

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

①1 N° de publication : **2 977 338**
(à n'utiliser que pour les
commandes de reproduction)

②1 N° d'enregistrement national : **11 55752**

⑤1 Int Cl⁸ : **G 06 F 9/50 (2013.01), G 06 Q 10/06**

⑫ **DEMANDE DE BREVET D'INVENTION**

A1

②2 **Date de dépôt** : 28.06.11.

③0 **Priorité** :

④3 **Date de mise à la disposition du public de la demande** : 04.01.13 Bulletin 13/01.

⑤6 **Liste des documents cités dans le rapport de recherche préliminaire** : *Se reporter à la fin du présent fascicule*

⑥0 **Références à d'autres documents nationaux apparentés** :

⑦1 **Demandeur(s)** : BULL SAS Société par actions simplifiée — FR.

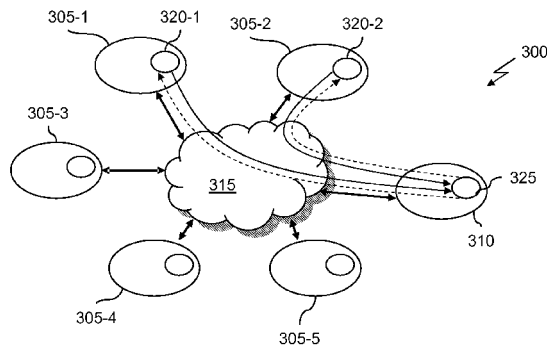
⑦2 **Inventeur(s)** : FICET ALINE.

⑦3 **Titulaire(s)** : BULL SAS Société par actions simplifiée.

⑦4 **Mandataire(s)** : SANTARELLI.

⑤4 **PROCEDE ET PROGRAMME D'ORDINATEUR POUR IDENTIFIER DYNAMIQUEMENT DES COMPOSANTS D'UN CLUSTER ET AUTOMATISER DES OPERATIONS DE GESTION OPTIMISEE DU CLUSTER.**

⑤7 L'invention a notamment pour objet l'identification dynamique de composants d'éléments (305, 310) d'un cluster (300) afin d'automatiser des opérations de gestion optimisée du cluster. Après avoir été établi dans un premier élément, un inventaire local comprenant des caractéristiques de composants matériels et/ou logiciels du premier élément est transmis à un second élément, distinct du premier. Un inventaire partiel ou global comprenant des caractéristiques de composants matériels et/ou logiciels du premier élément est ensuite établi dans le second élément et une opération de gestion du cluster est lancée selon ledit inventaire partiel ou global.



FR 2 977 338 - A1



5 La présente invention concerne la gestion de composants dans des clusters et plus particulièrement un procédé et un programme d'ordinateur pour identifier dynamiquement des composants d'un cluster et automatiser des opérations de gestion optimisée du cluster.

10 Le calcul haute performance, aussi appelé HPC (sigle de *High Performance Computing* en terminologie anglo-saxonne) se développe pour la recherche universitaire comme pour l'industrie, notamment dans des domaines techniques tels que l'automobile, l'aéronautique, l'énergie, la climatologie et les sciences de la vie. La modélisation et la simulation permettent en particulier de réduire les coûts de développement, d'accélérer la mise sur le marché de
15 produits innovants, plus fiables et moins consommateurs d'énergie. Pour les chercheurs, le calcul haute performance est devenu un moyen d'investigation indispensable.

 Ces calculs sont généralement mis en œuvre sur des systèmes de traitement de données appelés clusters. Un cluster comprend typiquement un
20 ensemble de nœuds interconnectés. Certains nœuds sont utilisés pour effectuer des tâches de calcul (nœuds de calcul), d'autres pour stocker des données (nœuds de stockage) et un ou plusieurs autres gèrent le cluster (nœuds d'administration). Chaque nœud est par exemple un serveur mettant en œuvre un système d'exploitation tel que Linux (Linux est une marque). La
25 connexion entre les nœuds est, par exemple, réalisée à l'aide de liens de communication Ethernet ou Infiniband (Ethernet et Infiniband sont des marques).

 La figure 1 illustre schématiquement un exemple d'une topologie 100 d'un cluster, de type *fat-tree*. Ce dernier comprend un ensemble de nœuds
30 génériquement référencés 105. Les nœuds appartenant à l'ensemble 110 sont ici des nœuds de calcul tandis que les nœuds de l'ensemble 115 sont des nœuds de service (nœuds de stockage et nœuds d'administration). Les nœuds

de calcul peuvent être regroupés en sous-ensembles 120 appelés îlots de calcul, l'ensemble 115 étant appelé îlot de service.

Les nœuds sont reliés les uns aux autres par des commutateurs (appelés *switch* en terminologie anglo-saxonne), par exemple de façon hiérarchique. Dans l'exemple illustré sur la figure 1, les nœuds sont connectés à des commutateurs 125 de premier niveau qui sont eux-mêmes reliés à des commutateurs 130 de deuxième niveau qui sont à leur tour reliés à des commutateurs 135 de troisième niveau.

Comme illustré sur la figure 2, chaque nœud comprend généralement un ou plusieurs microprocesseurs, des mémoires locales ainsi qu'une interface de communication. Plus précisément, le nœud 200 comporte ici un bus de communication 202 auquel sont reliés :

- des unités centrales de traitement ou microprocesseurs 204 (ou CPU, sigle de *Central Processing Unit* en terminologie anglo-saxonne) ;
- 15 - des composants de mémoire vive 206 (RAM, acronyme de *Random Access Memory* en terminologie anglo-saxonne) comportant des registres adaptés à enregistrer des variables et paramètres créés et modifiés au cours de l'exécution de programmes (comme illustré, chaque composant de mémoire vive est associé à un microprocesseur) ; et,
- 20 - des interfaces de communication 208 adaptées à transmettre et à recevoir des données.

Le nœud 200 dispose en outre ici de moyens de stockage interne 210, tels que des disques durs, pouvant notamment comporter le code exécutable de programmes.

Le bus de communication permet la communication et l'interopérabilité entre les différents éléments inclus dans le nœud 200 ou reliés à lui. Les microprocesseurs 204 commandent et dirigent l'exécution des instructions ou portions de code logiciel du ou des programmes. Lors de la mise sous tension, le ou les programmes qui sont stockés dans une mémoire non volatile, par exemple un disque dur, sont transférés dans la mémoire vive 206.

Le nombre de composants mis en œuvre dans un cluster, notamment d'équipements tels que des nœuds et des commutateurs, varie d'un

cluster à un autre. Il peut être relativement faible, par exemple quelques dizaines de nœuds, ou très important, par exemple quelques milliers voire dizaines de milliers de nœuds. En outre, les composants mis en œuvre, d'un même type, peuvent être différents. Ainsi, par exemple, certains nœuds utilisés
5 sont des nœuds comprenant une seule unité de calcul alors que d'autres nœuds en comprennent plusieurs. De même, la taille des mémoires et les versions logicielles ne sont pas nécessairement les mêmes pour chaque nœud.

