(54) Title of the Invention: Monitoring the performance of a computer
Abstract Title: Monitoring the performance of a computer

(57) A method for monitoring the performance of a computer 100 when in use by a user 102 comprises the steps of: detecting a wait state 106 in which the user is waiting for the computer; and collecting performance data (such as resource utilisation data) relating to the duration of the wait state 108; in which detecting a wait state includes: detecting a change in a cursor (204 in Figure 2) associated with a pointing device being used by the user and determining if the cursor is a wait cursor (such as an hourglass); or detecting a logon event (Figure 3); and where the performance data is used to determine the performance of the computer, or the measure and/or cause of the delay experienced by the user. The performance data may be analysed to identify a lack of resources and reduce the duration of the wait state by increasing availability of resources.

Figure 1

![Diagram of monitoring performance](image-url)
Figure 1

100
104 User Session

106 Detect Wait State?

Y
108 Measure Duration of Wait State

N

110 Store Resultant Wait Time and Resource Utilisation Counters

112 Improve Performance
Figure 2

202 Mouse

102 User

104 User Session

204 Mouse Cursor Change Event

206 Is it a Wait Cursor?

208 Start Wait Timer and Resource Utilisation Counters

212 Wait For Next Mouse Cursor Change Event

214 Is it a Wait Cursor?

216 Stop Wait Timer and Resource Utilisation Counters

110 Store Resultant Wait Time and Resource Utilisation Counters
Figure 3

102 User

302 User Logon and Monitoring Agent Start

304 Record Monitoring Agent Start Time

306 Is there a Logon Audit Event?

N Wait For Next User Logon

Y 308 Calculate Logon Delay by subtracting Logon Audit Event timestamp from Monitoring Agent Start Timestamp

110 Store Resultant Logon Delay Time
Figure 4

110
Stored Resultant
Wait Times and
Resource Utilisations Counters

502
Analyse Wait Times and
Resource Utilisations

504
Report:
Total Wait Times,
Potential Resource
Bottlenecks, etc.
The following terms are registered trade marks and should be read as such wherever they occur in this document:

Mac OS
Monitoring the Performance of a Computer

The present invention is concerned with monitoring the performance of a computer by detecting a wait state in which the computer is not available for user input in order to determine how much time a computer is available to a user during a user session.

GB2370140A discloses a method for determining system resource parameters to improve response time for specific users. US6046816A discloses measuring the total time between a user requesting a print of a document and the completion of the print job. JP2-105236A discloses a method for counting the response time in a time-sharing system.

Computer response time has been defined as the elapsed time between the end of an inquiry or demand on a computer system and the beginning of a response; for example, the length of the time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal. In other words, the computer is in a wait state of varying duration. For simple operations, the response time can appear to be instantaneous to the user.

For other operations where the response time is longer, an indication is given to the user that the computer or application is in a wait state and not available for user input, usually because it is busy performing an operation, or the operating system is active working on another task. This can be done, for example, by the input cursor ceasing to flash. On computer operating systems using a graphical user interface, for example those running on personal computers, laptops, smart phones and the like, a common way of letting the user know the computer is in such a wait state and cannot accept user input is to change the mouse cursor. The wait cursor (an hourglass in Windows® before Vista® and many other systems, spinning ring in Windows Vista, watch in classic Mac OS, or spinning ball in Mac OS X) is displayed when the mouse cursor is in the corresponding window. This wait state can be detected.
Another wait time is associated with logging on to the computer or system, in which the operating system is setting up parameters specific to the user, or returning from a standby or hibernated state. This can generate a wait state cursor, or it can be a display with a moving progress bar on it. The operating system is unavailable for user input during this phase of its operation.

A further wait state occurs as a result of poor network performance, in which the operating system or a program running on the operating system is waiting for information to be delivered or received by an external network, as for example when accessing email or a web site.

These wait states conspire to reduce the productivity of the user, increase a users stress level, and can generally lead to a poor user experience. Obtaining an objective measure of the delay caused to a user during these wait states by a lack of computer resources (of whatever kind: e.g. memory, processor, connectivity, device-related, network speed, operating system state, application system state) while interacting with a computer has long been a desired goal but, due to the difficulty of pinpointing the source of the delay precisely, not, so far, successfully solved.

