(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **3 000 578**

(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **13 63600**

(51) Int Cl[8] : ***G 06 F 15/16*** *(2013.01)*

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 26.12.13.

(30) **Priorité :** 28.12.12 US 13730450.

(43) **Date de mise à la disposition du public de la demande :** 04.07.14 Bulletin 14/27.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Ce dernier n'a pas été établi à la date de publication de la demande.*

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

(71) **Demandeur(s) :** *GENERAL ELECTRIC COMPANY* — US.

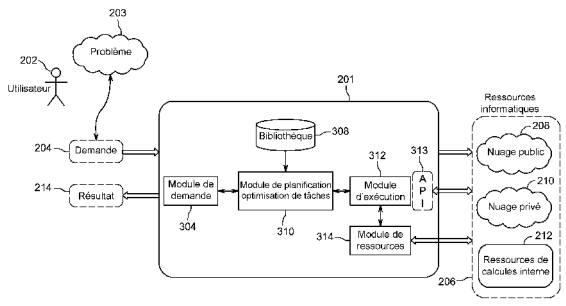
(72) **Inventeur(s) :** GILDER MARK RICHARD, WISE GERALD BOWDEN, YAN WEIZHONG et BRAHMAKSHATRIYA UMANG GOPALBHAI.

(73) **Titulaire(s) :** GENERAL ELECTRIC COMPANY.

(74) **Mandataire(s) :** CASALONGA & ASSOCIES.

(54) SYSTEME ET PROCEDE DE CALCUL PARTAGE UTILISANT UNE FOURNITURE AUTOMATISEE DE RESSOURCES INFORMATIQUES HETEROGENES.

(57) Système (201) pour calculs partagés, comportant un module de planification (310) de tâches conçu pour identifier une demande (204) de tâche comprenant des exigences de demande et comprenant une ou plusieurs tâches individuelles. Le système (201) comporte également un module de ressources (304) conçu pour déterminer un ensemble de ressources informatiques d'exécution parmi une série de ressources informatiques (206) d'après les exigences de la demande. Chaque ressource informatique de la série de ressources informatiques a une interface de programmation d'applications. La série de ressources informatiques (206) comprend des ressources d'informatique en nuage publique (208) et des ressources de calcul internes (210, 212). Le système (201) comporte en outre une pluralité de modules d'interfaçage (313), chaque module d'interfaçage étant conçu pour faciliter la communication avec les ressources informatiques (206) à l'aide de l'interface de programmation d'application correspondante. Le système (201) comporte également un module d'exécution (312) conçu pour identifier la module d'interfaçage approprié d'après la facilitation de la communication avec la ressource informatique d'exécution et pour transmettre des tâches à exécuter à la ressource informatique d'exécution à l'aide des modules d'interfaçage (313).



FR 3 000 578 - A1

Système et procédé de calcul partagé utilisant une fourniture automatisée de ressources informatiques hétérogènes

L'invention concerne globalement le calcul partagé et, plus particulièrement, un système informatique pour la fourniture de ressources informatiques hétérogènes, utilisant des ressources informatiques en nuage, ou « cloud computing » en anglais, et des ressources informatiques privées.

L'Apprentissage Automatique, une branche de l'intelligence artificielle, est une science qui a pour objet l'élaboration d'algorithmes qui analysent des données empiriques, réelles, la recherche de combinaisons dans ces données afin d'aboutir à des prévisions précises d'événements. Une partie essentielle et délicate de l'Apprentissage Automatique est la création de modèle, un processus de création d'un modèle qui repose sur un ensemble de "données d'entraînement". Ces données empiriques, observées et enregistrées, peuvent servir à généraliser d'après ces expériences antérieures. Au cours du processus de création de modèle, les praticiens trouvent le meilleur modèle pour le problème par tâtonnements, c'est-à-dire en générant de nombreux modèles différents d'après les données d'entraînement et en choisissant celui qui, par rapport à un ensemble de données de validation, répond le mieux aux critères de performances. La création de modèle est un problème de recherche complexe en ce qui concerne la structure de l'espace du modèle et les paramètres des diverses options de modélisation, et nécessite une grande puissance de calcul en raison des dimensions de l'espace de recherche.

La complexité croissante des calculs en Apprentissage Automatique nécessite davantage de moyens informatiques, sous la

forme de ressources informatiques plus rapides et/ou de ressources informatiques plus nombreuses. A la fin des années 1990, le projet SETI@home a mis en œuvre un mécanisme de calcul partagé pour exploiter des milliers d'ordinateurs individuels afin de contribuer à résoudre des tâches impliquant des calculs très lourds dans la Recherche d'une Intelligence Extra-Terrestre ("SETI"). SETI@home nécessitait l'analyse d'énormes masses de données d'observation fournies par un radiotélescope en quête d'émissions radio susceptibles de révéler l'existence de formes de vie intelligentes dans des galaxies lointaines.

Informatiquement, le problème a été réparti en fonction des données recueillies, en divisant le problème en millions de microrégions célestes. Pour faire face à la charge de travail, les données relatives à chaque microrégion ont été envoyées, par Internet, à des ordinateurs individuels. A mesure que chaque ordinateur finissait de traiter une unique microrégion, il devait renvoyer ses résultats à un serveur central qui les collectait. Pour SETI@home, des milliers d'ordinateurs ayant accès à Internet ont constitué un vaste environnement de calcul partagé exploité pour résoudre un problème impliquant des calculs complexes. De même, les problèmes d'Apprentissage Automatique représentent un problème faisant appel à des calculs complexes et qui peut par ailleurs aussi être divisé en éléments et traité à l'aide de nombreuses ressources informatiques indépendantes.

A la fin du XXème siècle, l' « informatique en nuage » est née en tant que source de ressources informatiques utilisables sur l'Internet. D'ordinaire, si un concepteur a besoin de ressources informatiques, le concepteur doit acheter du matériel, installer le matériel dans un centre de données, et installer et entretenir un système d'exploitation sur le matériel. Aujourd'hui, de nombreux

prestataires de services sur nuages proposent toutes sortes de services de ressources informatiques disponibles à la demande sur l'Internet, notamment "Infrastructure as a Service" ("IaaS") et "Platform as a Service" ("PaaS").

