Intellectual Property Office

BL O/521/14

08 December 2014

PATENTS ACT 1977

APPLICANT	Intel Corporation
ISSUE	S1(1)(c), (d), S1(2)(c), (d), S4(1)
HEARING OFFICER	Ben Buchanan

DECISION

Introduction

1 This decision concerns whether the inventions claimed in two patent applications relate to excluded matter and whether they are capable of industrial application.

2 Patent applications GB1020675.3 and GB1020734.8 were filed on 6 December 2010 and 7 December 2010 respectively in the name of Intel Corporation. The applications, entitled “Relational modelling for performance analysis of multi-core processors” and “Relational modelling for performance analysis of multi-core processors using virtual tasks” respectively, both claim a priority date of 23 December 2009. The applications were published on 29 June 2011 as GB2476544 A and GB2476548 A.

3 Following rounds of correspondence, the applicant has been unable to convince the Examiner that the invention as claimed in either application is not excluded from patentability under section 1(2) of the Patents Act 1977. In each case the Examiner considered the claims to define a program for a computer and the presentation of information. A joint hearing for both applications, and a single decision, was requested on 10 February 2010.

4 In due course the applicant requested a decision on the papers instead of a hearing and on 3 October 2014 a joint skeleton argument relating to both applications was submitted. The arguments put forward in the joint skeleton argument refer mainly to GB1020675.3 and by extension to GB1020734.8. The applicant acknowledges that corresponding method-system claims of each application stand or fall together. I will also address the skeleton arguments in respect of each application jointly.

The inventions

5 Both claimed inventions relate to performance analysis of multi-core processors when executing code. In particular, they provide a means of visualising the temporal sequence and the relationships between tasks and other primitives, such as parameters, running on a multi-core processor using trace or functional analysis. The

inventions work by there being two versions of the code being developed. Alongside the conventional executable code is a version called “instrumented code” which includes a number of Application Program Interfaces (APIs). The APIs provide trace information about the tasks running in threads which is processed and displayed on a graphical user interface (GUI). By visualising the tasks and relationships, a software developer may identify and make improvements to the way the code runs by reprogramming it.

6 The applicant’s skeleton argument acknowledges that GB1020734.8 shares common subject matter with, and is more limited than, GB1020675.3. It therefore makes sense to consider the skeleton argument first in respect of GB1020675.3, and then assess whether my consideration of GB1020734.8 is any different, at each stage.

The invention of GB1020675.3

7 This invention is characterised by presenting in a graphical user interface a visualisation of the *functional relationships* of one or more of said tasks, such as parent-child, sibling, dependency or data producer-consumer. In general these relationships represent how one task is created and/or related to another. The visualisation is produced by processing trace information which represents a functional relationship.

The invention of GB1020734.8

8 However not all tasks can be readily visualised using the above techniques, such as in rendering passes in graphics processing. This invention is characterised by enabling a user to select such a task and view a representation of its *duration*, based on the trace information. It works by creating a *virtual task* which identifies at least one associated task for which trace information is generated. The duration of the *virtual task* is determined by the duration of the respective associated tasks and is displayed to a user.

The law

9 The examiner has raised an objection that the invention is excluded from patentability under sections 1(1)(d), 1(2)(c) and 1(2)(d). In their skeleton arguments, the applicants have advanced a new argument under section 1(1)(c) and section 4(1).

10 The relevant provisions are provided below:

Section 1(1)
A patent may be granted only for an invention in respect of which the following conditions are satisfied, that is to say -
(a) the invention is new;
(b) it involves an inventive step;
(c) it is capable of industrial application;
(d) the grant of a patent for it is not excluded by subsections (2) and (3) or section 4A below;
and references in this Act to a patentable invention shall be construed accordingly.

Section 1(2)
It is hereby declared that the following (among other things) are not inventions for the purposes of this Act, that is to say, anything which consists of -
(a) a discovery, scientific theory or mathematical method;
(b) a literary, dramatic, musical or artistic work or any other aesthetic creation whatsoever;
(c) a scheme, rule or method for performing a mental act, playing a game or doing business, or a program for a computer;
(d) the presentation of information;
but the foregoing provision shall prevent anything from being treated as an invention for the purposes of this Act only to the extent that a patent or application for a patent relates to that thing as such.