According to a first aspect, the present invention provides a method for monitoring the performance of a computer when in use by a user, comprising the steps of: detecting a wait state in which the user is waiting for the computer; and collecting performance data relating to a duration of the wait state. This provides an assessment of the period of time the user is waiting for the computer to complete an action, process or command. The step of detecting a wait state includes:

detecting a change in a cursor associated with a pointing device being used by the user and determining if the cursor is a wait cursor, wherein the duration of the wait state is a period of time during which the cursor is a wait cursor; or
detecting a logon event, wherein the duration of the wait state is the period of time
during in which the computer is not available for user input.

The method assesses the performance provided to the user of the computer from the
performance data.

Preferably, the step of collecting performance data relating to the wait state includes
measuring a duration of the wait state. This provides an assessment of the period of time
the user is waiting for the computer to complete an action, process or command.

Preferably, the step of detecting a wait state includes detecting a change in a cursor
associated with a pointing device being used by the user and determining if the cursor is a
wait cursor, wherein the duration of the wait state is a period of time during which the cursor
is a wait cursor. The wait cursor is a clear indicator of the kind of wait state a user
experiences whilst using the computer.

Preferably, the pointing device is selected from the group consisting of: mouse, touch
pad and touch screen. Advantageously, the wait state can be determined from the user’s
use of a range of input devices. Input devices may include eye movement, motion detection,
voice control or any similar means allowing a user to interact with information shown on a
screen.

Preferably, the duration of the wait state is elapsed time between an end of a user
input and a beginning of a computer response. This gives a measure of the time during
which the user is unable to use the computer. Typically the duration of the wait state relates
to a single user session.

Preferably, the duration of the wait state is a sum of all the durations of a wait state
during a user session. This gives an indication of the total amount of time the user has not
been able to use the computer.
Preferably, the step of collecting performance data additionally comprises the step of: measuring a duration the computer is in use by the user. This provides information on how much time the user had been able to use the computer.

Preferably, the duration of time the computer is in use by the user is the duration of a user session. This provides information on the total amount of time the user had been able to use the computer.

Preferably, the performance of the computer is measured by the duration of the wait state and its ratio to the duration the computer is in use. This gives an indication of the proportion of the session the computer has been unavailable for use.

Preferably, the step of collecting performance data additionally comprises the step of measuring a duration the computer is available for use by the user. This provides information on the availability of the computer for use by the user.

Preferably, the duration the computer is available for use by the user is a sum of all the durations the computer is available for use during a user session. This provides information on the total amount of time the computer was available for use by the user.

Preferably, the performance of the computer is measured by a ratio between the duration of the wait state and its ratio with the duration the computer is available. This gives an indication of the proportion of time the computer has been unavailable for use.

Preferably, the step of detecting a wait state includes detecting a logon event, wherein the duration of the wait state is the period of time during in which the computer is not available for user input. This provides an assessment of the period of time the user is waiting for the computer to become available for use.

Preferably, the step of collecting performance data relating to the wait state includes data relating to resource utilisation. Preferably, data relating to resource utilisation includes one or more of: CPU utilisation, memory utilisation, disk queue utilisation, and disk free
space. This is especially important where the performance of the computer is slowed because of a lack of memory or disk space or other resource.

Preferably, the step of collecting performance data relating to the wait state includes data relating to the network application. Preferably, data relating to resource utilisation includes one or more of: network upload speed and network download speed. This is especially important where the performance of the computer is slowed because of a poor, badly configured or busy network connection.

Preferably, the step of collecting performance data relating to the wait state includes the additional step of: analysing the performance data; and identifying a lack of resources. A duration of the wait state may be reduced by increasing availability of resources.

Preferably, the performance data is analysed to provide a report, wherein the performance of the computer is improved. This is useful if, for example, the report shows that a key resource is limited – improvements to the computer or its connections can lead to an immediate improvement in performance.

Preferably, the performance data is uploaded for analysis. This means that the analysis does not consume user resources.

Preferably, the computer is one of a server, a desktop computer, a laptop computer, a mobile computer, a smart phone. The method may be applied to any computing device that the user may choose to use.

According to a further aspect, the present invention provides a method for measuring a delay experienced by a user interacting with a computer comprising the steps of: detecting a state in which the user is waiting for the computer and collecting performance data relating to a duration of the wait state. The step of detecting a wait state includes:

detecting a change in a cursor associated with a pointing device being used by the user and determining if the cursor is a wait cursor, wherein the duration of the wait state is a period of time during which the cursor is a wait cursor, or
detecting a logon event, wherein the duration of the wait state is the period of time
during in which the computer is not available for user input.