Avec IaaS et PaaS, des clients peuvent "louer" des ordinateurs individuels ou, plus souvent, des "serveurs virtuels" au prestataire de services sur nuage quand ils en ont besoin. Ces serveurs virtuels permettent de précharger une image d'un système d'exploitation et sont accessibles par l'Internet à l'aide d'une Interface de Programmation d'Application ("API"). Par exemple, un concepteur confronté à un problème nécessitant des calculs complexes pourrait utiliser l'API d'un prestataire de services sur nuage pour se procurer un serveur virtuel auprès du prestataire de services sur nuage, transmettre au serveur virtuel son code de logiciel ou ses instructions exécutables par ordinateur et réaliser sa tâche. Au terme de ce travail, le concepteur pourrait consulter ses résultats, puis arrêter le serveur virtuel. Ces services IaaS et PaaS offrent une possibilité pour ceux qui ont besoin de ressources informatiques supplémentaires, mais n'ont pas à y recourir régulièrement, ne disposent pas du budget ni de l'infrastructure leur permettant de posséder leur propre matériel spécifique. Pour les concepteurs qui veulent un environnement de conception souple pour des calculs dans le cadre de l'Apprentissage Automatique, l'informatique en nuage constitue une source prometteuse de ressources informatiques.

Selon un premier aspect, il est proposé un système de calculs partagés. Le système comporte un module de planification de tâches conçu pour identifier une demande de tâche(s). La demande de tâche(s) comprend une ou plusieurs conditions de demandes et une ou plusieurs tâches individuelles. Le système comporte également

un module de ressources conçu pour déterminer une série de ressources informatiques d'exécution parmi un ensemble de ressources informatiques au moins en partie d'après le/les besoin(s) de demande(s). A chaque ressource informatique de la série de ressources informatiques est associée une interface de programmation d'application. La série de ressources informatiques comprend soit au moins une ressource informatique interne soit au moins une ressource publique d'informatique en nuage. Le module de ressources attribue également une première ressource informatique, parmi la série de ressources informatiques d'exécution, à une première tâche individuelle parmi la/les tâches individuelles. Le système comporte en outre une pluralité de modules d'interfaçage. Chaque module d'interfaçage parmi la pluralité de modules d'interfaçage est conçu pour faciliter la communication avec une ou plusieurs ressources informatiques de la série de ressources informatiques à l'aide de l'interface de programmation d'application correspondante. Le système comporte aussi un module d'exécution conçu pour identifier un premier module d'interfaçage, parmi la pluralité de modules d'interfaçage, au moins en partie d'après la facilitation de la communication avec la première ressource informatique. Le module d'exécution est également conçu pour, à l'aide du premier module d'interfaçage, transmettre à la première ressource informatique la première tâche individuelle à exécuter.

Selon un autre aspect, il est proposé un procédé de calculs partagés. Le procédé est mis en œuvre par au moins un dispositif informatique comprenant au moins un processeur et au moins un dispositif de mémoire couplé au(x) processeur(s). Le procédé comporte l'identification d'une demande de tâche(s) comprenant une ou plusieurs tâches individuelles. Le procédé comporte

également l'identification d'un ou de plusieurs besoins de ressources informatiques pour la demande de tâche. Le procédé comporte en outre la détermination d'une série de ressources informatiques d'exécution parmi un ensemble de ressources informatiques au moins en partie d'après le/les besoin(s) de ressources informatiques. A chaque ressource informatique de la série de ressources informatiques est associée une interface de programmation d'application. La série de ressources informatiques comprend soit au moins une ressource informatique interne soit au moins une ressource informatique externe. Le procédé comporte aussi l'attribution d'une première ressource informatique, parmi la série de ressources informatiques d'exécution, à une première tâche individuelle parmi la/les tâches individuelles. Le procédé comporte en outre l'identification d'une pluralité de modules d'interfaçage. Chaque module d'interfaçage parmi la pluralité de modules d'interfaçage est conçu pour faciliter la communication avec une ou plusieurs ressources informatiques de la série de ressources informatiques à l'aide de l'interface de programmation d'application correspondante. Le procédé comporte aussi la sélection d'un premier module d'interfaçage, parmi une pluralité de modules d'interfaçage, au moins en partie d'après la facilitation de la communication avec la première ressource informatique. Le procédé comporte également la transmission à la première ressource informatique, par le/les dispositif(s) informatique(s), à l'aide du premier module d'interfaçage, de la première tâche individuelle à exécuter.

Selon encore un autre aspect, il est proposé un système de calculs partagés. Le système comporte un module de planification de tâches conçu pour identifier une première demande de tâche et une seconde demande de tâche. Le système comporte également un

module de ressources conçu pour attribuer à la première demande de tâche une première ressource informatique parmi un premier ensemble de ressources informatiques d'exécution correspondant à un premier prestataire de services sur nuage. La première ressource informatique a une première interface de programmation d'application. Le module de ressources est également conçu pour attribuer à la seconde demande de tâche une seconde ressource informatique parmi un second ensemble de ressources informatiques d'exécution correspondant à un second prestataire de services sur nuage, et un ensemble de ressources de calcul internes. La seconde ressource informatique a une seconde interface de programmation d'application. Le système comporte en outre un premier module d'interfaçage conçu pour faciliter la communication avec la première ressource informatique à l'aide de la première interface de programmation d'application. Le système comporte aussi un second module d'interfaçage conçu pour faciliter la communication avec la seconde ressource informatique à l'aide de la seconde interface de programmation d'application. Le système comporte en outre un module d'exécution conçu pour transmettre à la première ressource informatique, à l'aide du premier module d'interfaçage, la première demande de tâche à exécuter. Le module d'exécution est également conçu pour transmettre à la seconde ressource informatique, à l'aide du second module d'interfaçage, la seconde demande de tâche à exécuter.

L'invention sera mieux comprise à l'étude détaillée de quelques modes de réalisation pris à titre d'exemples non limitatifs et illustrés par les dessins annexés sur lesquels :

-la Figure 1 est un schéma de principe d'un exemple de système de calcul utilisable pour la fourniture automatisée de

ressources informatiques hétérogènes pour Apprentissage Automatique ;

-la Figure 2 est un schéma d'un exemple d'environnement d'application qui comporte un système pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique utilisant le système informatique représenté sur la Figure 1 ;

-la Figure 3 est un schéma de l'exemple d'environnement d'application représenté sur la Figure 2, représentant les principaux organes du système pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique représenté sur la Figure 2 ;

-la Figure 4 est un diagramme de flux de données d'un exemple de module de demande représenté sur la Figure 3, destiné à recevoir et traiter une demande liée à l'Apprentissage Automatique ;

-la Figure 5 est un diagramme de flux de données de l'exemple de module de planification/optimisation du système représenté sur la Figure 3, destiné à préparer l'exécution de tâches ;