Section 4(1)
An invention shall be taken to be capable of industrial application if it can be made or used in any kind of industry, including agriculture.

The test for patentability under section 1(2)

11 The starting point for determining whether an invention falls within the exclusions of section 1(2) is the judgment of the Court of Appeal in *Aerotel/Macrossan*[1].

12 The interpretation of section 1(2) was further considered by the Court of Appeal in *Symbian*[2]. *Symbian* arose under the computer program exclusion, but as with its previous judgment in *Aerotel/Macrossan*, the Court gave general guidance on section 1(2). Although the Court approached the question of excluded matter primarily on the basis of whether there was a technical contribution, it nevertheless (at paragraph 59) considered its conclusion in the light of the *Aerotel/Macrossan* approach. The Court was quite clear (see paragraphs 8-15) that the structured four-step approach to the question in *Aerotel/Macrossan* was never intended to be a new departure in domestic law; that it remained bound by its previous judgments, particularly *Merrill Lynch*[3] which rested on whether the contribution was technical; and that any differences in the two approaches should affect neither the applicable principles nor the outcome in any particular case. The *Symbian* judgment does however make it clear, that in deciding whether an invention is excluded, one must ask does it make a technical contribution? If it does then it is not excluded.

13 Therefore, subject to the clarification provided by *Symbian*, it is appropriate for me to proceed on the basis of the four-step approach set out in *Aerotel/Macrossan* namely:

1) Properly construe the claim.

2) Identify the actual contribution (although at the application stage this might have to be the alleged contribution).

3) Ask whether it falls solely within the excluded matter.

---

[1] Aerotel Limited v Telco Holdings Limited & Ors / Macrossan's Application [2007] RPC 7
[2] Symbian Ltd v Comptroller-General of Patents [2009] RPC 1
[3] Merrill Lynch's Application [1989] RPC 561

4) If the third step has not covered it, check whether the actual or alleged contribution is actually technical.

14 The operation of this test is explained at paragraphs 40-48 of the judgment. Paragraph 43 confirms that identification of the contribution is essentially a matter of determining what it is the inventor has really added to human knowledge, and involves looking at substance, not form. Paragraph 46 explains that the fourth step of checking whether the contribution is technical may not be necessary because the third step should have covered the point.

The AT&T/CVON signposts

15 In *Symbian*, the Court of Appeal stated that a computer program may not be excluded if it makes a technical contribution. The judgment in *AT&T/CVON*[4] provided guidance in the form of a number of signposts which may indicate that a computer program provides a technical contribution. The signposts were updated in *HTC/Apple*[5] and are considered as follows:

i) whether the claimed technical effect has a technical effect on a process which is carried on outside the computer;

ii) whether the claimed technical effect operates at the level of the architecture of the computer; that is to say whether the effect is produced irrespective of the data being processed or the applications being run;

iii) whether the claimed technical effect results in the computer being made to operate in a new way;

iv) whether the program makes the computer a better computer in the sense of running more efficiently and effectively as a computer;

v) whether the perceived problem is overcome by the claimed invention as opposed to merely being circumvented.

The claims

16 The applicant has requested that I should base this decision on the amended claim sets filed on 10 September 2014. There are four independent claims in each application which can be grouped as follows:

GB1020675.3

17 Claims 1 and 9 relate to a computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, and a computer system adapted to

[4] AT&T Knowledge Ventures LP and Cvon Innovations Ltd v Comptroller General of patents [2009] EWHC 343 (Pat)
[5] HTC Europe Co Ltd v Apple Inc [2013] EWCA Civ 451

provide a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code.

18 Claims 8 and 11 relate to a computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing code, and a computer system adapted to provide a performance analysis tool for analysing the performance of a multi-core processor when executing code. Claims 8 and 11 do not specify graphics processing code.