The method measures the delay experienced by the user from the performance data.

According to a further aspect, the present invention provides a method for determining
a cause of a delay experienced by a user interacting with a computer comprising the steps
of: detecting a wait state in which the user is waiting for the computer and collecting
performance data relating to a duration of the wait state. The step of detecting a wait state
includes:

detecting a change in a cursor associated with a pointing device being used by the
user and determining if the cursor is a wait cursor, wherein the duration of the wait state is a
period of time during which the cursor is a wait cursor; or

detecting a logon event, wherein the duration of the wait state is the period of time
during in which the computer is not available for user input.

The method determines a cause of the delay experienced by the user from the
performance data.

Preferred embodiments of the present invention will now be described, by way of
example only, with reference to the accompanying drawings, in which:

Figure 1 shows a schematic for a method for monitoring the performance of a
computer;

Figure 2 shows a schematic for a method for monitoring the performance of a
computer, wherein the wait state is associated with a wait state cursor;

Figure 3 shows a schematic for a method for monitoring the performance of a
computer, wherein the wait state is associated with a logon event;

Figure 4 shows a schematic showing how computer performance can be monitored
and improved.
Figure 1 shows a method for monitoring the performance of a computer 100 when in use by user 102. During user session 104, at step 106, a wait state in which the user is waiting for the computer is detected and its duration can be measured. This can be because a process running on the computer means that the computer is not available for user input (e.g. when a wait cursor is displayed, or during login). In step 108 performance data relating to the wait state are collected. This data can include the duration of a wait state and resource utilisation data, amongst others. In step 110, the data relating to the wait state are stored. The performance of the computer is determined from the performance data. In step 112, the performance of the computer can be improved by analysing the stored data. The method measures a duration of the delay, and/or determines a cause of the delay experienced by the user from the performance data. The cause of the delay is typically due to a lack of resources, and analysing the performance data and identifying a lack of resources means that a duration of the wait state may be reduced by increasing availability of resources.

Figure 2 shows a method for monitoring the performance of a computer 100 when in use by a user 102 interacting with computer 100 by means of a pointing device 202, which may be a mouse as shown, or it may be a touch pad, touch screen or other similar input device. The input device is any device which allows a user to interact with information shown, and it may also include eye movement, limb motion, neural activity or voice activation. The wait state is detected in steps 204 and 206. In step 204, a change in pointing device cursor is detected, and in step 206 the change is assessed: is it a change to a wait cursor? The wait state cursor (such as an hourglass in Windows® before Vista® and many other systems, spinning ring in Windows Vista, watch in classic Mac OS, or spinning ball in Mac OS X) is displayed when the mouse cursor is in the corresponding window. If the cursor is a wait cursor, performance data relating to the wait state are collected. These data include a duration of the wait state and resource utilisation, and in steps 208 to 216, these are measured. In step 208, a wait timer and various resource utilisation counters are actuated.
Alternatively, and usefully when the wait state of the operating system is so prolonged that
the operating system seems frozen, wait timer logs the time stamp at which the wait state
started. In step 212 the cursor is monitored, and if, in step 214 the cursor remains a wait
cursor, step 212 is repeated. If in step 214 the cursor is no longer a wait cursor, then in step
216 the wait timer and resource utilisation counters are halted. The duration of the wait state
is thus the period of time during which the cursor is a wait cursor. In step 110, performance
data relating to the wait state is stored. The method measures a duration of the delay, and /
or determines a cause of the delay experienced by the user from the performance data. The
cause of the delay is typically due to a lack of resources, and analysing the performance
data and identifying a lack of resources means that a duration of the wait state may be
reduced by increasing availability of resources.

By way of example, the method of the present invention can be achieved by an agent
running on the computer monitoring the cursor, be it a mouse, touch pad, touch screen or
other pointing device cursor, and detects when the cursor becomes a wait state cursor (an
hourglass or spinning circle that temporarily replaces the arrow). When the wait state has
been detected, the agent then:

Starts the wait state timer to measure the length of the wait state (point ‘W1’)
Initialises resource utilisation counters, i.e. sets them to zero, for example for one or
more of the following:

- CPU 0 Utilisation;
- Memory Utilisation;
- Disk 0 Queue Utilisation; and
- Disk 0 Free Space.