-la Figure 6 est un diagramme de flux de données de l'exemple de module d'exécution et du module de ressources du système représenté sur la Figure 3, ayant pour fonction d'attribuer des tâches à des ressources informatiques et de transmettre des tâches à exécuter ;

-la Figure 7 est un schéma de principe d'un exemple de procédé de fourniture de ressources informatiques hétérogènes pour Apprentissage Automatique utilisant le système représenté sur la Figure 3 ;

-la Figure 8 est un schéma de principe d'un autre exemple de procédé de fourniture de ressources informatiques hétérogènes pour

Apprentissage Automatique utilisant le système représenté sur la Figure 3 ;

-la Figure 9 est un schéma de principe représentant une première partie d'un exemple de structure de table de base de données pour le système représenté sur la Figure 3, représentant les principales tables utilisées par le module de demande représenté sur la Figure 3 ;

-la Figure 10 est un schéma de principe représentant une deuxième partie de l'exemple de structure de base de données pour le système 201 représenté sur la Figure 3, représentant les principales tables utilisées par le module de planification/optimisation de tâches représenté sur la Figure 3 ; et

-la Figure 11 est un schéma de principe représentant une troisième partie de l'exemple de structure de base de données pour le système représenté sur la Figure 3, représentant les principales tables utilisées par le module d'exécution et le module de ressources représentés sur la Figure 3.

Sauf indication contraire, les dessins présentés ici servent à illustrer des aspects novateurs essentiels de l'invention. On estime que ces aspects novateurs essentiels sont applicables dans toutes sortes de systèmes comportant une ou plusieurs formes de réalisation de l'invention. De la sorte, il est entendu que les dessins ne comprennent pas tous les aspects classiques dont les spécialistes ordinaires de la technique savent qu'ils sont nécessaires pour la pratique de l'invention.

Dans la description ci-après et les revendications seront cités un certain nombre de termes qui seront définis comme ayant les significations suivantes.

A moins que le contexte n'indique clairement le contraire, l'emploi des articles définis et indéfinis au singulier inclut le pluriel.

"Eventuel" ou "éventuellement" signifie que le fait ou la circonstance décrit à la suite peut survenir ou non et que la description couvre des cas où le fait survient et des cas où il ne survient pas.

Des termes d'approximation utilisés ici partout dans la description et les revendications peuvent être employés pour modifier toute représentation quantitative susceptible de varier sans provoquer de changement dans la fonction de base à laquelle ils se rapportent. De la sorte, une valeur modifiée par un ou plusieurs termes tels que "environ" et "sensiblement" ne doit pas se limiter à la valeur précise citée. Au moins dans certains cas, les termes d'approximation peuvent correspondre à la précision d'un instrument de mesure de la valeur. Ici et partout dans la description et les revendications, des limites de plages peuvent être combinées et/ou interchangées, ces plages sont identifiées et, à moins que le contexte ou la formulation n'indique le contraire, incluent toutes les plages partielles qu'elles contiennent.

Au sens de la présente description, l'expression "support non temporaire lisible par ordinateur" vise à représenter tout dispositif physique à base d'ordinateur, mis en œuvre dans tout procédé ou toute technologie pour le stockage de courte durée et de longue durée, dans tout dispositif, d'informations telles que des instructions, structures de données, modules et sous-modules de programmes ou autres données exploitables par ordinateur. Par conséquent, les procédés décrits ici peuvent être codés sous la forme d'instructions exécutables mises en œuvre sur un support physique non temporaire lisible par ordinateur dont, sans s'y

limiter, un périphérique de stockage et/ou un périphérique de mémoire. Ces instructions, lors de leur exécution par un processeur, amènent le processeur à exécuter au moins une partie des procédés décrits ici. De plus, au sens de la présente invention, l'expression "support non temporaire lisible par ordinateur" couvre tous les supports physiques lisibles par ordinateur dont, sans s'y limiter, des périphériques de stockage informatique non temporaire dont, sans s'y limiter, des supports rémanents ou non rémanents et des supports amovibles et non amovibles tels que des microprogrammes, des supports de stockage physiques et virtuels, des CD-ROM, des DVD et toute autre source numérique telle qu'un réseau ou l'Internet, ainsi que des moyens numériques encore à mettre au point, l'unique exception étant un signal temporaire en propagation.

Au sens de la présente description, l'expression "informatique en nuage" se rapporte globalement à, des services informatiques proposés sur l'Internet. En outre, au sens de la présente description, l'expression "prestataire de services sur nuage" se rapporte à la société ou l'entité proposant ou hébergeant le service informatique. Il existe de nombreux types de services informatiques relevant du "informatique en nuage", dont "Infrastructure as a Service" ("Iaas") et "Platform as a Service" ("Paas"). Par ailleurs, au sens de la présente description, le terme "IaaS" sert à désigner le service informatique impliquant la proposition de serveurs physiques ou virtuels à des clients. Au titre du modèle IaaS, le client "loue" un serveur physique ou virtuel au prestataire de services sur nuage, lequel fournit le matériel mais généralement pas le système d'exploitation ni des services d'applications d'un niveau supérieur. De plus, au sens de la présente description, le terme "PaaS" sert à désigner le service informatique proposant des serveurs physiques ou virtuels aux

clients, mais incluant aussi l'installation et la prise en charge du système d'exploitation, et éventuellement l'installation et la prise en charge de certaines applications de base telles qu'un serveur de base de données ou Web. Par ailleurs, au sens de la présente description, l'expression "informatique en nuage" et les termes "IaaS" et "PaaS" sont utilisés d'une façon interchangeable. Les systèmes et procédés décrits ici ne se limitent pas à ces deux modèles d'informatique en nuage. Tout service informatique permettant le fonctionnement des systèmes et procédés décrit ici peut être utilisé.

Au sens de la présente description, l'expression "nuage privé" se rapporte à une plate-forme de ressources informatiques similaire à l' « informatique en nuage » décrite plus haut, mais exploitée exclusivement pour un seul organisme. Par exemple, et sans s'y limiter, une grande société peut établir un nuage privé pour ses propres besoins informatiques. Au lieu d'acheter un matériel particulier pour divers projets ou services internes spécifiques, la société peut harmoniser ses ressources informatiques dans le nuage privé et permettre à ses concepteurs de mobiliser des ressources informatiques à l'aide du modèle sur nuage en assurant de la sorte une plus grande mobilisation de ses ressources informatiques dans l'ensemble de la société.

