device and the plurality of remote workers each comprises an interface (122) and wherein the interface

comprises a scheduler object (410) to represent the scheduler (131), a scheduler interface (420) for communicating with the scheduler (131) using a scheduler communication channel and a storage interface (430) for communicating with the storage (271) using a storage communication channel; the method comprising the steps of:

creating by the client device a job in a storage location of the storage using the storage interface; the job including a plurality of tasks, each task defining a technical computing command to be executed, a number of arguments of the technical computing command, and any input data to the argument, and each task being separately accessible from the storage;

transferring by the client device a reference to the storage location to the scheduler through the scheduler interface on the client;

distributing by the scheduler the tasks to the plurality of remote workers by transferring the reference to the storage location;

retrieving the tasks from the storage location by the plurality of remote workers using the reference; and

executing the tasks on the plurality of remote workers

wherein each worker executes at least one designated task and saves the result of executing the task in the storage location of the storage."

Claim 1 of the first auxiliary request is identical to that of the main request, with the following text added at its end:

"... wherein the reference comprises a machine name and a TCP port number, or an IP address and a TCP port number and the name of a database on that machine."

Claim 1 of the second auxiliary request is identical to that of the main request with the following text added at its end:

"... wherein the job contains algorithms or models executed repeatedly on varying data sets and the job is divided into independent tasks using a toolbox; and wherein a distributed computing engine is run on remote sessions of the remote workers to evaluated [sic] the tasks and the tasks are stored in the MAT-file format."

The main and the first auxiliary request also contain a corresponding independent system claim 6.

- VI. Oral proceedings took place as scheduled. At the end of the oral proceedings, the chairman announced the decision of the board.

Reasons for the Decision

The invention

1. The application relates to the distribution of computational "jobs" to the workers in a distributed computing environment. More specifically, the application discloses a means for the job creating clients and the compute servers to interface with each other via any ("arbitrary") job manager or scheduler, which may be supplied by different ("arbitrary") vendors (see page 3, paragraphs 1 and 2, and page 12, paragraph 2).
- 1.1 Applications are described as being defined in terms of "jobs" and "tasks", and it is defined that a "job is a logical unit of activities, or tasks that are processed

and/or managed collectively" and that a "task defines a technical computing command [...] to be executed, and the number of arguments and any input data to the arguments" (page 6, penultimate paragraph).

- 1.2 The invention proposes that the clients create and store the jobs and its tasks within each job in dedicated storage and transmit only the storage locations to the "arbitrary job scheduler" (see page 4, paragraph 1, page 11, lines 17-21, page 16, paragraph below the table, and page 21, lines 4-8). The worker which the scheduler selects for carrying out a job or task is forwarded the corresponding reference so that it can retrieve "the job or tasks" from that storage location (see page 21, last paragraph). The application suggests that tasks may be distributed and executed individually (see page 22, lines 2-4).

- 1.3 The application is specifically concerned with jobs defined in MATLAB for solving engineering and scientific problems (see page 1, last paragraph). It is disclosed that the "MATLAB-based distributed computing environment" provides what is called the "Distributed Computing Toolbox" which enables users to divide an application into "independent tasks" (see page 6, last paragraph, to page 7, paragraph 1). The claims are not, however, limited to any particular type of application or to MATLAB. Claim 1 of the second auxiliary request only alludes to MATLAB by mentioning "a toolbox" and that tasks are stored in the "MAT-file format".

The prior art

2. D1 discloses a distributed computing system with a portable application distributor (PAD) providing an API

via which client computers can use different distributed resource management systems (DRMSs) for the distribution of given "jobs" (see e.g. paragraphs 2, 3, 17, 22 and 31).