The agent continues to monitor the cursor, and when it detects that the cursor is no
longer a wait state cursor, the agent then:

Stops the wait state timer (point ‘W2’) and calculates the wait time ‘W2’ − ‘W1’ = ‘W3’
in milliseconds
Retrieves the resource utilisation counters for the resource allocation counters, for example, one or more of the following:

- CPU 0 Utilisation ‘RU1’ as a percentage;
- Memory Utilisation ‘RU2’ in Mbytes;
- Disk 0 Queue Utilisation ‘RU3’ in Integer Units (e.g. 0 to 10); and
- Disk 0 Free Space ‘RU4’ in Mbytes.

Writes the wait state data (including the Wait Time ‘W3’.) to the log file as a “WaitState” entry.

The approach isn’t limited to the monitoring of a cursor on a computer having a graphical user interface, but also includes any input form in which a wait state occurs when the computer is not available for input of data by the user.

Optionally the user can measure a duration of time the computer is in use by a user. This can be the total duration of a user session, or it can be the duration of time the computer is available for use by the user, which is the difference between total duration of a user session and wait time ‘W3’.

The data collected may be used to provide a measure of computer performance as experienced by a user from, for example, a ratio of a duration of wait time ‘W3’ to total duration of a user session, or a ratio of a duration of wait time ‘W3’ to the duration of time the computer is available for use by the user.

Figure 3 shows a method for monitoring the performance of a computer 100 when in use by a user 102, where the user is logging on to computer 100 and waiting for the computer to accept input. In step 302, a logon event is detected and the monitoring of the login process, and collection of performance data relating to the login process begins. These data include a duration of the login wait state. In step 304, the time stamp immediately following the logon event is captured. If there has been a logon audit event 306, the start time of the logon audit event is captured. In step 308 the duration of the wait state associated with logon is determined by the difference between time stamp immediately
following the logon event and start time of the logon audit event. The duration of the wait state is thus the period of time during in which the computer is not available for user input. The resultant performance data, i.e. the logon delay time, is logged 110. The method measures a duration of the delay, and / or determines a cause of the delay experienced by the user from the performance data. The cause of the delay is typically due to a lack of resources, and analysing the performance data and identifying a lack of resources means that a duration of the wait state may be reduced by increasing availability of resources.

By way of example, the method of the present invention can be achieved by an agent running on the computer and which is actuated following logon. When user 102 logs onto the target PC, the agent is automatically started and reads the Windows Event Log for the latest logon event to determine when the Windows OS time stamped the start of the logon sequence. The agent subtracts the start logon timestamp from the current time to determine the Logon Delay and writes the result to the log file as a “LogonDelay” entry.

In the above, the performance of the computer is measured by the duration of the wait state. The method measures a duration of the delay, and / or determines a cause of the delay experienced by the user from the performance data. The cause of the delay is typically due to a lack of resources, and analysing the performance data and identifying a lack of resources means that a duration of the wait state may be reduced by increasing availability of resources.

Figure 4 shows a schematic showing how computer performance can be monitored and improved. Data stored at step 110 can be uploaded to a server. In order for the uploading not to lead to a reduction in the performance of the computer, the data is uploaded when the computer is idle, for example when there has been no mouse or keyboard activity for at least 60 seconds. In step 502, the server analyses the data to improve the performance of the computer, and in step 504, provides a report. The method measures a duration of the delay, and / or determines a cause of the delay experienced by the user from the performance data. The cause of the delay is typically due to a lack of
resources, and analysing the performance data and identifying a lack of resources means that a duration of the wait state may be reduced by increasing availability of resources.

For example WaitEvent data are averaged with other WaitEvent entries on the same day, for example, the sum of all ‘RU1’ data divided by the number ‘RU1’ of entries. This is repeated for each of the ‘RU2’ – ‘RU4’ Resource Utilisation data to the form the Summarised WaitEvent data.