Pour gérer un cluster efficacement et, notamment, s'assurer de son bon fonctionnement, il est nécessaire de connaître tous les composants
10 matériels et logiciels qu'il comprend. La liste des composants d'un cluster est ainsi souvent définie dans une base de données. Alors qu'elle peut être déterminée de façon théorique à partir des données de conception du cluster, une liste des composants réellement mis en œuvre est généralement établie pour s'assurer de la conformité du cluster. Un tel inventaire est typiquement
15 réalisé par un opérateur qui va interroger, de façon manuelle ou automatique, chaque composant du cluster pour en déduire la composition du cluster.

Cependant, la complexité croissante des composants des clusters et l'évolution de la taille de ces derniers exigent des moyens d'administration et de gestion de plus en plus performants.

20 L'invention permet de résoudre au moins un des problèmes exposés précédemment.

L'invention a ainsi pour objet un procédé pour identifier dynamiquement des composants matériels et/ou logiciels d'au moins un élément d'un cluster comprenant une pluralité d'éléments et automatiser des
25 opérations de gestion optimisée dudit cluster, ledit au moins un élément appartenant à ladite pluralité d'éléments et chaque élément de ladite pluralité d'éléments ayant des capacités de traitement de données, ce procédé comprenant les étapes suivantes,

- dans ledit au moins un élément, appelé au moins un premier élément,
30
 - o établissement d'au moins un inventaire local, ledit au moins un inventaire local comprenant au moins une caractéristique d'au

- moins un composant matériel et/ou logiciel dudit au moins un premier élément ; et,
- transmission dudit au moins un inventaire local à un élément de ladite pluralité d'éléments dudit cluster, distinct dudit au moins un premier élément, appelé second élément ;
- 5
- dans ledit second élément,
 - réception dudit au moins un inventaire ;
 - établissement d'un inventaire partiel ou global comprenant au moins ladite au moins une caractéristique dudit au moins un
- 10
- lancement d'une opération de gestion dudit cluster selon ledit inventaire partiel ou global.

Le procédé selon l'invention permet ainsi d'améliorer l'utilisation d'un cluster à partir de la connaissance des composants de ses éléments. L'inventaire de composants étant réalisé à distance, il peut être réalisé automatiquement dans de nombreuses configurations et situations.

15

Selon un mode de réalisation particulier, le procédé comprend en outre, dans ledit au moins un premier élément, une étape de détection d'un événement, lesdites étapes d'établissement et de transmission dudit au moins un inventaire local étant effectuées en réponse à ladite détection. Il est ainsi possible de mettre à jour un inventaire selon des événements prédéterminés.

20

Toujours selon un mode de réalisation particulier, le procédé comprend en outre, dans ledit second élément, une étape d'émission d'une requête d'inventaire, ladite étape de détection d'un événement comprenant une détection d'une réception de ladite requête. Il est ainsi possible de mettre à jour un inventaire sur requête d'un nœud d'administration. Une telle requête peut être produite automatiquement ou manuellement.

25

Alternativement, ledit événement détecté est généré dans ledit au moins un premier élément. Il est ainsi possible de mettre à jour un inventaire selon un événement local, par exemple un redémarrage ou une mise à jour.

30

De façon avantageuse, le procédé comprend en outre une étape d'identification dudit au moins un composant matériel et/ou logiciel dudit au moins un premier élément, ledit au moins un inventaire local comprenant un identifiant dudit au moins un composant matériel et/ou logiciel dudit au moins un premier élément.

Selon des modes de réalisation particuliers, ladite opération de gestion dudit cluster comprend une optimisation de répartition de charge entre plusieurs composants matériels d'éléments de ladite pluralité d'éléments, une vérification de conformité d'au moins un composant matériel et/ou logiciel d'au moins un élément de ladite pluralité d'éléments et/ou une vérification de mise à jour d'au moins un composant logiciel d'au moins un élément de ladite pluralité d'éléments.

Toujours selon un mode de réalisation particulier, lesdites étapes de réception dudit au moins un inventaire et d'établissement d'un inventaire partiel ou global sont répétées pour tous les éléments de ladite pluralité d'éléments d'un type prédéterminé de telle sorte que l'inventaire partiel ou global soit relatif à une pluralité d'éléments.

Toujours selon un mode de réalisation particulier, au moins un inventaire local d'un élément dudit cluster n'ayant pas de capacité de traitement de données est établi par ledit second élément, au moins une donnée dudit inventaire local dudit élément n'ayant pas de capacité de traitement de données étant utilisée pour l'établissement dudit inventaire partiel ou global. L'inventaire partiel ou global comprend ainsi des caractéristiques d'éléments n'ayant pas de capacité de traitement de données, tels que des commutateurs. Les éléments ayant des capacités de traitement de données sont par exemple des nœuds.

L'invention a également pour objet un programme d'ordinateur comprenant des instructions adaptées à la mise en œuvre de chacune des étapes du procédé décrit précédemment lorsque ledit programme est exécuté sur un ordinateur. Les avantages procurés par ce programme d'ordinateur sont similaires à ceux évoqués précédemment.

D'autres avantages, buts et caractéristiques de la présente invention ressortent de la description détaillée qui suit, faite à titre d'exemple non limitatif, au regard des dessins annexés dans lesquels :

- la figure 1 illustre un exemple de topologie d'un cluster ;
- 5 - la figure 2 illustre un exemple d'architecture d'un nœud d'un cluster ;
- la figure 3 illustre schématiquement un exemple de mise en œuvre du procédé selon l'invention dans un cluster ;
- la figure 4 illustre schématiquement certaines étapes mises en œuvre dans un élément d'un cluster, par exemple un nœud, pour effectuer un inventaire local ;
- 10 - la figure 5 illustre schématiquement certaines étapes mises en œuvre dans un module d'inventaire de synthèse ; et
- la figure 6 illustre un exemple d'une séquence d'étapes permettant d'obtenir un inventaire global sous forme de fichier XML.
- 15

De façon générale, l'invention permet d'effectuer un inventaire des composants matériels et/ou logiciels d'un cluster ou d'une partie d'un cluster en combinant une détermination locale automatique des composants de certains éléments ayant des capacités de traitement de données (par exemple des équipements de types prédéterminés), ou de tous, et une synthèse globale des composants de l'ensemble de ces éléments afin d'automatiser des opérations de gestion optimisée du cluster.

- 20

A ces fins, chaque élément visé (ayant des capacités de traitement de données), par exemple chaque équipement d'un type prédéterminé, typiquement chaque nœud, d'un cluster comprend un module d'inventaire local tandis qu'un élément de gestion, par exemple de nœud de management, comprend un module d'inventaire de synthèse. Selon les besoins, le module d'inventaire de synthèse peut interroger les modules d'inventaire local et les modules d'inventaire local peuvent, de façon automatique, transmettre des informations relatives aux composants de l'élément considéré, par exemple lorsqu'un événement déterminé est identifié dans cet élément.