Au sens de la présente description, l'expression "ressources de calcul internes" se rapporte globalement à des ressources informatiques possédées ou autrement utilisables par l'entité employant les systèmes et procédés décrits ici, à l'exclusion des ressources publiques d'"informatique en nuage". Par ailleurs, au sens de la présente description, les nuages privés sont également considérés comme des ressources de calcul internes. En outre, au sens de la présente description, l'expression "ressources

informatiques externes" couvre les ressources publiques d'« informatique en nuage ».

Au sens de la présente description, le terme "fourniture" se rapporte au processus d'établissement d'une ressource informatique en vue de son utilisation. Pour rendre utilisable une ressource, il peut être nécessaire de "fournir" la ressource. Par exemple, et sans s'y limiter, quand un utilisateur sollicite une ressource informatique telle qu'un serveur virtuel auprès d'un prestataire de services sur nuage, l'utilisateur prend part à une transaction afin de "fournir" le serveur virtuel en vue de son utilisation par le client pendant un certain temps. La "fourniture" établit l'attribution de la ressource informatique à l'utilisateur. Dans le cadre des calculs sur nuage de serveurs virtuels, le processus de "fourniture" peut réellement amener le prestataire de services sur nuage à créer un serveur virtuel, voire à installer sur le serveur virtuel une image et des applications de base d'un système d'exploitation avant d'autoriser l'utilisateur à se servir de la ressource. Selon une autre possibilité, le terme "fourniture" sert aussi à faire référence au processus d'attribution d'une ressource informatique déjà disponible mais pour l'instant inutilisée. Par exemple, un serveur sur nuage qui a déjà été "fourni" par le prestataire de services sur nuage mais n'est pour l'instant pas occupé par une tâche de calcul peut être qualifié de "fourni" pour une nouvelle tâche de calcul quand il est affecté à cette tâche. Par ailleurs, au sens de la présente description, les termes "attribution", "affectation" et "fourniture", par rapport à des ressources d'informatique en nuage, s'emploient d'une manière interchangeable.

Au sens de la présente description, le terme "abandon" est le corollaire de "fourniture". L'"abandon" est le processus consistant à cesser l'utiliser la ressource informatique. Afin de rendre vacante la

ressource, il faut “abandonner” la ressource. Par exemple, et sans s’y limiter, quand un utilisateur a fini d’utiliser un serveur virtuel loué à un prestataire de services sur nuage, l’utilisateur “abandonne” le serveur virtuel. L’“abandon” informe le prestataire de services sur nuage de ce que la ressource n’est plus nécessaire à l’utilisateur ni utilisée par ce dernier, et de ce que la ressource peut être réattribuée.

Au sens de la présente description, le terme “algorithme” fait globalement référence à toute méthode pour résoudre un problème. Par ailleurs, au sens de la présente description, le terme “modèle” fait globalement référence à un algorithme pour résoudre un problème. En outre, au sens de la présente description, les termes “modèle” et “algorithme” sont employés d’une manière interchangeable. Plus particulièrement, dans le contexte de l’Apprentissage Automatique et de l’apprentissage contrôlé, un “modèle” comprend un ensemble de données recueillies auprès de quelque source de données réelles, dans lequel sont rassemblés un ensemble de variables d’entrées et leurs variables de sortie correspondantes. Quand le modèle est correctement configuré, il peut servir de prédicteur pour un problème si le modèle utilise des variables semblables à un problème. Un modèle peut, sans s’y limiter, être un classifieur à classe unique, un classifieur à plusieurs classes ou un prédicteur. Dans d’autres contextes, le terme “algorithme” peut faire référence à des méthodes pour résoudre d’autres problèmes, notamment, sans s’y limiter, la conception d’expériences et de simulations. Dans certaines formes de réalisation, un “algorithme” comprend un code source et/ou des instructions exécutables par ordinateur qui peuvent être partagées et utilisées pour “résoudre” le problème à l’aide d’une exécution par une ressource informatique.

Au sens de la présente description, le terme “tâche” sert à désigner globalement des travaux identifiés pour, sans s’y limiter, l’exécution, le traitement ou le calcul. La “tâche” peut être divisible en multiples tâches plus petites qui, lorsqu’elles sont exécutées et regroupées, permettent d’achever la “tâche”. Par ailleurs, au sens de la présente description, le terme “tâche” peut aussi servir à désigner une ou plusieurs des multiples petites tâches qui composent une tâche plus grande. En outre, au sens de la présente description, l’expression “tâche à exécuter” est employée d’une manière interchangeable avec “tâche” et peut aussi servir à désigner une “tâche” prête à être exécutée.

Au sens de la présente description, les termes “demande d’exécution”, “demande de tâche” et “demande” sont employés, d’une manière interchangeable, pour désigner le problème de calcul à résoudre à l’aide des systèmes et procédés décrits ici.

Au sens de la présente description, les termes “exigence”, “limitation” et “restriction” font globalement référence à un paramètre de configuration associé à une demande de tâche. Par exemple, et sans s’y limiter, quand un utilisateur saisit une demande de tâche qui définit l’utilisation d’un modèle particulier *M1*, l’utilisateur a spécifié une “exigence” que la demande soit exécutée à l’aide du modèle *M1*. Une “exigence” peut aussi être caractérisée comme une “limitation” ou une “restriction” quant à la demande de tâche. Par exemple, et sans s’y limiter, quand un utilisateur saisit une demande de tâche qui restreint le traitement de la demande aux seules ressources de calcul internes, cette restriction peut être caractérisée à la fois comme “exigence” de ce que “seules des ressources de calcul internes soient utilisées” et comme “limitation” ou “restriction” visant à ce que “aucune ressource informatique non interne ne puisse être utilisée pour traiter la demande”.

Au sens de la présente description, l'expression "ressources informatiques hétérogènes" fait référence à un ensemble de ressources informatiques qui diffèrent par un aspect concernant le système d'exploitation ou la configuration du processeur (à savoir une processeur unique ou plusieurs processeurs) et l'architecture de la mémoire (à savoir de 32 bits ou de 64 bits). Par exemple, et sans s'y limiter, si un ensemble de ressources informatiques comprend System X, qui exécute le système d'exploitation Linux, et System Y, qui exploite le système d'exploitation Windows™ Server 2003, l'ensemble de ressources informatiques est alors considéré comme "hétérogène". En outre, par exemple, et sans s'y limiter, si un ensemble de ressources informatiques comprend System 1, qui a un unique processeur à base d'Intel exécutant le système d'exploitation Linux, et System 2, qui a quatre processeurs à base d'Intel exécutant le système d'exploitation Linux, cet ensemble de ressources informatique est alors considéré comme "hétérogène".