19 For brevity I repeat only claim 1 here. The remaining impendent claims can be found in the annex.

> *A computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the method comprising steps of;*
>
> *executing in the multi-core processor a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks, wherein the identified functional relationships between tasks comprise at least one of:*
>
> *a) a parent-child relationship whereby a child task is created by a parent tasks;*
> *b) a sibling relationship whereby a child task is related to another task;*
> *c) a dependency relationship whereby one task is dependant for execution upon inputs from another task; and*
> *d) a producer and consumer relationship whereby one task produces data placed in a buffer and the data is consumed by another task;*
>
> *processing the trace information to provide in a graphic user interface a representation of the threads and the tasks;*
>
> *receiving a user selection via the graphical user interface of one of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of the functional relationship between the selected task in one of said thread and one of more other tasks in another of said threads.*

GB1020734.8

20 An administrative error seems to have occurred in the filing of the latest claims resulting in independent claim 13 being incomplete. I have based my assessment of claim 13 on the "tracked changes" version of this claim which was helpfully filed along with the formal replacements. There are four independent claims which can be grouped as follows:

21 Claims 1 and 11 relate to a computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, and a computer system adapted to provide a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code.

22 Claims 10 and 13 are similar to claims 1 and 11 but do not specify graphics processing code.

23 Again, I repeat only claim 1 here. The remaining independent claims can be found in the annex.

> *A computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the method comprising steps of;*
>
> *executing in the multi-core processor instrumented code comprising a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks,*
>
> *processing the trace information to display in a graphic user interface a representation of the threads and the tasks executing in the processor;*
>
> *receiving a user selection of one or more of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of each selected task including a representation of a duration of each selected task, the duration of each selected task being determined according to begin and end times in the trace information;*
>
> *the instrumented code further comprising code which creates at least one virtual task and identifies one of more tasks as being associated tasks, each associated task comprising one of the tasks for which trace information is generated, the virtual task having a duration determined by begin and end times of the one or more of the associated tasks; and*
>
> *displaying a visualization of the virtual task including a representation of the duration of the virtual task.*

Arguments and analysis

24 There appear to be two lines of argument put forward by the applicant in respect of both applications. Firstly, it is maintained that the claims relate to a method of testing applicable to the improvement or control of a graphics engine and because both graphics engines and methods of testing are capable of industrial application, this alone should be sufficient to "take these claims out of the exclusions". Secondly, it is argued that the invention makes a technical contribution.

Industrial applicability

25 The first line of argument relies on the Manual of Patent Practice at sections 4.06 and 4.03 and quotes as follows:

> *4.06 Methods of testing are generally regarded as capable of industrial application if the test is applicable to the improvement or control of a product, apparatus or process which itself is capable of industrial application.*
>
> *4.03 In many cases, an invention which passes (or fails) the test for industrial application under s.4(1) will equally pass (or fail) the test for patentability under s.1(2)....*

26 However, section 4.03 of the Manual goes on to give further guidance as follows (my emphasis added):

It should be remembered however that, even though they will frequently give the same answer, ***the tests for industrial application and patentability are separate and independent.***

27 The argument put forward by the applicant appears to overlook the fact that the requirement for an invention to be capable of industrial application is a separate requirement to that for the invention not to be excluded by sections 1(2) and 1(3) and 4A of the Act.

28 Whilst I accept that methods of testing can be industrially applicable, it is not necessary for me to determine whether the claims of the applications in hand relate to a method of testing which is capable of industrial application. The examiner has raised no objection under section 1(1)(c). Instead, I must independently consider whether the claims define an invention which provides a technical contribution, to decide whether the invention is excluded from patentability under sections 1(1)(d) and 1(2).

Application of the Aerotel test

29 To determine whether the claimed inventions are excluded from patentability under section 1(2) of the Act, I will consider GB1020675.3 and GB1020734.8 jointly, commenting separately where necessary. I will consider the independent claims of each application, noting that the only difference in substance between the method-system claim pairs for each application is limited to executing graphics processing code.

Step 1: Properly construe the claim

GB1020675.3

30 Construing the claims presents no problems. Claim 1 defines a computer-implemented method of providing a performance analysis tool. The performance analysis tool is suitable for analysing the performance of a multi-core processor when executing graphics processing code. The method executes a version of the code which includes a number of APIs adapted to provide trace information identifying threads running in the processor, tasks running within respective threads and identifying functional relationships between tasks. The method is characterised by presenting on a graphical user interface a visualisation of the functional relationships of one or more of said tasks. It is noted that the method executes and analyses the instrumented code relating to the (graphics) code not the actual (graphics) code itself. The other independent claims may be similarly substantively construed, although claims 8 and 11 are not limited to graphics processing code.

