Internal distribution code:

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

Datasheet for the decision of 27 April 2016

Case Number:	T 1130/11 - 3.5.06
Application Number:	05252239.8
Publication Number:	1580661
IPC:	G06F9/46
Language of the proceedings:	EN

Title of invention:
Scheduling in a high-performance computing (HPC) system

Applicant:
RAYTHEON COMPANY

Headword:
HPC scheduling/RAYTHEON

Relevant legal provisions:
EPC 1973 Art. 84, 111(2), 56

Keyword:
Claims - clarity (yes)
Remittal to the department of first instance - (no)
Inventive step - (no)

Decisions cited:

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

Catchword:

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

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 1130/11 - 3.5.06

DECISION
# of Technical Board of Appeal 3.5.06
of 27 April 2016

Appellant: (Applicant)	RAYTHEON COMPANY 870 Winter Street Waltham MA 02451-1449 (US)
Representative:	Lawrence, John Barker Brettell LLP 100 Hagley Road Edgbaston Birmingham B16 8QQ (GB)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 2 December 2010 refusing European patent application No. 05252239.8 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	W. Sekretaruk
Members:	S. Krischer
	A. Teale

Summary of Facts and Submissions

I. The appeal is directed against the decision of the examining division, dated 2 December 2010, to refuse the application 05252239 for lack of clarity. In an obiter dictum, an objection of lack of inventive step was raised, citing the following documents:

D1 "HP AlphaServer SC User Guide", Internet article, pages 1-38, XP002336777, downloaded on 19 July 2005 from http://web1.quadrics.com/onlinedocs/AlphaServer/Eagle/html/AlphaServerUserGuide.

D2 Krevat, E. et al.: "Job Scheduling for the BlueGene/L System", Lecture Notes in Computer Science, 2002, vol. 2537, pages 38-54; XP002336423.

D3 Heiss, H-U.: "Processor Management in Two-Dimensional Grid-Architectures", Interner Bericht Nr. 20/92, University of Karlsruhe, 1992, pages 1-52; XP002416087.

D4 Wanqian Liu et al.: "Non-contiguous Processor Allocation Algorithms for Distributed Memory Multicomputers", Proceedings of Supercomputing'94, IEEE, DOI: 10.1109/SUPERC.1994.344282, pages 227-236; XP010100481; ISBN: 978-0-8186-6605-6.

II. A notice of appeal was received on 11 February 2011. The appeal fee was received the same day. A statement of the grounds of appeal was received on 12 April 2011. Claim sets of a main and two auxiliary requests were filed. Oral proceedings were requested.

III. In its summons to oral proceedings, the board gave reasons for its preliminary opinion that the claims were clear and that claim 1 of all of the requests lacked an inventive step.

IV. In a letter dated 24 March 2016, the appellant argued against lack of inventive step and corrected an obvious error in claim 1 of the first auxiliary claim request ("minimises" instead of "maximises") and filed a third auxiliary claim request.

V. Oral proceedings were held on 27 April 2016. At their end, the board announced its decision.

VI. The appellant requests that the decision be set aside and the case be remitted to the first instance according to Article 111(2) EPC, or that a patent be granted based on a main claim request (filed with the grounds of appeal), a first auxiliary claim request (filed on 24 March 2016), a second auxiliary claim request (filed with the grounds of appeal) or a third auxiliary claim request (filed on 24 March 2016).
The further text on file is: description pages 1, 5-96 as originally filed, page 2 filed on 5 December 2006, pages 3, 4, 4a and 4b filed on 12 June 2007; drawing sheets 1-11 as originally filed.

VII. Claim 1 of the main claim request reads as follows:

"1. Logic for scheduling in a high-performance computing (HPC) system, the logic encoded in a computer-readable medium and when executed operable to:

receive a call from a management engine (130, 515) operable to manage a cluster (220) of nodes (115) in the HPC system (100), the call specifying a request comprising a job (150) for scheduling comprising one or more processes for execution at one or more nodes (115) in the cluster (220), the call further specifying a number of nodes (115) for executing the job (150);

determine whether the request is spatial, compact, or nonspatial and noncompact, according to a request type in the request, the request being spatial if the job (150) assumes spatial relationships between nodes (115) executing the job (150), the request being compact if the job (150) assumes proximity between nodes (115) executing the job (150), the request being nonspatial and noncompact if the job (150) assumes no spatial relationships or proximity between nodes (115) executing the job;