Les exemples de systèmes et de procédés décrits ici permettent à un utilisateur de mobiliser harmonieusement une série diverse, hétérogène de ressources informatiques pour effectuer des tâches de calcul en recourant à divers prestataires de services d'informatique en nuage, à des nuages internes et autres ressources de calcul internes. Plus particulièrement, le système sert à rechercher des conceptions ou configurations de calcul optimales, telles que des modèles d'apprentissage automatique et des paramètres de modèles correspondants, en fournissant automatiquement ces tâches de recherche en recourant à diverses ressources informatiques proposées par divers prestataires de services informatiques. Une base de données d'algorithme contient diverses versions d'un code exécutable par ordinateur ou des binaires adaptés pour les diverses architectures de ressources

informatiques susceptibles d'être mobilisées. Un module d'exécution gère les diverses ressources informatiques et communique avec celles-ci par l'intermédiaire d'un module d'Interface de Programmation d'Application ("API"), permettant au système de communiquer avec divers prestataires de services d'informatique en nuage différents, ainsi que des ressources de calcul internes telles qu'un nuage privé ou une grappe de serveurs privés. Un utilisateur peut saisir une demande qui adapte ceux des algorithmes qui servent à traiter la demande, en spécifiant également les restrictions informatiques à utiliser pour l'exécution. Par conséquent, l'utilisateur peut présenter au système sa tâche impliquant énormément de calculs, personnalisée avec des exigences de performances et certaines restrictions, et de ce fait mobiliser harmonieusement une série de ressources informatiques potentiellement grande, diverse et hétérogène.

La Figure 1 est un schéma de principe d'un exemple de système informatique 120 utilisable pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique. Selon une autre possibilité, on peut employer toute architecture informatique permettant l'exploitation des systèmes et procédés décrits ici.

Dans l'exemple de forme de réalisation, le système informatique 120 comporte un périphérique de mémoire 150 et un processeur 152 coopérant avec le périphérique de mémoire 150 pour exécuter des instructions. Dans certaines formes de réalisation, des instructions exécutables sont stockées dans le périphérique de mémoire 150. Le système informatique 120 est configurable pour effectuer, en programmant le processeur 152, une ou plusieurs opérations décrites ici. Par exemple, le processeur 152 peut être programmé en codant une opération sous la forme d'une ou de

plusieurs instructions exécutables et en chargeant les instructions exécutables dans le périphérique de mémoire 150. Le processeur 152 peut comprendre une ou plusieurs unités de traitement, p.ex., sans s'y limiter, dans une configuration multinœuds.

Dans l'exemple de forme de réalisation, le périphérique de mémoire 150 est un ou plusieurs périphériques permettant le stockage et la consultation d'informations telles que des instructions exécutables et/ou autres données. Le périphérique de mémoire 150 peut comprendre un ou plusieurs supports physiques, non temporaires, lisibles par ordinateur, tels que, sans s'y limiter, une mémoire vive (RAM), une mémoire vive dynamique (DRAM), une mémoire vive statique (SRAM), un disque SSD, un disque dur, une mémoire morte (ROM), une mémoire morte programmable effaçable (EPROM), une mémoire morte programmable effaçable électriquement (EEPROM) et/ou une mémoire vive rémanente (NVRAM). Les types de mémoires ci-dessus ne sont que des exemples et ne sont donc nullement limitatifs quant aux types de mémoires utilisables pour le stockage d'un programme informatique.

Par ailleurs, dans l'exemple de forme de réalisation, le dispositif de mémoire 150 peut être conçu pour stocker des informations associées à la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique, dont, sans s'y limiter, des modèles d'Apprentissage Automatique, des interfaces de programmation d'applications, des ressources d'informatique en nuage et des ressources de calcul internes.

Dans certaines formes de réalisation, le système informatique 120 comprend une interface de présentation 154 couplée au processeur 152. L'interface de présentation 154 présente des informations, telles qu'une interface utilisateur et/ou une alerte,

à un utilisateur 156. Par exemple, l'interface de présentation 154 peut comprendre un adaptateur d'affichage (non représenté) qui peut être couplé à un périphérique d'affichage (non représenté) tel qu'un tube cathodique (CRT), un écran à cristaux liquides (LCD), un écran à DEL organiques (OLED) et/ou un dispositif portatif à écran. Dans certaines formes de réalisation, l'interface de présentation 154 comprend un ou plusieurs périphériques d'affichage. De plus, ou selon une autre possibilité, l'interface de présentation 154 peut comprendre un périphérique de sortie audio (non représenté) (p.ex. un adaptateur audio et/ou une enceinte).

Dans certaines formes de réalisation, le système informatique 120 comprend une interface utilisateur de saisie 158. Dans l'exemple de forme de réalisation, l'interface utilisateur de saisie 158 est couplée au processeur 152 et reçoit des instructions saisies par l'utilisateur 156. L'interface utilisateur de saisie 158 peut comprendre, par exemple, un clavier, un périphérique de pointage, une souris, un stylet et/ou un panneau tactile, p.ex. un pavé tactile ou un écran tactile. Un organe unique, tel qu'un écran tactile, peut servir de périphérique d'affichage de l'interface de présentation 154 ainsi que de l'interface utilisateur de saisie 158.

Par ailleurs, une interface de communication 160 est couplée au processeur 152 et est conçue pour communiquer avec un ou plusieurs autres périphériques tels que, sans s'y limiter, un autre système informatique 120, et tout périphérique apte à accéder au système informatique 120 dont, sans s'y limiter, un ordinateur portatif, un assistant numérique personnel (PDA) et un téléphone mobile. L'interface de communication 160 peut comprendre, sans s'y limiter, un adaptateur câblé de réseau, un adaptateur radioélectrique de réseau, un adaptateur pour télécommunications mobiles, un adaptateur de communication série et/ou un adaptateur

de communication parallèle. L'interface de communication 160 peut recevoir des données d'un ou de plusieurs dispositif distants et/ou émettre des données vers celui/ceux-ci. Par exemple, l'interface de communication 160 d'un premier système informatique 120 peut transmettre des informations de transaction à l'interface de communication 160 d'un autre système informatique 120. Le système informatique 120 peut être activé par internet pour des communications à distance, par exemple avec un ordinateur de bureau distant (non représenté).

L'interface de présentation 154 et/ou l'interface de communication 160 est/sont aussi, toutes deux, aptes à fournir des informations utilisables avec les procédés décrits ici, par exemple, à l'utilisateur 156 d'un autre dispositif. De la sorte, l'interface de présentation 154 et l'interface de communication 160 peuvent être appelés périphériques de sortie. De même, l'interface utilisateur de saisie 158 et l'interface de communication 160 sont aptes à recevoir des informations utilisables avec les procédés décrits ici et peuvent être appelés périphériques d'entrée.