- 25
- 30

Les éléments ne disposant pas de capacité de traitement de données, par exemple des commutateurs, peuvent être interrogés par des éléments en disposant, par exemple des nœuds d'administration, qui peuvent alors établir des inventaires locaux susceptibles d'être agrégés avec d'autres inventaires locaux et d'être utilisés pour mettre à jour un ou plusieurs inventaires partiels et/ou global.

La figure 3 illustre schématiquement un exemple de mise en œuvre du procédé selon l'invention dans un cluster 300.

Comme illustré, le cluster 300 comprend un ensemble d'éléments ayant des capacités de traitement de données, ici les nœuds 305-1 à 305-5 (génériquement référencés 305) et le nœud de management 310, connectés les uns aux autres via le réseau de communication 315 (comprenant ici des liens et des commutateurs, non représentés).

Chaque nœud 305 comprend un module d'inventaire local génériquement référencé 320. Ainsi, par exemple, le nœud 305-1 comprend le module d'inventaire local 320-1 et le nœud 305-2 comprend le module d'inventaire local 320-2. Le nœud de management 310 comprend un module d'inventaire de synthèse 325.

Un inventaire local est réalisé par un module d'inventaire local à l'aide de commandes standard, par exemple de commandes Linux. Ces commandes sont regroupées en séquences, par exemple une séquence de commandes écrite conformément au langage PERL. Ces commandes ont pour objet d'obtenir des informations relatives aux composants matériels et logiciels de l'élément considéré.

Un exemple d'un extrait d'une telle séquence est présenté en annexe (code 1). Les commandes données ici visent l'obtention de données relatives à un connecteur Ethernet d'un nœud. La commande *ethtool* permet d'obtenir ces données qui sont alors traitées pour en extraire le nom et la référence de microcode (*firmware*) ainsi que le type et la version du composant, la version de son microcode et une description du bus. Des commandes équivalentes sont utilisées pour obtenir des informations relatives à d'autres composants (microprocesseurs, barrettes mémoires, disques durs, etc.).

Ces commandes peuvent notamment consister en des interrogations des composants visés ou en l'obtention, typiquement la lecture, de données de configuration de l'élément considéré ou de données de journaux de bord systèmes, c'est-à-dire de données d'exécution telles que celles présentées en annexe (Journal de bord syslog). Cet exemple indique notamment des nombres de mémoires et de microprocesseurs disponibles ainsi que l'information selon laquelle ces nombres ont été correctement transmis.

Les composants matériels d'un nœud sont, par exemple, les unités centrales de traitement ou microprocesseurs 204, les composants de mémoire vive 206, les interfaces de communication 208 et les moyens de stockage interne 212 illustrés sur la figure 2. Les composants logiciels sont typiquement des modules du noyau du système d'exploitation du nœud ainsi que des modules liés à ce noyau.

A titre d'illustration, les caractéristiques obtenues sont les suivantes :

- composants matériels (processeurs, mémoires, etc.) :
 - o nom, type, vendeur, modèle, localisation, numéro de série, référence, description et microcode ; et,
 - o quantité de chaque composant ;
- composants logiciels (dont le système d'exploitation) :
 - o nom, type, version, date d'installation.

Un inventaire local est ici réalisé suite à la détection d'un événement particulier. Un tel événement peut notamment être lié à un changement d'état de l'élément considéré (c'est-à-dire l'élément dans lequel est effectué l'inventaire local), par exemple la détection d'une panne, ou à une commande. Cette dernière peut être générée localement, par exemple une commande de redémarrage (généralement appelée *reboot* en terminologie anglo-saxonne) ou reçue d'un dispositif distant, par exemple une commande d'exécution du module d'inventaire local reçue du module d'inventaire de synthèse du nœud de management comme illustré sur la figure 3 par les flèches en trait pointillé.

Lorsqu'un inventaire local a été établi, il est transmis au module d'inventaire de synthèse du nœud de management, comme illustré sur la figure 3 par les flèches unidirectionnelles en trait continu, pour, le cas échéant, former

un inventaire global. L'inventaire global consiste par exemple en la concaténation d'inventaires locaux (pouvant notamment comprendre l'inventaire local du nœud de management). L'inventaire global est typiquement mémorisé dans une base de données accessible par le nœud de management. Il peut également être mémorisé localement par ce dernier.

Les nœuds 305 et 310 peuvent en outre comprendre un module d'inventaire local permettant l'établissement d'inventaires locaux d'éléments ne disposant pas de capacité de traitement de données (non représentés sur la figure 3).

Selon un mode de réalisation particulier, les inventaires locaux ainsi que l'inventaire global sont échangés et/ou stockés sous forme de fichiers au format XML (sigle d'*Extensible Markup Language* en terminologie anglo-saxonne).

A titre d'illustration, un extrait d'un fichier au format XML est présenté en annexe (Extrait de fichier d'inventaire au format XML). Bien que l'exemple donné ici vise plusieurs composants de type différents, il est simplifié, un élément pouvant notamment comprendre plusieurs dizaines ou centaines d'équipements de mêmes types ou de types différents.

Selon cet exemple, l'inventaire local fournit des caractéristiques de l'élément concerné, ici un équipement, par exemple son nom (*nom_1*), son type (*Node*), son vendeur (*vendeur_1*), son modèle (*modele_1*), son adresse IP (sigle d'Internet Protocol en terminologie anglo-saxonne) (*12.18.64.254*), sa fonction (surveillance ou *mngt*), son statut (*up*) ainsi que sa position géographique (référence d'armoire informatique et position dans cette dernière). Il fournit également des caractéristiques sur les composants logiciels installés dans l'élément concerné, notamment leur nom (par exemple *logiciel_1*), leur type (par exemple *install*), leur version (par exemple *1.12*) et leur date d'installation (par exemple *2011-06-07 08:04:45*). L'inventaire local fournit en outre des caractéristiques sur des composants matériels, par exemple des systèmes, microprocesseurs, mémoires vives et disques durs, de l'élément concerné, telles que leur nom (par exemple *bullx*), leur type (par exemple *system*), leur vendeur (par exemple *Bull SAS*), leur modèle (par

exemple *modele_1*) et, le cas échéant, leur numéro de série (par exemple *CR-1763578DF485*), une référence (ou *part number* en terminologie anglo-saxonne), une description et des informations relatives au microcode telles qu'un nom (par exemple *BIOS*), une version (par exemple *1.0b*,
5 *Revision:8.16(08/14/2009)*) et une description (par exemple *Vendor from American Megatrends Inc.*).

L'inventaire global (ou un ou plusieurs inventaires partiels) peut alors être utilisé pour automatiser des opérations de gestion optimisée du cluster, l'optimisation étant liée à l'inventaire global et/ou à un ou plusieurs inventaires
10 partiels.

Ainsi, par exemple, en connaissant les caractéristiques des nœuds d'un cluster, en termes de disponibilité et de performance, il est possible d'améliorer la répartition de charge en fonction des processus à exécuter. A ces fins, un algorithme standard d'allocation de ressources identifiées peut être
15 utilisé.