The summarised WaitEvent data are displayed, on demand by the user, within a User determinable date range (for example, in 1 day units) on to a GUI driven by the server, typically a web page. The summary includes one or more of the following:

Total Wait Time (Total Lost Time);
A presentation of the cost of the Total Wait Time is determined by converting the Total Wait Time to hours and then multiplying by a cost per hour figure, as defined and stored in the Users local parameter file on the User’s PC; or centrally on a server
An Application Wait Time Rating is determined by comparison with predetermined thresholds e.g. Poor, OK, Good, and presented to the GUI. The thresholds are stored centrally and are considered global to the whole system to provide consistency and facilitate comparisons between disparate PC’s and installations.
A raw data export facility can also be provided to allow organisations to perform their own customised analysis.

Similarly the LogonDelay event data are aggregated with other LogonDelay events on the same day

The aggregated LogonWait data are displayed, on demand by the user, on a GUI driven by the server, typically a web page, including:

A Logon Delay Rating is determined by comparison with predetermined thresholds e.g. Poor, OK, Good
Similarly the NetSpeed data are aggregated with other NetSpeed entries on the same day.

The aggregated NetSpeed data are displayed, on demand by the user, on a GUI driven by the server, typically a web page, including:

Network Speed Ratings (upload and download) which is determined by comparison with predetermined thresholds e.g. Poor, OK, Good.

The approach can be applied to detecting and/or measuring any existing wait state parameter provided directly or indirectly by the operating system kernel of a computer (i.e. any device capable of being so monitored), or, if absent, to supply one by modifying or supplementing the kernel of the operating system and to record, by means of whatever accurate timing algorithm or timing resource available on the device, the changes in state in a local data file held on the computer’s storage medium (or another connected computer). In an alternative embodiment, if there were no reliable operating system wait state flag, the present invention would simulate a hardware layer input “beneath” the operating system to allow a wait state flag to be set and passed through whatever operating system was then installed upon the computer.

The method of the invention monitors events, such as commencing a login to a connected resource and achieving login, or accessing a resource on a computer connected by network to the computer on which the monitoring is taking place, and information about the events is captured to the data file.

Accurate timing means are utilised (such that the synchronicity of events is preserved as between the events monitored) to capture resource utilisation of the user’s computer by means of known activity measures, (for example – disk data transfer rate relative to maximum data transfer rate for the disk; memory data transfer rate/ number of read / write requests, and so forth), such that measures of the relative intensity of resource utilisation is synchronously captured in the data file.
Retrieval of the data for either a) near real time monitoring or b) archival retrieval of the records in the data file enables determinations of which resource constraints generate which wait state. This process being partly or wholly conducted by analysis of the local datafile and / or partly or wholly from global analysis and comparison of patterns revealed from analysis of the community database of all possible performance data collected from monitored devices of similar type and configuration.

Collation of such data from all possible monitored devices enables further analysis, to provide supplementing analysis scoring/recommendations for detected patterns and related causes of wait states on monitored devices of similar type and configuration.
Claims

1. A method for monitoring the performance of a computer when in use by a user, comprising the steps of:
   detecting a wait state in which the user is waiting for the computer; and
   collecting performance data relating to a duration of the wait state;
   in which the step of detecting a wait state includes:
   detecting a change in a cursor associated with a pointing device being used by the user and determining if the cursor is a wait cursor, wherein the duration of the wait state is a period of time during which the cursor is a wait cursor; or
   detecting a logon event, wherein the duration of the wait state is the period of time during in which the computer is not available for user input;
   wherein, in use, the performance of the computer is determined from the performance data.

2. A method according to claim 1, in which the pointing device is selected from the group consisting of: mouse, touch pad and touch screen.

3. A method according to claim 2, in which the duration of the wait state is elapsed time between an end of a user input and a beginning of a computer response.

4. A method according to claim 2 or claim 3, in which the duration of the wait state is a sum of all the durations of a wait state during a user session.

5. A method according to any of claims 2 to 4, additionally comprising the step of:
   measuring a duration the computer is in use by the user.
6. A method according to claim 5, in which the duration of time the computer is in use by
the user is the duration of a user session.

7. A method according to claim 5 or claim 6, in which the performance of the computer is
measured by the duration of the wait state and its ratio to the duration the computer is
in use.

8. A method according to claim 2, additionally comprising the step of measuring a
duration the computer is available for use by the user.

9. A method according to claim 8, in which the duration the computer is available for use
by the user is a sum of all the durations the computer is available for use during a user
session.

10. A method according to claim 8 or claim 9, in which the performance of the computer is
measured by a ratio between the duration of the wait state and its ratio with the
duration the computer is available.