En outre, le processeur 152 et/ou le périphérique de mémoire 150 peut/peuvent aussi coopérer avec un périphérique de stockage 162. Le périphérique de stockage 162 est tout matériel à fonctionnement informatique permettant de stocker et/ou de consulter des données telles que, mais d'une manière nullement limitative, des données associées à une base de données 164. Dans l'exemple de forme de réalisation, le périphérique de stockage 162 est intégré dans le système informatique 120. Par exemple, le système informatique 120 peut comprendre, comme périphérique de stockage 162, un ou plusieurs lecteurs de disques durs. De plus, par exemple, le périphérique de stockage 162 peut comprendre de multiples unités de stockage telles que des disques durs et des

disques DSS sous une configuration de regroupement redondant de disques peu onéreux (RAID). Le périphérique de stockage 162 peut comprendre un réseau de stockage (SAN), un système de stockage en réseau (NAS) et/ou un stockage sur nuage. Selon une autre possibilité, le périphérique de stockage 162 est à l'extérieur du système informatique 120 et est consultable à l'aide d'une interface de stockage (non représentée).

De plus, dans l'exemple de forme de réalisation, la base de données 164 comprend diverses données statiques et dynamiques associées, sans s'y limiter, à des modèles d'Apprentissage Automatique, des ressources d'informatique en nuage et des ressources de calcul internes.

Les formes de réalisation illustrées et décrites ici, ainsi que des formes de réalisation non spécifiquement décrites ici mais entrant dans le cadre d'aspects de l'invention, constituent des exemples de moyens pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique. Par exemple, le système informatique 120, et n'importe quel autre dispositif informatique similaire ajouté à celui-ci ou inclus dans celui-ci, quand ils sont intégrés l'un avec l'autre, comprennent des supports de stockage suffisants, lisibles par ordinateur, programmés avec des instructions suffisantes, exécutables par ordinateur, pour exécuter des processus et des techniques à l'aide d'un processeur décrit ici. En particulier, le système informatique 120 et n'importe quel autre dispositif informatique similaire ajouté à celui-ci ou inclus dans celui-ci, quand ils sont intégrés l'un avec l'autre, constituent un exemple de moyen pour enregistrer, stocker, consulter et afficher des données d'exploitation associées à un système (non représenté sur la Figure 1) pour la fourniture

automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique.

La figure 2 est un schéma d'un exemple d'environnement d'application 200 qui comporte un système 201 pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique à l'aide du système informatique 120 (représenté sur la Figure 1). Un utilisateur 202 conçoit un problème 203 et présente une demande 204 au système 201. Le système 201 coopère avec des ressources informatiques 206 afin de traiter la demande 204. Dans l'exemple de forme de réalisation, les ressources informatiques 206 consistent en un ou plusieurs nuages publics 208, un ou plusieurs nuages privés 210 et des ressources de calcul internes 212. En fonctionnement, le système 201 reçoit la demande 204 de l'utilisateur 202 et exécute automatiquement la demande à l'aide de l'ensemble des ressources informatiques hétérogènes 206, en permettant de ce fait à l'utilisateur 202 de ne pas se préoccuper des détails de l'exécution. Les détails du système 201 sont expliqués plus précisément ci-après.

La Figure 3 est un schéma de l'exemple d'environnement d'application 200 (représenté sur la Figure 2), représentant les principaux organes du système 201 pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique. L'utilisateur 202 crée une demande 204 et la présente à un module de demande 304. Le module de demande 304 traite la demande 204 en créant une ou plusieurs "tâches" à traiter. Un module de planification/ optimisation 310 de tâches analyse une bibliothèque 308 et, d'après la demande 204, sélectionne les modèles et paramètres qui conviennent le mieux pour servir à exécuter la tâche. Dans certaines formes de réalisation, la bibliothèque 308 est une base de données de modèles.

Par ailleurs, dans l'exemple de forme de réalisation, la bibliothèque 308 est une base de données d'algorithmes d'Apprentissage Automatique. Selon une autre possibilité, la bibliothèque 308 est une base de données d'autres algorithmes de calcul. Chaque modèle présent dans la bibliothèque 308 comprend un ou plusieurs ensembles d'instructions exécutables par ordinateur, compilées pour différentes architectures de matériel et de système d'exploitation. Les instructions exécutables par ordinateur sont des binaires précompilés pour une architecture donnée. Selon une autre possibilité, les instructions exécutables par ordinateur peuvent être un code source non compilé écrit en langage de programmation ou de script, notamment Java et C++. Le nombre d'algorithmes dans la bibliothèque 308 n'est pas fixe, c'est-à-dire que des algorithmes peuvent être ajoutés ou retirés. La taille des algorithmes d'apprentissage automatique présents dans la bibliothèque 308 peut être adaptable à celle des données.

En outre, dans l'exemple de forme de réalisation, un module 314 de ressources détermine et attribue un sous-ensemble de ressources informatiques, issu des ressources informatiques 206, qui convient pour la tâche. Une fois qu'un sous-ensemble de ressources informatiques a été attribué à la tâche, un module d'exécution 312 gère la présentation de la tâche aux ressources informatiques attribuées. Pour communiquer avec les diverses ressources informatiques 206, le module d'exécution 312 utilise des modules d'API 313. Le fonctionnement de chaque organe du système est expliqué en détail ci-après.

Sur les figures 4 à 8 est illustré le fonctionnement de chaque organe du système 201. La Figure 4 représente un exemple de module de demande 304. La Figure 5 représente un exemple de module de planification/optimisation 310 de tâches. La Figure 6

représente un exemple de module d'exécution 312 et de module 314 de ressources. La Figure 7 représente un exemple d'illustration du système 201 comprenant les organes des figures 4 à 6. Le fonctionnement de chaque organe du système est expliqué en détail ci-après.

Dans certaines formes de réalisation, les organes du système 201 communiquent les uns avec les autres par l'intermédiaire de la base de données 164. La saisie d'informations dans une table de la base de données 164 par un organe peut déclencher une action d'un autre organe. Ce mécanisme de communication ne constitue qu'un exemple de procédé pour transmettre des informations entre des organes et faire progresser le travail. Selon une autre possibilité, on peut recourir à tout mécanisme de communication et à tout déroulement de travail permettant le fonctionnement des systèmes et procédés décrits ici.