GB1020734.8

31 Again the claims may be readily construed. In this case claim 1 is characterised by enabling a user to select a task and view a representation of its duration, based on trace information provided by an API. It works by creating a virtual task which identifies at least one associated task for which trace information is provided. The duration of the virtual task is determined by the duration of the respective associated tasks and is displayed to a user. Again, the other independent claims may be

similarly substantively construed, although claims 10 and 13 are not limited to graphics processing code.

Step 2: Identify the actual contribution (although at the application stage this might have to be the alleged contribution)

GB1020675.3 & GB1020675.3

32 In the skeleton argument, the applicant has identified the contribution as being twofold: Primarily, the contribution is concerned with the detection of conditions prevailing within the computer system, including relationships between tasks, and displaying these conditions representing the performance of the multi-core processor when executing code. Secondly, the limitation in respective relevant claims to the execution of graphics processing code ties the claim to graphics processing which itself is recognised as a technical process.

33 The applicant argues that the invention generates an output indicative of conditions arising within the computer system at the level of the multi-threaded, multi-core architecture and that the invention performs testing at the level of the multi-threaded, multi-core architecture of the computer system. This would appear to be aimed at aligning the present application with EPO Board of Appeal decisions T 85/0115 (*IBM*) and T 05/0717 (*Labtronix*) which were found to solve a technical problem and provide a technical contribution by detecting and displaying information about prevailing conditions in a computer. Moreover (argues the applicant), in the present case the detection and display is of conditions at the level of the architecture and relates to tasks and their relationships which are essentially technical in nature regardless of the nature of the process.

34 Despite not being bound by EPO Board of Appeal decisions[6] it would appear prudent for me to briefly consider *IBM* and *Labtronix*, in case their respective inventions provide a similar contribution to that of either of the present inventions, and are persuasive in my determination at step three of the test.

35 *IBM* is concerned with decoding stored phrases and obtaining a read-out of events in a text processing system. The method, which gave visual indications automatically about conditions prevailing in the apparatus/system, was held to be technical by the Board.

36 *Labtronix*, which cited *IBM*, is concerned with the conditions within a game and was regarded as not excluded even though the rules of games are excluded as such. The contribution related to informing a gaming machine player about past wins and losses in the gaming machine and in so doing, the Board found, to display the internal state of the apparatus. This was likened to a display of the internal state of apparatus in a more classical field, such as displaying the temperature of an internal combustion engine or the pressure of an autoclave.

Does the invention relate to detecting conditions in a computer system?

[6] Manual of Patent Practice section 1.06

37 In the present applications, the invention generates a visual representation of conditions "arising" when a graphics program under development is executed in accordance with the invention at the level of multi-threaded, multi-core processing. These conditions "arise" entirely in accordance with the rules and parameters defined by the graphics program. In fact the description[7] describes how the software designer configures the relationships between tasks (and the description also states that they may be determined automatically by code analysis). The present inventions display information about the state of instrumented code that is running on the apparatus. The conditions prevailing are determined in accordance with programmed code and do not reflect the internal state of a computer. Therefore this is not the same as determining the "internal state" of apparatus on the basis of past events. That the tasks run in a multi-threaded environment on a multi-core processor is immaterial; the underlying architecture is unchanged.

38 What the inventions do then, is display information about tasks and processes running in accordance with their programmed configuration. In other words, the output is a visual representation of how tasks under the control of the graphics processing code exist and relate in accordance with the way they are programmed, at the level of the processor where they are processed. As the opening paragraphs of the description state, "analyzing the performance of a processor may involve a complex undertaking, given the number of tasks and the number of different threads that may be running". It is this complex undertaking that the inventions perform and present the results of to a user. This is very different from the problem solved by the invention in *IBM* in which storage requirements and response time for decoding phrases were improved by a new way of presenting text messages to a user to indicate conditions such a disk drive door being open and other events and errors.

Does the invention relate to graphics processing?

39 The applicant has argued that the inventions relate to the improvement or control of a graphics processing system and that it is widely accepted that graphics processing is a technical process which is not excluded from patentability. In arguing that the claims relate to graphics processing, the applicant would appear to be aligning the present application with EPO Board of Appeal Decision T 028/84 (*Vicom*).