if the request is spatial:

generate spatial combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call and further accommodating the assumed spatial relationships between nodes (115) executing the job (150); and

select one of the spatial combinations that is schedulable according to a list of nodes (115) in the cluster (220) available for scheduling;

if the request is compact:

generate compact combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call which comprise combinations of nodes that minimize the maximum communication distance or hop count between each pair of nodes; and

select one of the compact combinations that is schedulable according to the list of nodes (115) in the cluster (220) available for scheduling and that minimizes a maximum communication distance, or hop count, between each pair of nodes (115);

if the request is nonspatial and noncompact:

identify one or more nodes (115) schedulable according to the list of nodes (115) in the cluster (220) available for scheduling;

generate a nonspatial and noncompact combination of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call, the nonspatial and noncompact combination comprising one or more of the one or more identified nodes (115) that are schedulable according to the list of nodes (115) in the cluster (220) available for scheduling; and

communicate a return to the management engine (130, 515) identifying one or more nodes (115) in the selected spatial, compact, or nonspatial and noncompact combination of nodes (115) in the cluster (220) for executing the job (150)."

VIII. Claim 1 of the first auxiliary claim request reads as follows:

"1. Logic for scheduling in a high-performance computing (HPC) system, the logic encoded in a computer-readable medium and when executed operable to:

receive a call from a management engine (130, 515) operable to manage a cluster (220) of nodes (115) in the HPC system (100), the call specifying a request comprising a job (150) for scheduling comprising one or more processes for execution at one or more nodes (115) in the cluster (220), the call further specifying a number of nodes (115) for executing the job (150) and an aggressive flag indicating whether or not there is leeway for scheduling the spatial request as a compact request;

determine whether the request is spatial, compact, or nonspatial and noncompact, according to a request type in the request, the request being spatial if the job (150) assumes spatial relationships between nodes (115) executing the job (150), the request being compact if the job (150) assumes proximity between nodes (115) executing the job (150), the request being nonspatial and noncompact if the job (150) assumes no spatial relationships or proximity between nodes (115) executing the job;

if the request is spatial: and

if the nodes cannot accommodate the spatial request and the aggressive flag indicates leeway in the allocation of nodes to the spatial request, scheduling the spatial request as a compact request

generate spatial combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call and further accommodating the assumed spatial relationships between nodes (115) executing the job (150); and

select one of the spatial combinations that is schedulable according to a list of nodes (115) in the cluster (220) available for scheduling;

if the request is compact:

generate compact combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call which comprise combinations of nodes that minimize the maximum communication distance or hop count between each pair of nodes; and

select one of the compact combinations that is schedulable according to the list of nodes (115) in the cluster (220) available for scheduling and that minimises a maximum communication distance between each pair of nodes;

if the request is nonspatial and noncompact:

identify one or more nodes (115) schedulable according to the list of nodes (115) in the cluster (220) available for scheduling;

generate a nonspatial and noncompact combination of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call, the nonspatial and noncompact combination comprising one or more of the one or more identified nodes (115) that are schedulable according to the list of nodes (115) in the cluster (220) available for scheduling; and

communicate a return to the management engine (130, 515) identifying one or more nodes (115) in the selected spatial, compact, or nonspatial and noncompact combination of nodes (115) in the cluster (220) for executing the job (150)."

IX. Claim 1 of the second auxiliary claim request reads as follows:

"1. Logic for scheduling in a high-performance computing (HPC) system, the logic encoded in a computer-readable medium and when executed operable to:

receive a call from a management engine (130, 515) operable to manage a cluster (220) of nodes (115) in the HPC system (100), the call specifying a request comprising a job (150) for scheduling comprising one or more processes for execution at one or more nodes (115) in the cluster (220), the call further specifying a number of nodes (115) for executing the job (150) and an aggressive flag for indicating a degree of leeway for scheduling nodes to a job when selecting a spatial combination of nodes, a compact combination of nodes, or a non-spatial and non-compact combination of nodes;

determine whether the request is spatial, compact, or nonspatial and noncompact, according to a request type in the request, the request being spatial if the job (150) assumes spatial relationships between nodes (115) executing the job (150), the request being compact if the job (150) assumes proximity between nodes (115) executing the job (150), the request being nonspatial and noncompact if the job (150) assumes no spatial relationships or proximity between nodes (115) executing the job;