La Figure 4 est un diagramme 400 de flux de données dans un exemple de module de demande 304 du système 201 (représenté sur la figure 3), ayant pour fonction de recevoir et de traiter la demande 204 liée à l'Apprentissage Automatique. Par exemple, l'utilisateur 202 peut présenter une demande 204 sollicitant qu'une exploration de modèles soit effectuée à l'aide de la tâche de classification. Cette exploration de l'espace d'un modèle constitue une tâche exigeant énormément de calculs, laquelle peut être divisée en sous-tâches et exécutée en recourant à un ensemble de ressources informatiques dans le but de bénéficier de l'utilisation de multiples ressources informatiques.

Par ailleurs, dans l'exemple de forme de réalisation, le module de demande 304 stocke des informations de demande 404 concernant la demande 204. Dans certaines formes de réalisation, les informations de demande 404 est stockées dans la base de

données 164 (représentée sur la Figure 1). Selon une autre possibilité, les informations de demande 404 peuvent être stockées de n'importe quelle autre manière, notamment, sans s'y limiter, dans le périphérique de mémoire 150 (représenté sur la figure 1), ou de n'importe quelle manière permettant le fonctionnement des systèmes et procédés décrits ici. Les informations de demande 404 peuvent comprendre, sans s'y limiter, des informations de définition de problèmes, des noms de modèles, des paramètres de modèles, des données d'entrée, un nombre de colonnes d'étiquettes dans le fichier de données indiquant la "réalité concrète" pour l'entraînement/l'optimisation, le type de tâche, p.ex. une classification ou un regroupement ou une régression ou un ajustement de règles, des critères de performances, un procédé d'optimisation, p.ex. des points d'un maillage pour une recherche par maillage ou des limites de recherche pour une optimisation évolutive, des exigences et préférences de calculs, de confidentialité et de cryptage de données. Selon une autre possibilité, les informations de demande 404 peuvent comprendre toute information permettant le fonctionnement des systèmes et procédés décrits ici.

En outre, dans l'exemple de forme de réalisation, le module de demande 304 crée une tâche 402. Dans certaines formes de réalisation, la tâche 402 est représentée par une rangée unique dans la base de données 164. Selon une autre possibilité, la demande 204 peut solliciter de multiples tâches 402 pour satisfaire la demande 204. Par exemple, quand l'utilisateur 202 saisit la demande 204 sollicitant la réalisation d'une exploration de modèles au moyen d'une classification, le module de demande 304 saisit, dans la table des tâches 402, une rangée indiquant une nouvelle tâche de classification et relie la tâche 402 à ses propres informations de

demande 404. Le module de planification/optimisation 310 de tâches vérifie périodiquement dans la table des tâches 402 la présence de nouvelles tâches non traitées. Une fois que le module de planification/optimisation 310 aura constaté la tâche 402, il agira pour traiter encore la tâche comme décrit ci-après.

De plus, dans l'exemple de forme de réalisation, le module de demande 304 reçoit des résultats 406 de demandes une fois qu'une tâche a été complètement traitée. Dans certaines formes de réalisation, les résultats 406 de demandes sont stockés dans la base de données 164. En fonctionnement, le module de demande 304 doit recevoir les résultats 406 de demandes en constatant qu'un résultat 406 de demande nouvellement renvoyé a été inscrit dans la base de données 164. Ce traitement de résultat est une étape ultérieure dans le fonctionnement global du système 201 (représenté sur la Figure 3) et est expliqué plus en détail ci-après.

La Figure 5 est un diagramme 500 de flux de données de l'exemple de module de planification/optimisation 310 de tâches du système 201 (représenté sur la Figure 3), ayant pour fonction de préparer des tâches 402 à exécuter. Le module de planification/optimisation 310 de tâches analyse la tâche 402 et les informations de demande 404 et sélectionne un ou plusieurs modèles dans la bibliothèque 308. D'après les informations de demande 404, le module de planification/optimisation 310 de tâches crée un ou plusieurs modèles 502 de tâches. Par exemple, quand le module de planification/optimisation 310 de tâches constate une tâche demandant une classification, le module de planification/optimisation 310 de tâches examine les informations de demande 404 pour savoir si un type de classification particulier tel qu'une Machine à Vecteur Support ("SVM") ou un Réseau Neuronal Artificiel ("ANN") a été spécifié par l'utilisateur 202. Si

aucun modèle spécifique n'a été spécifié, le module de planification/optimisation 310 de tâches crée alors, dans le modèle 502 de tâche, une rangée pour chaque type de classification qui convient et est disponible dans la bibliothèque 308.

Par ailleurs, dans l'exemple de forme de réalisation, un modèle individuel 504 de tâche est créé par le module de planification/optimisation 310 pour chaque modèle 502 de tâche. En fonctionnement, le modèle individuel 504 de tâche sert à limiter encore la manière dont le modèle 502 de tâche peut être exécuté et le lieu où il peut l'être. Le module de planification/optimisation 310 limite le modèle individuel de tâche d'après les informations de demande 404 et les restrictions du modèle, notamment, sans s'y limiter, les ressources informatiques préférées indiquées par l'utilisateur 202 (représentés sur la Figure 3), et la plate-forme requise spécifiée par le modèle particulier sélectionné dans la bibliothèque 308. Par exemple, quand le module de planification/optimisation 310 de tâches crée une tâche 402 à classer à l'aide d'un SVM, le module de planification/optimisation 310 de tâches consulte le modèle de SVM dans la bibliothèque 308 et la demande dans les informations de demande 404. Si le modèle de SVM dans la bibliothèque a une restriction de calcul telle que le fait de n'avoir qu'une version compilée de modèle pour Linux 32 bits, le modèle individuel 504 de tâche sera limité à la seule utilisation d'hôtes de Linux 32 bits. Selon une autre possibilité, si les informations de demande 404 spécifient l'utilisation des seules ressources de calcul internes 212, le modèle individuel 504 de tâche sera ainsi restreint. Dans certaines formes de réalisation, le modèle individuel 504 de tâche peut consister en une ou plusieurs tâches d'exécution définie(s) par des informations sur l'espace de recherche faisant partie des informations de demande 404. Les

tâches d'exécution peuvent être partagées et exécutées par une pluralité de ressources informatiques dans le module d'exécution 312, comme expliqué ci-après.