40 In *Vicom*, the Board of Appeal held that a claim directed to a technical process carried out under the control of a computer program was not excluded from patentability. The claimed method of digitally processing image data was held to be a technical process in which a mathematical method was used (rather than being a claim to a mathematical method as such). This was because the process was carried out on a physical entity (an image stored as an electric signal) by technical means and resulted in a certain change in that entity (see paragraph 5 of the judgment). Since it was held to be a technical process for these reasons, the claimed method was held to amount to more than a program for a computer as such.

41 In the present applications, where the claims are limited to executing graphics processing code, any improvement is to the analysis and display of information relating to a graphics processing program under development. The claims do not define an improvement to a graphics processing system or apparatus per se. A

[7] GB1020675.3 on page 5 at lines 12-14; GB1020734.8 on page 4 lines 7-9

software engineer may act on the information presented by the method of the current claims, and make unspecified improvements to the graphics processing program, but no improvements to the graphics processing program itself are defined by any claim. *Vicom* provides that if an image processing system or apparatus provides a technical contribution then it satisfies section 1(2)(c). The current claims define no such contribution because they do not define the image processing system or apparatus itself, or specify any improvement thereto.

So what is the contribution?

42 I consider that the contribution provided by both inventions involves displaying relationships between tasks and other primitives in code. The relationships are identified by APIs within a special "instrumented" version of the program code and may be configured manually or automatically by analysing the code. The contribution defined by the claims is not the detection of conditions prevailing within an apparatus or system, but the analysis and display of functional relationships, predefined in a special version of program code, when running on a multi-core processor. The contribution in both cases is different to that defined by the inventions of *IBM* and *Labtronix.* I therefore need consider these decisions no further.

43 Whilst it is clear that a software developer could use either invention to visualise information which would inform decisions about making changes that could lead to subsequent enhancements to the software, any such benefits would depend on the skill and discretion of the developer. They are certainly not an inherent feature of the inventions defined by the claims, and so the limitation of either invention to "graphics processing" code does not make it "technical".

GB1020675.3

44 When considering each claim as a whole, the contribution is an improved tool suitable for analysing the performance of code during execution on a multi-core processor by identifying conditions including temporal and functional relationships that exist between different tasks within the code. The tool processes and displays these functional relationships on a graphical user interface to allow a user to visualise the relationships and make decisions as to how to enhance the code. The code may be graphics processing code (as defined in claims 1 and 9).

GB1020734.8

45 The difference between the two applications is summarised at page 10 of the description:

"In short, virtual tasks may be useful for building profiling tools for complex, highly parallel software. Without them, the profiling tool may not be able to show anything more than the mechanism executing an API, which may prevent gaining simple and critical performance insights in some cases. Using virtual tasks, an abstraction is obtained that allows such issues to be more clear from the start, enabling more productive and efficient top down profiling, in some embodiments."

46 The contribution is fundamentally similar to that of GB1020675.3 and specifically uses instrumented code to create virtual tasks which enable the duration of one or

more associated tasks to be visualised. Again, the code may be graphics processing code (claims 1 and 11).

Steps 3 and 4: Does the contribution fall solely within excluded subject matter? Is the contribution technical in nature?

Program for a computer

47 As both contributions involve a computer program, I must decide whether the program provides a relevant technical effect or relevant technical contribution so as to satisfy the requirements of section 1(2) for either invention.

48 Clearly the invention is implemented in software; the apparatus is not new. In order to assist my consideration of whether or not the invention is more than a program for a computer *as such*, I shall apply the *AT&T/CVON* signposts. The applicant has argued that the invention satisfies signposts (ii) and (iv), but I shall consider them all.

i) whether the claimed technical effect has a technical effect on a process which is carried on outside the computer;

The computer program does not produce a technical effect outside a computer because the result merely analyses and presents information on a GUI about program code primitives running within a development environment on the computer.

ii) whether the claimed technical effect operates at the level of the architecture of the computer; that is to say whether the effect is produced irrespective of the data being processed or the applications being run;

The architecture of a computer on which the invention may be implemented comprises a multi-core processor to enable multi-threaded execution of tasks from program code. Contrary to the applicant's argument, this does not suggest a technical effect at the level of the architecture of the computer; it is conventional operation of the architecture. The invention is implemented in special "instrumented" program code which uses APIs to provide trace information about tasks which is then processed to provide a visualisation. The effect of processing and displaying information about relationships between tasks is produced entirely based on processing the data (trace information) provided by the APIs and the instrumented code. The architecture and the operating system are conventional and do not operate differently because of a technical effect.