- 2.1 D1 discloses that DRMSs generally use one of two "modes" of distribution, a job-centric mode and data-centric mode, depicted in figures 3A and 3B. In the data-centric mode the distributed jobs retrieve their input data from and write their output data to joint central storage (see figure 3A and paragraph 27, lines 6-11), whereas in the job-centric mode jobs are distributed with "private copies" of their input data and have to return their output data to the DRMS (see figure 3B and paragraph 27, lines 12-17).
- 2.2 In D1, a distinction is made between "jobs" and their input and output files. Therefore, the board takes the view that the skilled person would take a "job" according to D1 to mean some sort of program code or instructions distinct from the processed data.
- 2.3 The board agrees with the appellant that D1 does not "contain any disclosure relating to how jobs are distributed by the DRMS" and in particular "no indication that the DRMS sends storage locations to the remote compute[]server systems" (see the grounds of appeal, page 5, paragraphs 3 and 4, with reference to D1, paragraphs 32-50). However, the board considers it to be implicit in D1 that the jobs are transferred from the master host to the remote compute servers one way or another (see D1, paragraph 1).

Claim construction

3. In its comparison of claim 1 with D1, the examining division states that jobs are created in a storage location (see reasons, page 4, paragraph 2) and seems to argue that, in the data-centric mode in D1, jobs are retrieved from this storage location since the compute servers retrieve the *job data* from a central storage location (see decision, page 4, last 16 lines). In particular, it is explained that specifying, in the data-centric mode of D1, the input and output files in a shared storage space (see D1, paragraphs 27 and 28) "corresponds", in the claimed invention, "to providing the 'reference to the storage location'" of the job to the one or more workers.
- 3.1 In the board's understanding this argument assumes that the claimed "jobs" are identified with the input data of D1.
- 3.2 While, as explained in the annex to the summons, the board tends to find this a plausible interpretation of the claimed invention because the jobs and tasks constitute, in a sense, the input data to the distributed computing engine in a remote MATLAB session (see the application, page 7, paragraphs 1 and 2), the board takes the view that the skilled person would prefer the more conventional interpretation according to which the claimed jobs and tasks consist primarily of code (if not exclusively so, see again the description, page 6, line 25).

4. Claim 1 (of all requests) discloses that "the tasks" are distributed, retrieved and executed, and that "each worker executes at least one designated task".

4.1 According to the appellant, as it argued in oral proceedings and suggested in its letter of 19 December 2016, claim 1 implied that workers may retrieve the tasks individually from their storage location. Accordingly, the units of distribution were the individual tasks.

4.2 The board does not accept that the claim language implies this reading. Rather, claim 1 does not exclude the interpretation that "the tasks" are distributed, retrieved and executed in groups as defined by the jobs, where a job may contain only a single task. The unit of distribution from this perspective would be the jobs.

Inventive step

5. As mentioned above (see point 3.2), the board considers that the skilled person would identify the *jobs* of D1 with the units of distribution according to the invention, i.e. with the jobs or tasks as the case may be.

5.1 Whether each individual task is stored at a separate storage location or whether, as the appellant considers, all the tasks of a job are stored at the same, i.e. the job's, location (see e.g. the description, page 4, lines 5-7), is, in the board's view, immaterial, because the individual tasks could also be accessed at their job's location.

6. In view of the foregoing, the board finds that the subject-matter of claim 1 of the main request differs from D1 in that

1) the jobs and tasks are created in a shared storage location from where the workers retrieve the tasks.

6.1 When the unit of distribution is the job, this is the only difference, because, in the board's view, the complex jobs of D1 implicitly comprise parts which can be identified with the claimed "tasks".

6.2 When the unit of distribution is the task, claim 1 of the main request further differs from D1 in that

2) the jobs of D1 (= the claimed tasks) are obtained as part of a larger entity (= the claimed job).

6.3 The appellant argued that these differences had two advantages. Difference 1) in particular had the effect that clients could use "arbitrary" schedulers because all schedulers were equipped to receive address parameters and thus jobs or tasks "by reference", whereas not all schedulers could receive the entire job or task code ("by value"). Both differences, but in particular difference 2), also had the additional effect of simplifying parallelisation so that it was available to unskilled programmers (see e.g. the appellant's letter of 19 December 2016, page 7, paragraph 4).