if the request is spatial:

generate spatial combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call and further accommodating the assumed spatial relationships between nodes (115) executing the job (150); and

select one of the spatial combinations that is schedulable according to a list of nodes (115) in the cluster (220) available for scheduling; and

if the nodes cannot accommodate the spatial request and the aggressive flag indicates a degree of leeway in the allocation of nodes to the spatial request by defining a maximum hop count between pairs of nodes when the request is allocated as a compact request, scheduling the spatial request as a compact request

if the request is compact:

generate compact combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call which comprise combinations of nodes that minimize the maximum communication distance or hop count between each pair of nodes; and

select one of the compact combinations that is schedulable according to the list of nodes (115) in the cluster (220) available for scheduling and that minimizes a maximum hop count between each pair of nodes (115); and

allocating the compact combination of nodes to the request if the maximum hop count is within the limit on hop counts set by the aggressive flag;

if the request is nonspatial and noncompact:

identify one or more nodes (115) schedulable according to the list of nodes (115) in the cluster (220) available for scheduling;

generate a nonspatial and noncompact combination of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call, the nonspatial and noncompact combination comprising one or more of the one or more identified nodes (115) that are schedulable according to the list of nodes (115) in the cluster (220) available for scheduling; and

communicate a return to the management engine (130, 515) identifying one or more nodes (115) in the selected spatial, compact, or nonspatial and noncompact combination of nodes (115) in the cluster (220) for executing the job (150)."

X. Claim 1 of the third auxiliary claim request reads as follows:

"1. Logic for scheduling in a high-performance computing (HPC) system, the logic encoded in a computer-readable medium and when executed operable to:

receive a call from a management engine (130, 515) operable to manage a cluster (220) of nodes (115) in the HPC system (100), the call specifying a request comprising a job (150) for scheduling comprising one or more processes for execution at one or more nodes (115) in the cluster (220), the call further specifying a number of nodes (115) for executing the job (150);

determine whether the request is spatial, compact, or nonspatial and noncompact, according to a request type in the request, the request being spatial if the job (150) assumes spatial relationships between nodes (115) executing the job (150), the request being compact if the job (150) assumes proximity between nodes (115) executing the job (150), the request being nonspatial and noncompact if the job (150) assumes no spatial relationships or proximity between nodes (115) executing the job;

if the request is spatial:

generate spatial combinations of nodes (115) in the cluster (220) accommodating the number of nodes

(115) specified in the call and further accommodating the assumed spatial relationships between nodes (115) executing the job (150); and

select one of the spatial combinations that is schedulable according to a list of nodes (115) in the cluster (220) available for scheduling;

if the request is compact:

generate compact combinations of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call; and

select one of the compact combinations that is schedulable according to the list of nodes (115) in the cluster (220) available for scheduling and that is more compact than other compact combinations that are schedulable according to the list of nodes (115) in the cluster (220) available for scheduling, a compact combination being more compact than other compact combinations when it has a smallest maximum communication distance, or hop count, between each pair of nodes (115;

if the request is nonspatial and noncompact:

identify one or more nodes (115) schedulable according to the list of nodes (115) in the cluster (220) available for scheduling;

generate a nonspatial and noncompact combination of nodes (115) in the cluster (220) accommodating the number of nodes (115) specified in the call, the nonspatial and noncompact combination comprising one or more of the one or more identified nodes (115) that are schedulable according to the list of nodes (115) in the cluster (220) available for scheduling; and

communicate a return to the management engine (130, 515) identifying one or more nodes (115) in the selected spatial, compact, or nonspatial and noncompact combination of nodes (115) in the cluster (220) for executing the job (150);

wherein the call from the management engine (130, 515) further specifies an aggressive flag indicating a degree of leeway allotted for selecting a spatial combination, a compact combination, or a nonspatial and noncompact combination of nodes (115) in the cluster (220) for executing the one or more processes in the job (150);

the logic being operable to select a spatial combination, a compact combination, or a nonspatial and noncompact combination of nodes in the cluster (220) for executing the one or more processes in the job (150) according to the aggressive flag specified in the call from the management engine (130, 515)."