Toujours à titre d'illustration, l'inventaire permet de vérifier le bon déroulement de mises à jour de composants logiciels, à l'échelle du cluster, d'un équipement ou plus généralement d'un élément ayant des capacités de traitement de données. De même, l'inventaire permet de vérifier le matériel
20 devant être présent dans un équipement au regard du matériel attendu contractuellement. De telles vérifications peuvent être effectuées en comparant les caractéristiques des composants du cluster, telles que définies dans l'inventaire global et/ou dans un ou plusieurs inventaires locaux avec des données prédéterminées, par exemple des données de configuration du cluster.

La figure 4 illustre schématiquement certaines étapes mises en œuvre dans un élément d'un cluster, par exemple un nœud, pour effectuer un inventaire local.
25

Comme illustré, une première étape (étape 400) a pour objet la détection d'un événement particulier, par exemple un événement appartenant à
30 une liste d'événements prédéterminés. Un tel événement peut notamment être lié à l'élément pour lequel l'inventaire doit être établi ou à un événement

extérieur tel qu'une demande d'établissement d'un inventaire local reçue d'un nœud de management.

Dans des étapes suivantes (étapes 405 et 410), des composants matériels et/ou logiciels sont identifiés. Comme indiqué précédemment, ces composants peuvent être identifiés à partir d'interrogations ou de lectures pouvant prendre la forme d'une séquence de commandes, notamment d'une séquence de commandes décrite dans le langage PERL. Des caractéristiques de ces composants, par exemple des références de version et des quantités, sont avantageusement obtenues de façon similaire et associées aux composants identifiés.

Les composants identifiés et les caractéristiques associées obtenues sont alors utilisés pour former un inventaire local, de préférence selon un format prédéterminé, comme décrit précédemment (étape 415), qui est transmis à un module d'inventaire de synthèse pouvant notamment être mis en œuvre dans un nœud de management (étape 420), pour être agrégé avec d'autres inventaires locaux pour former un inventaire partiel et/ou global.

Typiquement, l'implémentation de l'algorithme décrit en référence à la figure 4 est scindée en deux parties. Une première partie, référencée 425, comprenant l'étape 400, est intrinsèquement liée au système d'exploitation mis en œuvre dans l'élément considéré (ou en lien avec ce dernier) afin de détecter un événement particulier et, en réponse à cette détection, lancer l'exécution d'un module de construction et de transmission d'un inventaire local. La seconde partie, référencée 430, correspondant au module de construction et de transmission d'un inventaire local, comprend ici les étapes 405 à 420. Comme indiqué précédemment, cette seconde partie utilise par exemple le langage PERL.

La figure 5 illustre schématiquement certaines étapes mises en œuvre dans un module d'inventaire de synthèse. Comme décrit précédemment, un module d'inventaire de synthèse appartient typiquement à un nœud de management.

Une première étape (étape 500), optionnelle (comme représenté par l'utilisation de traits pointillés), a pour objet l'émission d'une requête

d'inventaire. Selon les besoins, une telle requête est adressée à un élément particulier, conformément à un mode de communication appelé *unicast*, à un ensemble d'éléments, par exemple un îlot, conformément à un mode de communication appelé *multicast* ou à tous les éléments conformément à un mode de communication appelé *broadcast*. La requête peut notamment prendre la forme d'un message, sa transmission utilisant un protocole de communication standard

5
10 Dans une étape suivante (étape 505), un inventaire local est reçu d'un élément, par exemple d'un nœud, ce dernier pouvant utiliser un algorithme tel que celui décrit en référence à la figure 4.

L'inventaire reçu est alors utilisé pour mettre à jour (étape 510) un ou plusieurs inventaires partiels, par exemple un inventaire lié à un îlot et un inventaire lié à un ensemble d'îlots, et/ou un inventaire global. Une telle étape consiste par exemple à modifier, dans un ou des inventaires partiels et/ou global, les données correspondant aux informations reçues (inventaire local).

15 Il est observé ici que si le ou les inventaires partiels et ou global n'existent pas, il est ou ils sont créés.

Comme illustré par la flèche en trait pointillé, les étapes 505 et 510 sont avantageusement répétées pour tous les inventaires locaux reçus.

20 Le ou les inventaires partiels et/ou global peuvent alors être utilisés pour automatiser des opérations de gestion optimisée du cluster, par exemple en optimisant la répartition de la charge (étape 515-1), en vérifiant la conformité du cluster (étape 515-2) et/ou en vérifiant une mise à jour d'un ou de plusieurs éléments du cluster (étape 515-n). Les algorithmes d'optimisation de la répartition de la charge, de vérification de la conformité du cluster et de vérification de mise à jour d'un ou de plusieurs éléments du cluster sont ici des algorithmes standard.

25 Les algorithmes décrits en référence aux figures 4 et 5 sont mis en œuvre dans des éléments ayant des capacités de traitement de données, typiquement des équipements tels que des nœuds dont l'architecture est similaire à celle décrite en référence à la figure 2.

30

La figure 6 illustre un exemple d'une séquence d'étapes permettant d'obtenir un inventaire global sous forme de fichier XML.

Une première étape (étape 600) vise ici l'exécution de la commande « *clminventory inventory -name nom_1* » ayant pour objet l'exécution de la fonction « *clminventory* » avec comme paramètre le nom « *nom_1* », qui permet
5 l'obtention de l'inventaire local de l'élément dont le nom lui est donné en paramètre.

Dans une étape suivante (étape 605), le type de l'élément dont l'inventaire local doit être établi, c'est-à-dire le type de l'élément ayant pour nom
10 « *nom_1* » est déterminé. Le type de l'élément est déterminé à partir de son nom à l'aide de données de configuration du cluster. Si l'élément considéré n'a pas de capacités de traitement de données, l'inventaire local est établi par un autre élément, de façon classique (non représenté).

Au contraire, si l'élément considéré dispose de capacités de
15 traitement de données, une instruction lui est adressée, par exemple sous forme d'événement dans un message qui lui est transmis, afin qu'en réponse il transmette son inventaire local (étape 610).

L'élément visé, c'est-à-dire l'élément nommé « *nom_1* », crée son inventaire local comme décrit précédemment (étape 615), en utilisant par
20 exemple des données d'exécution (*syslog*). L'inventaire local est ici créé sous forme de fichier XML. Il est ensuite transmis à l'élément à l'origine de la requête, c'est-à-dire l'élément ayant lancé l'inventaire à distance.

Après avoir obtenu le fichier au format XML représentant l'inventaire local (étape 620), le contenu de ce fichier est utilisé pour mettre à jour un
25 inventaire partiel ou global du cluster (étape 625) se présentant également sous forme d'un fichier XML qui est ensuite stocké (étape 630). La mise à jour de l'inventaire partiel ou global peut consister à ajouter des informations manquantes et/ou à modifier des informations préalablement mémorisées.

Les étapes 600 à 610 et 620 à 630 sont ici mises en œuvre dans un
30 élément 635 duquel il est possible de lancer automatiquement ou manuellement un inventaire d'un élément du cluster auquel il appartient. L'élément 635 est typiquement un nœud d'administration. L'étape 615 est mise en œuvre dans un

élément 640 du cluster ayant des capacités de traitement de données, par exemple un nœud de calcul.