iii) whether the claimed technical effect results in the computer being made to operate in a new way;

The computer enables the invention to work by executing instrumented code and processing and analysing trace information. As I said above, the computer itself does this by operating in a conventional, not a new, way.

iv) whether the program makes the computer a better computer in the sense of running more efficiently and effectively as a computer;

The computer operates conventionally and no more efficiently and effectively. The applicant argues that the purpose of *testing* the system is to improve it and make it more efficient. When I construed the claim I did not find this to be the invention defined by the claims. Even if I had construed the claim as a method of testing, the invention only presents information to a software developer who may exercise discretion in making changes to the normal and the instrumented code. Any improvement which might arise from those changes is attributable to his or her skill and is not within the scope of the claimed invention.

v) whether the perceived problem is overcome by the claimed invention as opposed to merely being circumvented.

As the description relates, the problem is that analysing the performance of a multi-core processor executing multi-threaded tasks is complex. However this is not a technical problem[8]. In this case the problem is addressed by processing and displaying information to a user. No technical problem is solved, therefore this signpost is not met.

49 The contributions do not meet the requirements of any of the signposts. They do not provide a contribution to a graphics processor per se; rather any contribution is to the visualisation of information about the way relationships between tasks exist when program code under development is executed. The claims, then, would seem to define a program for a computer as such.

Presentation of information

50 The inventions present information in a GUI relating to the tasks and relationships running in a development environment on a multi-core processor. The information is characterised by displaying the relationships between tasks, including their duration, which are running in a multi-threaded environment. The contributions are characterised by the content of the data displayed and therefore relate to the presentation of information *as such*.

Conclusion

51 Consequently, there is no doubt in my mind that the contribution of each invention consists solely of excluded matter. Whether the executed code which is analysed and visualised is graphics processing code is immaterial to my finding because none of the claims define a technical contribution such as that of *Vicom*.

52 I find the claims of GB1020675.3 and GB1020734.8 to be excluded from patentability as they each relate solely to a program for a computer and the presentation of information as such. Having read the specifications in full, I do not think that any saving amendment is possible. I therefore refuse both applications under section 18(3).

Appeal

53 Any appeal must be lodged within 28 days

[8] Manual of Patent Practice 1.38.5

Ben Buchanan

Deputy Director, acting for the Comptroller

ANNEX: The independent claims

GB1020675.3

Claim 1

A computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the method comprising steps of;

executing in the multi-core processor a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks, wherein the identified functional relationships between tasks comprise at least one of:

a) a parent-child relationship whereby a child task is created by a parent tasks;
b) a sibling relationship whereby a child task is related to another task;
c) a dependency relationship whereby one task is dependant for execution upon inputs from another task; and
d) a producer and consumer relationship whereby one task produces data placed in a buffer and the data is consumed by another task;

processing the trace information to provide in a graphic user interface a representation of the threads and the tasks;

receiving a user selection via the graphical user interface of one of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of the functional relationship between the selected task in one of said thread and one of more other tasks in another of said threads.

Claim 8

A computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing code, the method comprising steps of;

executing in the multi-core processor a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks, wherein the identified functional relationships between tasks comprise at least one of:

a) a parent-child relationship whereby a child task is created by a parent tasks;
b) a sibling relationship whereby a child task is related to another task;
c) a dependency relationship whereby one task is dependant for execution upon inputs from another task; and
d) a producer and consumer relationship whereby one task produces data placed in a buffer and the data is consumed by another task;

processing the trace information to provide in a graphic user interface a representation of the threads and the tasks;

receiving a user selection via the graphical user interface of one of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of the functional relationship

between the selected task in one of said thread and one of more other tasks in another of said threads.