Reasons for the Decision

1. *Overview of the invention*

The application relates to scheduling a job request in a high-performance computing (HPC) system which consists of a cluster of computing nodes (e.g. blade servers or PCs; see original description page 7, paragraph 2). A schedule is a combination of nodes to be allocated to satisfy the request. When the user submits a job request, he also has to specify its request type (page 23, lines 17-20: "spatial", "compact" or "any"; in the claims: "spatial", "compact" and "non-spatial and non-compact"; see also original claim 2). A spatial request may require a line of nodes (i.e. a one dimensional request), a two-dimensional or a three-dimensional grid (page 23, line 20 to page 24, line 7). A compact request does not have spatial specifications, but requires the maximal hop count (i.e. the communication distance) between each pair of nodes of a schedule to be minimal among all possible schedules (page 24, lines 20-23). An "any" or "non-spatial and non-compact" request merely requires that the requested number of nodes be available to be scheduled (page 24, lines 23-26).

A user may also add a so-called "aggressive flag" to his request (page 24, line 27 to page 25, line 15). This is a floating point number between zero and one and indicates a degree of leeway for the scheduler to allocate nodes (page 24, lines 28-30). If the aggressive flag is zero, then a spatial request is strictly scheduled as a spatial request (page 25, lines 1-4). If the aggressive flag is greater than zero, a spatial request is treated like a compact request if it cannot be fulfilled as a spatial request (lines 4-7 and page 26, lines 23-24). For compact requests (either original ones or "relaxed" spatial ones), the aggressive flag ("a") is used to impose a limit on the maximal hop counts between pair of nodes according to the formula "1/(1-a)" (page 25, lines 12-15).

2. *Overview of the decision*

2.1 The claims of all requests are clear (Article 84 EPC 1973).

2.2 The request for remittal to the first instance according to Article 111(2) EPC 1973 is refused.

2.3 Claim 1 of all requests lacks an inventive step (Article 56 EPC 1973).

3. *Clarity*

3.1 The board does not agree with the appealed decision (2.1, 2.2) that the expression "generate compact combinations of nodes" as used in claim 1 of all requests is unclear. This results from the fact that it is immaterial how "compact" is defined and which "compact" schedules (meaning combinations of nodes to be allocated to the request) are generated, since one of the *minimal* schedules with respect to the hop counts is selected and outputted. This can be seen from the following.

3.2 The essence of claim 1 of all requests can be summarised as follows:

- receive a request together with a type (either "spatial", "compact" or "non-spatial and non-compact");
- if type == "spatial" then generate spatial schedules satisfying the request and select one;
- if type == "compact" then generate compact schedules satisfying the request and comprising the schedules where the maximal hop count between each pair of nodes is minimal, and select one of the minimal schedules;
- if type == "non-spatial and non-compact" then generate any schedule satisfying the request;
- output the schedule.

3.3 In the case of the type "compact", the minimal schedules (with respect to the hop counts) are generated amongst other schedules, and then one of the minimal ones is selected. It follows that the generation of other than the minimal schedules has the only purpose of searching for the minimal ones. Such a search has to be done among *all* schedules satisfying the request in order to find the minimal ones.

3.4 In any case the board understands "compact" to be a relative term, meaning (relatively) short communication distances (i.e. hop counts) in the schedule. Therefore, any schedule is (more or less) compact.

3.5 Thus, the step of generating compact combinations which comprise minimal combinations comprises all supersets of the minimal schedules as possible interpretations.

3.6 Therefore, claim 1 of all requests is clear.

4. *Remittal*

According to Article 111(1) EPC 1973, the board has to examine the allowability of the appeal and then has discretion either to exercise any power within the competence of the department which was responsible for the decision appealed or to remit the case to that department for further prosecution. The board agrees with the appellant that procedures before the EPO are designed so that issues are normally decided by two instances. As follows from Article 111(1) EPC 1973, however, this does not give the parties an absolute right to two instances. In the case at hand, the appellant requested a remittal of the case to the first instance according to Article 111(2) EPC 1973 for three reasons.

4.1 The first reason is that the ground for refusal (lack of clarity, Article 84 EPC) is considered by the board to be overcome. The appellant would prefer to have the matter of inventive step considered in substance by both instances (letter dated 24 March 2016, page 1, last paragraph).

4.1.1 However, the objections concerning inventive step over D1 are not new. They were already raised in the first communication of the examining division dated 26 July 2006, and maintained in the second communication dated 2 February 2007, the summons to oral proceedings before the examining division and the decision (sections 5 and 6). Thus, there was a detailed consideration of inventive step by the first instance. The board considers that no purpose would be served by remitting a case for further prosecution based on a request that was held unallowable by the examining division.