Naturellement, pour satisfaire des besoins spécifiques, une personne compétente dans le domaine de l'invention pourra appliquer des modifications
5 dans la description précédente.

ANNEXE

```

# kawela firmware
my $kawela = qx(ethtool -i eth0);
5 my $kawelaComponent = Bull::CLM::Inventory::Component->new({'name','Kawela'});
my $kawelaFirmware = Bull::CLM::Inventory::Firmware->new({'name','Firmware Kawela'});
my @k_lines = split( /\n/, $kawela );
foreach my $line (@k_lines) {
    if( $line =~ /^driver: (.*)/){
10     $kawelaComponent->{'type'} = $1;
    }
    elsif( $line =~ /^version: (.*)/){
        $kawelaComponent->{'description'} .= ' version : '.$1;
    }
15     elsif( $line =~ /^firmware-version: (.*)/){
        $kawelaFirmware->{'version'} = $1;
    }
    elsif( $line =~ /^bus-info: (.*)/){
20     $kawelaComponent->{'description'} = ' bus-info : '.$1;
    }
}
}

```

Code 1

```

25 1308061409 2011 Jun 14 16:23:29 berlin32 local0 info inventory [inventory@clmBull
action="cpu_memory" memory_available="36" cpu_available="8" host="nom_1"
status="success"] Cpu_available and memory_available successfully sent

```

Journal de bord syslog

```

30 <?xml version="1.0" encoding="UTF-8"?>
<bcm:configuration xmlns:bcm="http://www.xml.bcm.bull/inventory">
  <date>2011-06-14 12:33:06</date>
  <equipment>
    <name>nom_1</name>
    <type>Node</type>
35    <vendor>vendeur_1</vendor>
    <model>modele_1</model>
    <ip>12.18.64.254</ip>
    <function>mngt</function>
    <status>up</status>
40    <location>
      <rackLabel>M3-0</rackLabel>
      <x>A</x>
      <y>3</y>
      <z>K</z>
45    <slot>1</slot>
    </location>
    <software>
      <name>logiciel_1</name>
      <type>install</type>
50      <version>1.12</version>
      <installDate>2011-06-07 08:04:45</installDate>
    </software>
  </equipment>
</configuration>

```

```

5     <name>bullx</name>
      <type>System</type>
      <vendor>Bull SAS</vendor>
      <model>modele_1</model>
      <serialNumber>CR-1763578DF485</serialNumber>
      <firmware>
10        <name>BIOS</name>
          <version>1.0b, Revision:8.16 (08/14/2009)</version>
          <description>Vendor from American Megatrends Inc.</description>
        </firmware>
      </component>
      <component>
15        <name>CPU_1 </name>
          <type>Processor</type>
          <vendor>Intel</vendor>
          <model>XeonIntel(R) Xeon(R) CPU X5550 @ 2.67GHz</model>
          <serialNumber>13547RE2156</serialNumber>
          <partNumber>1385735465499FER546DA</partNumber>
          <description>Core Count:4, Core Enabled:4, Thread Count:4</description>
20        </component>
        <component>
          <name>P1_DIMM1A#0</name>
          <type>Memory</type>
          <vendor>Qimonda</vendor>
25          <model>DIMM Type:Other</model>
          <location>P1_DIMM1A (BANK0)</location>
          <serialNumber>2665C30329</serialNumber>
          <partNumber>IMSH2G13A1F1C-10F</partNumber>
          <description>Size:2048 MB</description>
30        </component>
        <component>
          <name>host6/disk#0</name>
          <type>Disk</type>
          <vendor>SEAGATE</vendor>
35          <model>ST314664855SS</model>
          <location>Enclosure 0, Slot 0</location>
          <serialNumber>3LN7BHIV900009922LBM1</serialNumber>
          <description>Size:140014 MB, Protocol:SAS, Status:OK, </description>
40        </component>
      </equipment>
      <checksumEquipment>0b0eefac72d93530243f4aef134765</checksumEquipment>
</bcm:configuration>

```

Extrait de fichier d'inventaire au format XML

REVENDEICATIONS

1. Procédé pour ordinateur pour identifier dynamiquement des composants matériels et/ou logiciels (204, 206, 208, 210) d'au moins un nœud d'un cluster (300) comprenant une pluralité de nœuds (305, 310) et automatiser des opérations de gestion optimisée dudit cluster, ledit au moins un nœud appartenant à ladite pluralité de nœuds et chaque nœud de ladite pluralité de nœuds ayant des capacités de traitement de données, ce procédé étant caractérisé en ce qu'il comprend les étapes suivantes,

- dans ledit au moins un nœud (305), appelé au moins un premier nœud,
 - o établissement (415) d'au moins un inventaire local, ledit au moins un inventaire local comprenant au moins une caractéristique d'au moins un composant matériel et/ou logiciel dudit au moins un premier nœud ; et,
 - o transmission (420) dudit au moins un inventaire local à un nœud de ladite pluralité de nœuds dudit cluster, distinct dudit au moins un premier nœud, appelé second nœud ;
- dans ledit second nœud,
 - o réception (505) dudit au moins un inventaire ;
 - o établissement (510) d'un inventaire partiel ou global comprenant au moins ladite au moins une caractéristique dudit au moins un composant matériel et/ou logiciel dudit au moins un premier nœud ; et,
 - o lancement d'une opération de gestion dudit cluster selon ledit inventaire partiel ou global.

2. Procédé selon la revendication 1 comprenant en outre, dans ledit au moins un premier nœud, une étape de détection (425) d'un événement, lesdites étapes d'établissement et de transmission dudit au moins un inventaire local étant effectuées en réponse à ladite détection.

3. Procédé selon la revendication 2 comprenant en outre, dans ledit second nœud, une étape d'émission d'une requête d'inventaire, ladite étape de

détection d'un événement comprenant une détection d'une réception de ladite requête.

4. Procédé selon la revendication 2 selon lequel ledit événement détecté est généré dans ledit au moins un premier nœud.

5. Procédé selon l'une quelconque des revendications précédentes comprenant en outre une étape d'identification (405, 410) dudit au moins un composant matériel et/ou logiciel dudit au moins un premier nœud, ledit au moins un inventaire local comprenant un identifiant dudit au moins un composant matériel et/ou logiciel dudit au moins un premier nœud.

6. Procédé selon l'une quelconque des revendications précédentes selon lequel ladite opération de gestion dudit cluster comprend une optimisation de répartition de charge entre plusieurs composants matériels de nœuds de ladite pluralité de nœuds, une vérification de conformité d'au moins un composant matériel et/ou logiciel d'au moins un nœud de ladite pluralité de nœuds et/ou une vérification de mise à jour d'au moins un composant logiciel d'au moins un nœud de ladite pluralité de nœuds.

7. Procédé selon l'une quelconque des revendications précédentes selon lequel lesdites étapes de réception dudit au moins un inventaire et d'établissement d'un inventaire partiel ou global sont répétées pour tous les nœuds de ladite pluralité de nœuds d'un type prédéterminé.