Claim 9

A computer system adapted to provide a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the system adapted to;

execute in the multi-core processor a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks, wherein the identified functional relationships between tasks comprise at least one of:

a) a parent-child relationship whereby a child task is created by a parent tasks;
b) a sibling relationship whereby a child task is related to another task;
c) a dependency relationship whereby one task is dependant for execution upon inputs from another task; and
d) a producer and consumer relationship whereby one task produces data placed in a buffer and the data is consumed by another task;

process the trace information to provide in a graphic user interface a representation of the threads and the tasks;

receive a user selection via the graphical user interface of one of the tasks and, in response to receiving said user selection, process the trace information to provide in the graphic user interface a visualization of the functional relationship between the selected task in one of said thread and one of more other tasks in another of said threads.

Claim 11

A computer system adapted to provide a performance analysis tool for analysing the performance of a multi-core processor when executing code, the system adapted to;

execute in the multi-core processor a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks, wherein the identified functional relationships between tasks comprise at least one of:

a) a parent-child relationship whereby a child task is created by a parent tasks;
b) a sibling relationship whereby a child task is related to another task;
c) a dependency relationship whereby one task is dependant for execution upon inputs from another task; and
d) a producer and consumer relationship whereby one task produces data placed in a buffer and the data is consumed by another task;

process the trace information to provide in a graphic user interface a representation of the threads and the tasks;

receive a user selection via the graphical user interface of one of the tasks and, in response to receiving said user selection, process the trace information to provide in the graphic user interface a visualization of the functional relationship between the selected task in one of said thread and one of more other tasks in another of said threads.

GB1020734.8

Claim 1

A computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the method comprising steps of;

executing in the multi-core processor instrumented code comprising a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks,

processing the trace information to display in a graphic user interface a representation of the threads and the tasks executing in the processor;

receiving a user selection of one or more of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of each selected task including a representation of a duration of each selected task, the duration of each selected task being determined according to begin and end times in the trace information;

the instrumented code further comprising code which creates at least one virtual task and identifies one of more tasks as being associated tasks, each associated task comprising one of the tasks for which trace information is generated, the virtual task having a duration determined by begin and end times of the one or more of the associated tasks; and

displaying a visualization of the virtual task including a representation of the duration of the virtual task.

Claim 10

A computer-implemented method of providing a performance analysis tool for analysing the performance of a multi-core processor when executing code, the method comprising steps of;

executing in the multi-core processor instrumented code comprising a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks,

processing the trace information to display in a graphic user interface a representation of the threads and the tasks executing in the processor;

receiving a user selection of one or more of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of each selected task including a representation of a duration of each selected task, the duration of each selected task being determined according to begin and end times in the trace information;

the instrumented code further comprising code which creates at least one virtual task and identifies one of more tasks as being associated tasks, each associated task comprising one of the tasks for which trace information is generated, the virtual task having a duration determined by begin and end times of the one or more of the associated tasks; and

displaying a visualization of the virtual task including a representation of the duration of the virtual task.

Claim 11

A computer system adapted to provide a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the system adapted to;

execute in the multi-core processor instrumented code comprising a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks,

process the trace information to display in a graphic user interface a representation of the threads and the tasks executing in the processor;

receive a user selection of one or more of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of each selected task including a representation of a duration of each selected task, the duration of each selected task being determined according to begin and end times in the trace information;

the instrumented code further defining at least one virtual task having associated therewith one or more associated tasks, each associated task comprising one of the tasks for which trace information is generated, the virtual task having a duration determined by begin and end times of the one or more of the associated tasks; and

the system further adapted to display a visualization of the virtual task including a representation of the duration of the virtual task.

Claim 13

A computer system adapted to provide a performance analysis tool for analysing the performance of a multi-core processor when executing graphics processing code, the system adapted to;

execute in the multi-core processor instrumented code comprising a version of the code which includes a number of application program interfaces, API's, adapted to provide trace information indentifying threads running in the processor, tasks running within respective threads and functional relationships between the tasks,

process the trace information to display in a graphic user interface a representation of the threads and the tasks executing in the processor;

receive a user selection of one or more of the tasks and, in response to receiving said user selection, processing the trace information to provide in the graphic user interface a visualization of each selected task including a representation of a duration of each selected task, the duration of each selected task being determined according to begin and end times in the trace information;

the instrumented code further defining at least one virtual task having associated therewith one or more associated tasks, each associated task comprising one of the tasks for which trace information is generated, the virtual task having a duration determined by begin and end times of the one or more of the associated tasks; and

the system further adapted to display a visualization of the virtual task including a representation of the duration of the virtual task.