4.2 The second reason according to the appellant is that further investigation about D3 had to be done by the examining division. D3 is marked as "Interner Bericht Nr. 20/92" (i.e. "internal report") at the bottom of page 1. Therefore, the appellant contests that D3 was publicly available and argues that D3 does not belong to the prior art.

4.2.1 Before the oral proceedings, the board performed a limited investigation in the Internet about D3 and presented its results to the appellant during the oral proceedings. The board found one United States patent (US 8,190,714 B2, page 3, right column) and three scientific papers citing D3. Furthermore, D3 is listed in the online catalogue of the Deutsche National-bibliothek and is available in its reading rooms in Frankfurt and Leipzig with a number (Signatur) from 1993 (see http://d-nb.info/931177162), i.e. well before the priority date of the application (2004). Thus, the board maintains its opinion that in the field of universities "Interner Bericht" does not mean confidential, but something like "technical report", and that D3 was publicly available prior art at the priority date.

4.3 The third reason according to the appellant is that a significant change in the case had emerged. Now, D3 is considered as closest prior art, instead of D1 as in the obiter dictum of the decision.

4.3.1 This turned out during oral proceedings to be a misunderstanding. The board never stated that it took D3 as closest prior art, but wrote for example in the summons (6.1) that it considered the reasoning of the decision sound and convincing and that it wanted to add some reasons which confirmed the decision. Thus, the board considers D1 to be a suitable starting point for assessing inventive step.

4.4 Therefore, the board exercises its discretion and decides not to remit the case to the first instance.

5. *Inventiveness*

5.1 The decision gives a complete inventive step argumentation in its obiter dicta (5.-7.2). The board considers it to be sound and convincing. Furthermore, the board adds the following reasons confirming the decision.

5.2 Concerning the three different types of requests, these are apparently all known in the field, as demonstrated in the decision. Section 5.3 refers to D3 for spatial and compact requests. Indeed, D3, page 16, section 2.3, first paragraph discloses a two-dimensional spatial request (called "shaped" in D3) and an "unshaped" request with an additional degree of freedom (called "scalar"). The decision identifies "scalar" requests with "compact requests".

5.3 The board is of the opinion that D3 also discloses how to compute schedules for "compact" requests, namely the search for minimal schedules with respect to compactness, see D3, page 17, lines 10-14: "compactness optimal". In this definition, the diameter of the territory of an allocation corresponds to the hop count of the schedule in the claim. First, the maximum diameter of a territory of a schedule and then the minimum of them over all schedules is computed. This is exactly what the claims describe as "minimiz[ing] the maximum communication distance or hop count between each pair of nodes".

5.4 The third type, "non-spatial and non-compact", is merely a complicated formulation of imposing no constraints on the schedule. It collects all other requests. Therefore, this type is called "any" in the description. It is thus an obvious option.

5.5 Therefore, the board does not agree with the appellant that the use of these three request types establishes an inventive step (see grounds, page 2, end of second paragraph).

5.6 As to the auxiliary claim requests, they contain the following additional step:

- if type == "spatial" and request not spatially allocatable and aggressive flag indicates "leeway" then type := "compact";

5.7 The appellant argued that there is no disclosure in the prior art about an aggressive flag, nor about there being any leeway. This gives the user a degree of freedom about how strict the requirements are. The user does not need to submit a second request to the scheduling program, since he can express his wish in one request.

5.8 In the board's view, this constitutes an over-ruling of one user-input information ("type") by another user-input information ("aggressive flag"). The board does not consider this over-ruling to produce a technical effect establishing an inventive step. The mere idea to offer more input options in a program does not per se produce a technical effect in comparison with a prior program. Whether an input is formulated in one part or in two parts is an organisational matter and does not contribute to the technical character of the present invention.

5.9 Additionally, the second auxiliary claim request contains the following step:

- if type == "compact" and maximum hop count is above the limit set by the aggressive flag then exit without outputting a schedule;

5.10 Again, a user-input information ("limit set by the aggressive flag") is used to express which schedule the user really wants. The board is not of the opinion that such a possibility for the user to input its wishes would produce a technical effect and establish an inventive step.

5.11 Therefore, claim 1 of all requests is not inventive.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar:	The Chairman:
B. Atienza Vivancos	W. Sekretaruk

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

Decision electronically authenticated