8. Procédé selon l'une quelconque des revendications précédentes selon lequel au moins un inventaire local d'un élément dudit cluster n'ayant pas de capacité de traitement de données est établi par ledit second nœud, au moins une donnée dudit inventaire local dudit élément n'ayant pas de capacité de traitement de données étant utilisée pour l'établissement dudit inventaire partiel ou global.

9. Programme d'ordinateur comprenant des instructions adaptées à la mise en œuvre de chacune des étapes du procédé selon l'une quelconque des revendications précédentes lorsque ledit programme est exécuté sur un ordinateur.

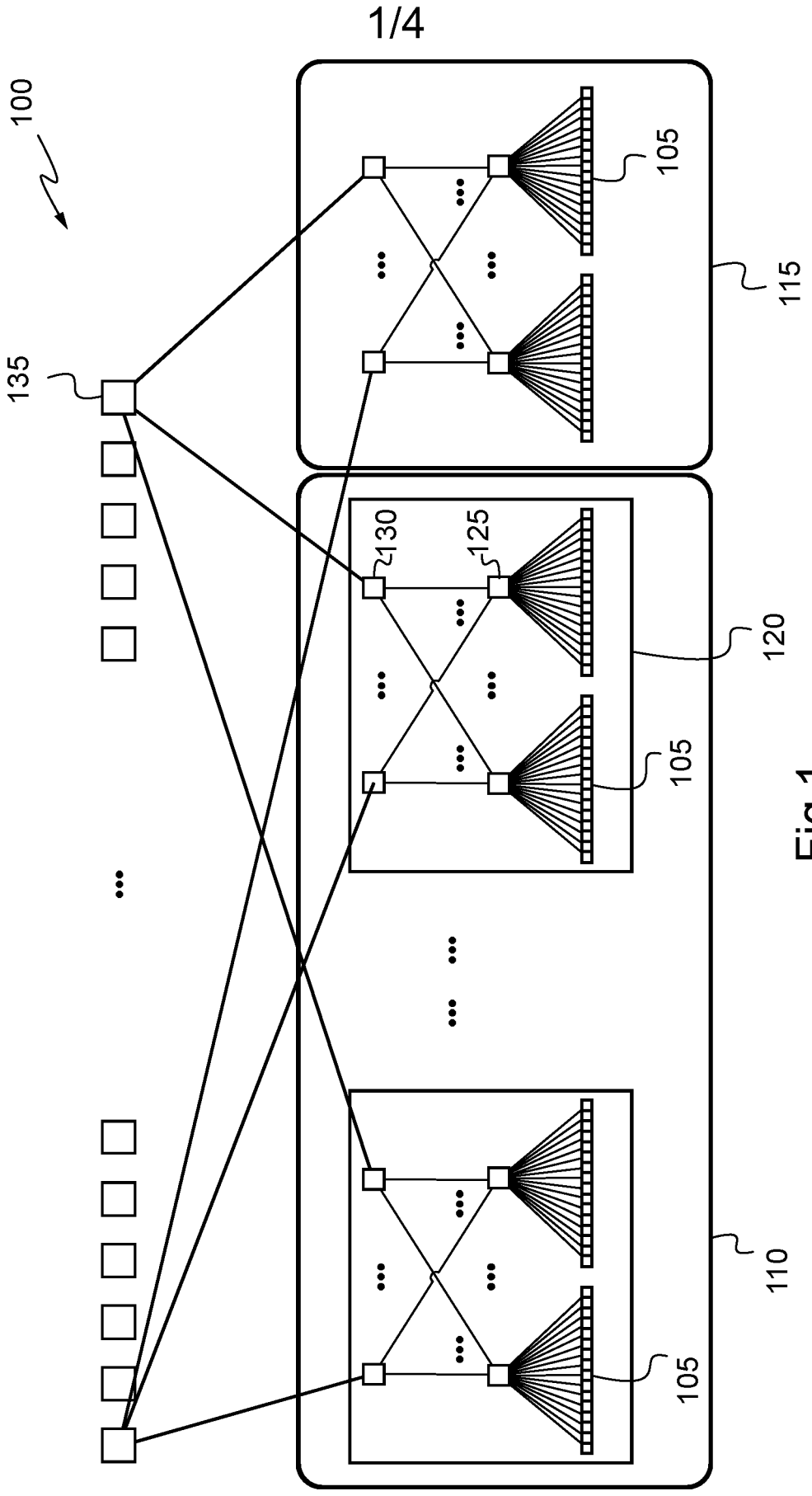
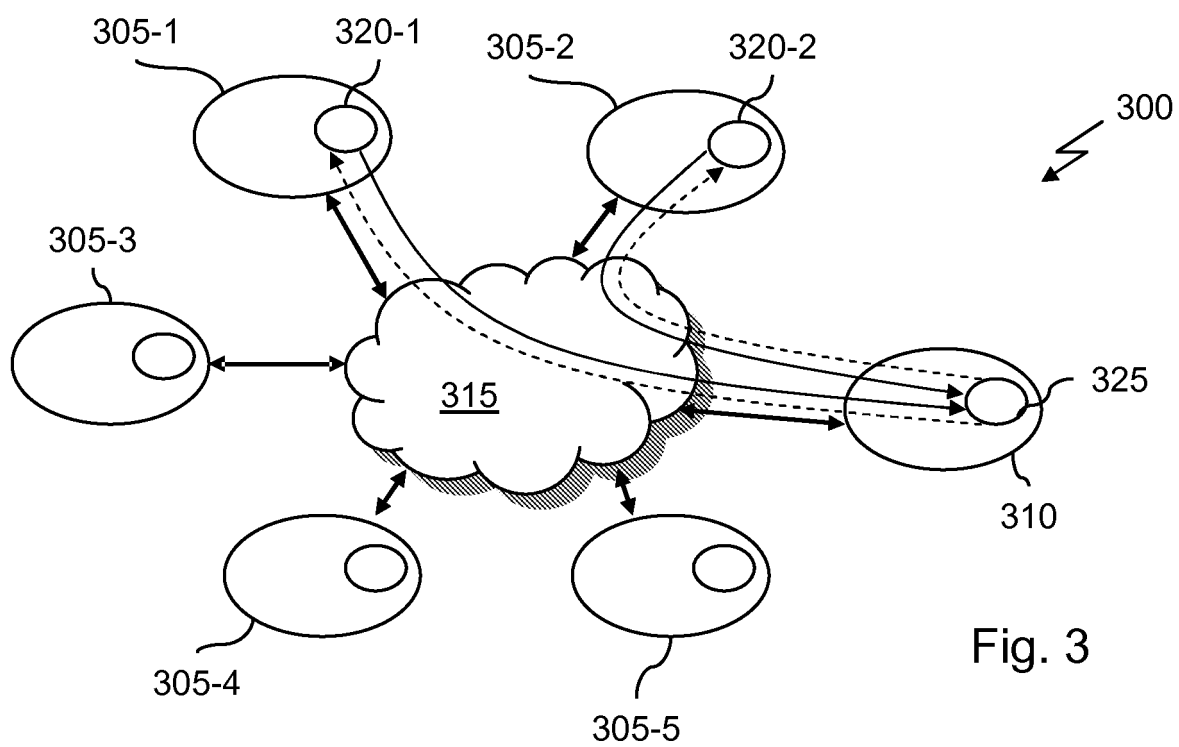
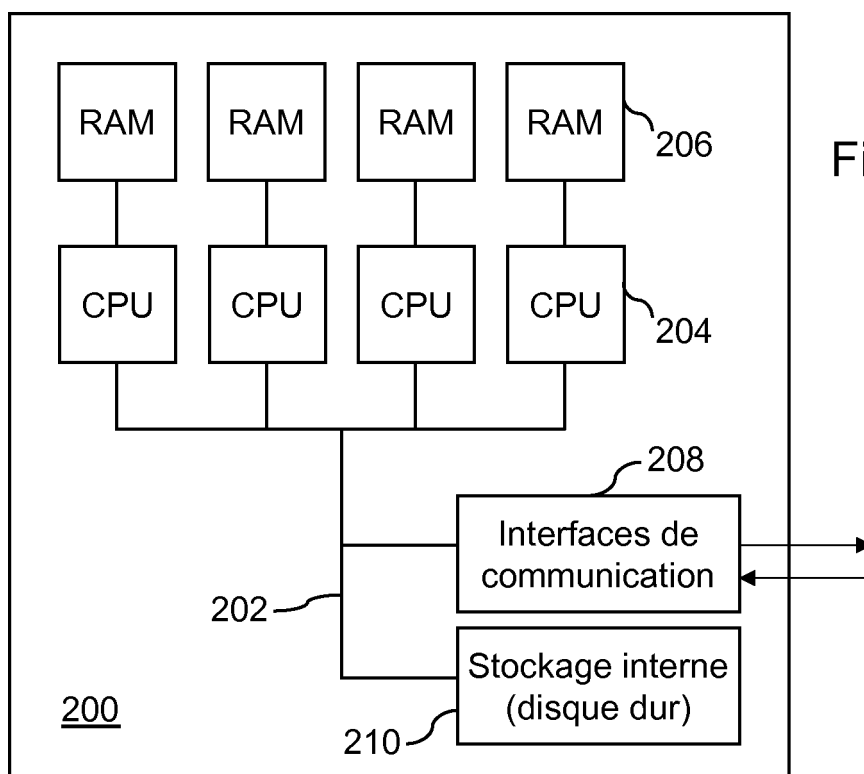
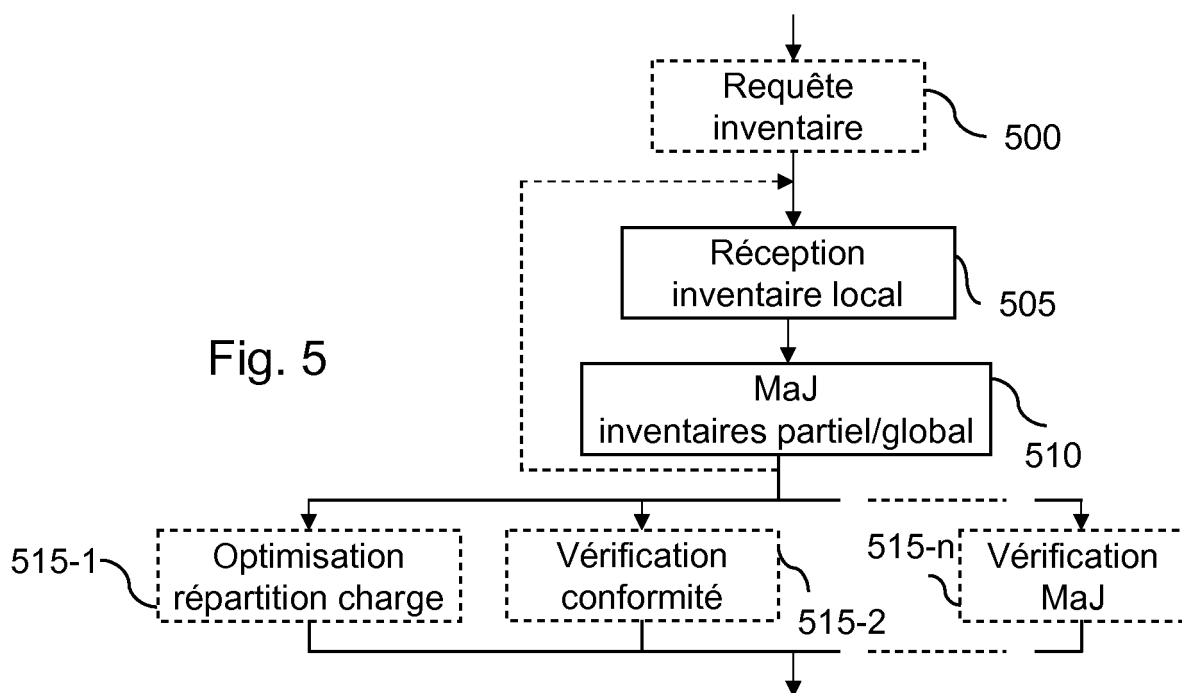
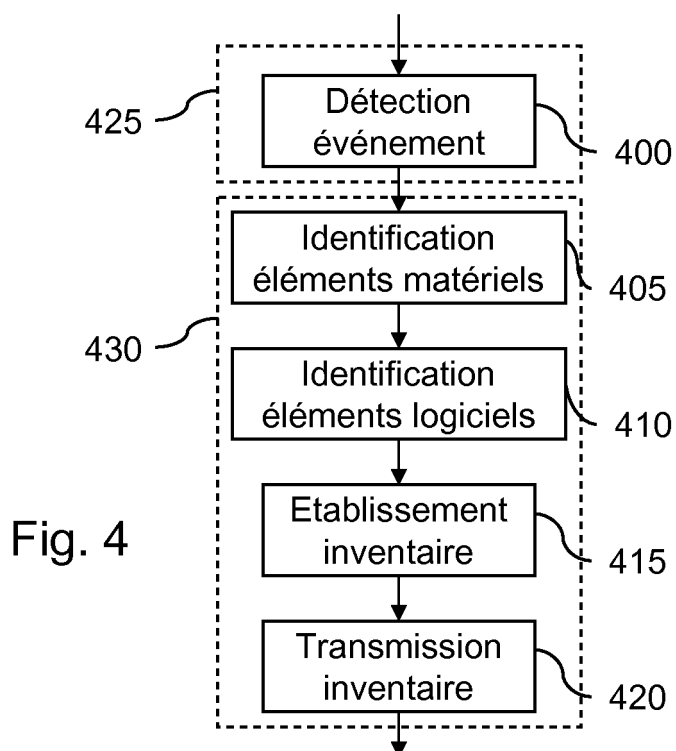