11. A method according to claim 2, in which the step of detecting a wait state includes
detecting a logon event, and in which the step of collecting performance data relating
to the wait state includes data relating to resource utilisation.

12. A method according to claim 11, in which data relating to resource utilisation includes
one or more of: CPU utilisation, memory utilisation, disk queue utilisation; and disk free
space.

13. A method according to claim 12, in which data relating to resource utilisation includes
one or more of: network upload speed and network download speed.
14. A method according to any of claims 11 to 13, comprising the additional step of:
analysing the performance data; and
identifying a lack of resources;
whereby a duration of the wait state may be reduced by increasing availability of resources.

15. A method according to any preceding claim, in which the performance data is
analysed to provide a report, wherein the performance of the computer is improved.

16. A method according to claim 15, including the additional step of uploading the
performance data for analysis.

17. A method according to any preceding claim, in which the computer is one of a server,
a desktop computer, a laptop computer, a mobile computer, a smart phone.

18. A method for measuring a delay experienced by a user interacting with a computer
comprising the steps of:
detecting a wait state in which the user is waiting for the computer;
collecting performance data relating to a duration of the wait state;
in which the step of detecting a wait state includes:
detecting a change in a cursor associated with a pointing device being used by the
user and determining if the cursor is a wait cursor, wherein the duration of the wait
state is a period of time during which the cursor is a wait cursor; or
detecting a logon event, wherein the duration of the wait state is the period of time
during in which the computer is not available for user input;
wherein, in use, the measure and cause of the delay experienced by the user is
determined from the performance data.
19. A method for determining a cause of a delay experienced by a user interacting with a computer comprising the steps of:

detecting a wait state in which the user is waiting for the computer;
collecting performance data relating to a duration of the wait state;
in which the step of detecting a wait state includes:
detecting a change in a cursor associated with a pointing device being used by the user and determining if the cursor is a wait cursor, wherein the duration of the wait state is a period of time during which the cursor is a wait cursor; or
detecting a logon event, wherein the duration of the wait state is the period of time during in which the computer is not available for user input;
wherein, in use, the cause of the delay experienced by the user is determined from the performance data.

20. A method substantially as described herein described with reference to the drawings.
Application No: GB1313239.4  
Examiner: Dr Christopher Batty  
Claims searched: 1 to 10, and 15 to 20  
Date of search: 20 February 2014

Patents Act 1977: Search Report under Section 17

Documents considered to be relevant:

<table>
<thead>
<tr>
<th>Category</th>
<th>Relevant to claims</th>
<th>Identity of document and passage or figure of particular relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1 to 10, and 15 to 20</td>
<td>US2012/0144246 A1 (DREYFOOS et al.) See Figure 3 and paragraphs [0044] to [0052] for example.</td>
</tr>
<tr>
<td>A</td>
<td>-</td>
<td>US5630148 A (NORRIS) See Figure 2, and lines 5 to 27 of column 5, for example.</td>
</tr>
<tr>
<td>A</td>
<td>-</td>
<td>WO98/44418 A1 (LANDMARK SYSTEMS CORP.) See lines 16 to 23 of page 1, and line 19 of page 6 to line 2 of page 7, for example.</td>
</tr>
</tbody>
</table>

Categories:

<table>
<thead>
<tr>
<th>X</th>
<th>Document indicating lack of novelty or inventive step</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Document indicating lack of inventive step if combined with one or more other documents of same category</td>
</tr>
<tr>
<td>&amp;</td>
<td>Member of the same patent family</td>
</tr>
</tbody>
</table>

| A         | Document indicating technological background and/or state of the art. |
| P         | Document published on or after the declared priority date but before the filing date of this invention. |
| E         | Patent document published on or after, but with priority date earlier than, the filing date of this application. |

Field of Search:

Search of GB, EP, WO & US patent documents classified in the following areas of the UKC:

Worldwide search of patent documents classified in the following areas of the IPC

G06F

The following online and other databases have been used in the preparation of this search report:

EPODOC, WPI; TXTE
### International Classification:

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Subgroup</th>
<th>Valid From</th>
</tr>
</thead>
<tbody>
<tr>
<td>G06F</td>
<td>0011/30</td>
<td>01/01/2006</td>
</tr>
<tr>
<td>G06F</td>
<td>0011/34</td>
<td>01/01/2006</td>
</tr>
</tbody>
</table>