6.4 The board does not accept that either of these effects can be ascribed to the claimed invention.

6.4.1 There is no basis in the application for any assumption as to what type of arguments known schedulers could

receive, nor for the argument that "arbitrary schedulers" might become usable merely due to the decision to transmit jobs to schedulers "by reference" rather than "by value".

6.4.2 The board takes the position that the division of programs into separately executable (i.e. parallelisable) parts is, in general, a well-researched and well-understood problem. Whether such division is "simple" may depend on the given type of programs (i.e. the jobs), the kind of parts which the division is meant to produce (i.e. the tasks), and on further conditions the division may have to satisfy (e.g., in view of data-dependencies). The application leaves open whether the division of jobs into tasks was fully automatic, or only semi-automatic requiring manual intervention by the programmer. The board thus concludes that the claims and the application as a whole do not support the assertion that the differences achieve (or even contribute to) the effect of making parallelisation available to an unskilled programmer.

7. Instead, the board considers that difference 1), i.e. the transfer of code "by reference" into shared memory, *inter alia* reduces the required amount of buffer space at the compute servers, and that difference 2), if it exists, has the effect of making the system of D1 available to larger entities (than the jobs of D1).

7.1 It is well known in the art that parameters of a procedure call can be transmitted "by reference" into shared memory or "by value", as are the relative advantages and disadvantages of these alternatives. The board therefore takes the view that the skilled person would not hesitate to use either of them, should the

circumstances so require - e.g. so as to save buffer space on the compute servers.

- 7.2 And the idea of splitting a large program into smaller parts is a well-known and obvious solution to the problem of enabling parallelism within that program. Once these parts are defined, they can be straightforwardly parallelised, as explained in D1.
- 7.3 In summary, the board concludes that neither difference 1) nor 2) is sufficient to establish an inventive step of claims 1 or 10 over D1, Article 56 EPC 1973.
8. As regards the auxiliary requests, the appellant has not provided arguments that and in what way the additional features contribute to inventive step of the claimed subject-matter. Rather, during oral proceedings the appellant stated that the auxiliary requests were meant to further elucidate the inventive-step argument vis-à-vis the main request, and that the second auxiliary request was meant to address the above-mentioned objection from the summons based on the identification of the claimed jobs or tasks with the input data of D1, which the board decided not to pursue.
 - 8.1 Having said that, the board considers the following.
 - 8.1.1 As regards the first auxiliary request, the board is of the opinion that the specification of a required storage location in a distributed system in terms of "a TCP port number, or an IP address and a TCP port number and the name of a database on that machine" is an obvious option which the skilled person would consider without exercising inventive skill.

8.1.2 As regards the second auxiliary request, three additional features of claim 1 need to be considered:

- (a) That a "job contain[ing] algorithms or models executed repeatedly on varying data sets", as is regularly the case in mathematical and scientific computation, may be qualified for straightforward parallelisation is, in the board's view, well-known in the field. How this fact is, however, taken into account in the division of jobs into tasks (or beyond that) is not claimed.
- (b) The further fact that "the job is divided into independent tasks using a toolbox" raises concerns under Article 84 EPC 1973 due to the undefined term "toolbox". If, however, "toolbox" is interpreted as some sort of library and "using" the toolbox is interpreted as not excluding manual intervention by the programmer, the board finds that this feature only reflects the obvious option of using software support in the process of parallelisation.
- (c) The choice of the MAT-file format for storing the tasks is, in the board's view, in principle an obvious choice and does not have any impact on the problem of how jobs and tasks are created and distributed across the workers.

8.2 In summary, the board concludes that the additional features of the two auxiliary requests do not establish inventive step over D1 either.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:

The Chairman:



B. Atienza Vivancos

W. Sekretaruk

Decision electronically authenticated