Fig.1

2/4



3/4



4/4

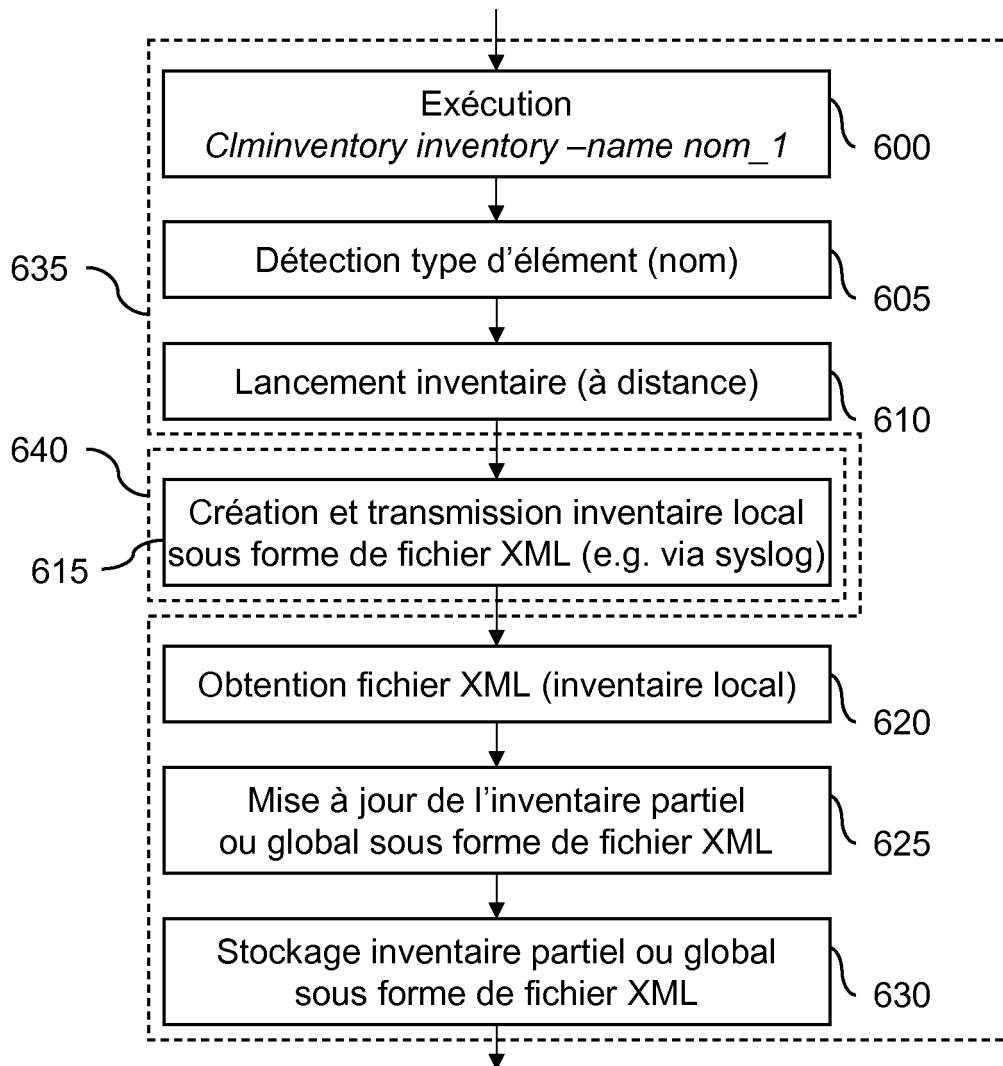


Fig. 6



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

N° d'enregistrement national

établi sur la base des dernières revendications déposées avant le commencement de la recherche

FA 755338
FR 1155752

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	US 2008/049644 A1 (HALBERT SCOTT [US]) 28 février 2008 (2008-02-28)	1,5-9	G06Q10/00 G06F9/50
Y	* abrégé; figure 3 5 6 10 11 * * alinéa [0006] - alinéa [0010] * * alinéa [0023] - alinéa [0036] * * alinéa [414647] * * alinéa [63747889100] * * alinéa [0103] - alinéa [0113] *	2-4	
Y	US 2002/165893 A1 (HUNSINGER KENNETH MARK [US] ET AL) 7 novembre 2002 (2002-11-07) * abrégé; figures 3A-5 * * alinéa [561318] * * alinéa [0038] - alinéa [0051] *	2-4	
A	SUKHYUN SONG ET AL: "Blue Eyes: Scalable and reliable system management for cloud computing", PARALLEL&DISTRIBUTED PROCESSING, 2009. IPDPS 2009. IEEE INTERNATIONAL SYMPOSIUM ON, IEEE, PISCATAWAY, NJ, USA, 23 mai 2009 (2009-05-23), pages 1-8, XP031487677, ISBN: 978-1-4244-3751-1 * page 1 - page 3, colonne de gauche * * page 5 *	1-9	DOMAINES TECHNIQUES RECHERCHÉS (IPC) G06F H04L G06Q
A	US 2008/168459 A1 (BENEDETTI FABIO [IT] ET AL) 10 juillet 2008 (2008-07-10) * alinéa [0005] - alinéa [0012] * * alinéa [0026] - alinéa [0041] * * alinéas [0044], [0046] * * alinéa [0051] - alinéa [0053] *	1-9	
		-/--	
Date d'achèvement de la recherche		Examineur	
1 février 2012		Bauer, Rodolphe	
CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention	
X : particulièrement pertinent à lui seul		E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.	
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie		D : cité dans la demande	
A : arrière-plan technologique		L : cité pour d'autres raisons	
O : divulgation non-écrite		
P : document intercalaire		& : membre de la même famille, document correspondant	

1
EPO FORM 1503 12.99 (P04C14)



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 755338
FR 1155752

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	JACOB LIBERMAN ET AL: "Integrating hardware management with cluster administration toolkits", CLUSTER COMPUTING, 2007 IEEE INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 17 septembre 2007 (2007-09-17), pages 468-472, XP031324128, ISBN: 978-1-4244-1387-4 * le document en entier * -----	1-9	
A	MAOYUAN XIE ET AL: "Cluster abstraction: towards uniform resource description and access in multicluster grid", COMPUTER AND COMPUTATIONAL SCIENCES, 2007. IMSCCS 2007. SECOND INTERNATIONAL MULTI-SYMPIOSIUMS ON, IEEE, PISCATAWAY, NJ, USA, 13 août 2007 (2007-08-13), pages 220-227, XP031338403, ISBN: 978-0-7695-3039-0 * le document en entier * * figure 1 3 * -----	1-9	
			DOMAINES TECHNIQUES RECHERCHÉS (IPC)
Date d'achèvement de la recherche		Examineur	
1 février 2012		Bauer, Rodolphe	
CATÉGORIE DES DOCUMENTS CITÉS			
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	

1

EPO FORM 1503 12.99 (P04C14)

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE
RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 1155752 FA 755338**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.

Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du **01-02-2012**

Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2008049644 A1	28-02-2008	AUCUN	

US 2002165893 A1	07-11-2002	AUCUN	

US 2008168459 A1	10-07-2008	CN 101206589 A	25-06-2008
		US 2008168459 A1	10-07-2008