En fonctionnement, dans l'exemple de forme de réalisation, le module de planification/optimisation 310 de tâches vérifie périodiquement les tâches 402 pour déceler des saisies non traitées. En détectant une nouvelle tâche 402, le module de planification/optimisation 310 de tâches analyse les informations de demande 404 et sélectionne plusieurs modèles dans la bibliothèque 308. Le module de planification/optimisation 310 de tâches crée alors, dans les modèles 502 de tâches, une nouvelle rangée pour chaque modèle nécessaire au traitement de la tâche 402. En outre, le module de planification/optimisation 310 de tâches crée un exemple 504 de modèle de tâche pour chaque modèle 502 de tâche, limitant encore la manière dont est traité le modèle 502 de tâche. Chacun de ces modèles individuels 504 de tâches est créé sous la forme de rangées individuelles dans la base de données 164. Ces modèles individuels 504 de tâches seront traités par le module d'exécution 312 et le module 314 de ressources, comme expliqué ci-après.

Par ailleurs, dans certaines formes de réalisation, le module de planification/optimisation 310 de tâches peut exécuter une série de tâches itératives qui nécessite la présentation 506 de modèles 502 de tâches et de modèles individuels 504 de tâches supplémentaires après la réception de résultats d'un modèle individuel antérieur 504 de tâche. Dans certaines formes de réalisation, notamment lorsque le procédé d'optimisation est spécifié sous la forme d'une recherche en maillage ou autre organisation combinatoire, la présentation et le traitement d'un unique ensemble de modèles 502 de tâches et de modèles individuels 504 de tâches suffiront pour satisfaire la tâche 402.

Dans d'autres formes de réalisation, où sont spécifiés d'autres procédés d'optimisation tels que, sans s'y limiter, une recherche heuristique, des algorithmes évolutifs et une optimisation stochastique, certaines tâches 402 peuvent nécessiter un traitement d'un premier ensemble de résultats après exécution, suivi d'une présentation d'ensembles supplémentaires de modèles 502 de tâches et de modèles individuels 504 de tâches. Ce post-traitement de résultat et cette présentation de modèles supplémentaires 502 de tâches peuvent avoir lieu un certain nombre de fois ou jusqu'à ce qu'une condition de satisfaction soit remplie. En fonction du nombre de critères de performances spécifiés dans les informations de demande 404, l'optimisation peut être une optimisation à objectif unique ou à objectifs multiples.

La Figure 6 est un diagramme 600 de flux de données d'exemples de module d'exécution 312 et de module de ressources 314 du système 201 (représentés sur la Figure 3), ayant pour fonction d'attribuer des tâches à des ressources informatiques 206 et de transmettre des tâches à exécuter. La disponibilité de ressources informatiques est maintenue par le module de ressources 314 à l'aide d'une table de ressources individuelles 602. Chaque rangée de la table de ressources individuelles 602 est corrélée avec une ou plusieurs ressources informatiques 206 pouvant servir à exécuter des modèles individuels 504 de tâches. Dans certaines formes de réalisation, chaque ressource individuelle 602 est une rangée stockée dans la base de données 164 (représentée sur la figure 1). Au sens de la présente description, l'expression "ressource individuelle" peut désigner soit une table d'une base de données servant à réaliser un suivi des ressources informatiques 206, soit les ressources informatiques individuelles dont la table sert à réaliser le suivi.

Par ailleurs, dans l'exemple de forme de réalisation, le module de ressources 314 sélectionne un sous-ensemble de ressources informatiques 206 et attribue ces ressources individuelles 602 à chaque modèle individuel 504 de ressources d'après, sans s'y limiter, les restrictions informatiques associées au modèle individuel 504 de tâche, les informations 404 de demande et la disponibilité de ressources informatiques. En fonctionnement, quand une ressource individuelle 602 est attribuée au modèle individuel 504 de tâche, le module de ressources 314 crée, dans la base de données 164, une rangée servant à réaliser un suivi de l'attribution de la ressource individuelle 602 au modèle individuel 504 de tâche. Par exemple, le module de ressources 314 constate qu'un nouvel exemple 504 de modèle de tâche nécessite un ensemble de ressources informatiques. Le module de ressources 314 examine les restrictions de ressources informatiques dans le modèle individuel 504 de tâche et constate qu'il existe une restriction imposant l'utilisation exclusive de nœuds Linux, mais que n'importe quels nœuds Linux publics ou privés sont acceptables. Le module de ressources 314 recherche alors la ressource individuelle 602 pour trouver un ensemble approprié de ressources informatiques Linux convenant pour le modèle individuel 504 de tâche. L'ensemble de ressources informatiques est ensuite attribué au modèle individuel 504 de tâche à exécuter.

En outre, dans certaines formes de réalisation, le système 201 peut gérer, dans la base de données 164, une deuxième table (non représentée) qui actualise une liste de toutes les ressources disponibles à un moment pour le système 201, de façon que chaque rangée dans la ressource individuelle 602 soit corrélée avec une rangée de la deuxième table. Cette deuxième table peut comprendre des ressources informatiques individuelles fournies à un moment

par le nuage publique 208 ou le nuage privé 210, et peut également comprendre des ressources de calcul internes individuelles 212. Dans certaines formes de réalisation, le système peut aussi gérer, dans la base de données 164, une troisième table (non représentée) qui actualise une liste de tous les prestataires de ressources informatiques, de façon que chaque ressource informatique individuelle figurant sur la liste de la deuxième table soit corrélée avec un prestataire figurant sur la liste de la troisième table.

De plus, dans certaines formes de réalisation, le module de ressources 314 considère la demande 204 et/ou les informations de demande 404 au moment de décider de la manière d'attribuer des ressources. La demande 204 peut comprendre des restrictions de coût, de délais et/ou de sécurité concernant l'utilisation de ressources informatiques, notamment, sans s'y limiter, l'utilisation de ressources informatiques disponibles gratuitement, l'utilisation de ressources informatiques à tarif limité par nœud, l'utilisation de ressources informatiques à un coût d'un montant maximal fixe, les contraintes de délais, l'utilisation exclusive de ressources informatiques privées et l'utilisation de ressources informatiques sécurisées. Si, par exemple, l'utilisateur 202 a spécifié dans la demande une limitation quant à une utilisation exclusive d'hôtes "sécurisés", ou à une dépense non supérieure à une somme donnée pour exécuter la demande, le module de ressources 314 doit factoriser ces limitations supplémentaires dans le processus de sélection pendant l'attribution de ressources. Selon une autre possibilité, le module de planification/optimisation 310 de tâches peut avoir pris en considération la demande 204 et/ou les informations 404 de demande lors de l'ajout de restrictions dans le modèle individuel 504 de tâche.

Par ailleurs, dans l'exemple de forme de réalisation, le module de ressources parallélise l'exécution du modèle individuel 504 de tâche à l'aide des multiples ressources individuelles 602 pour satisfaire l'exécution du modèle individuel 504 de tâche. Au sens de la présente description, la "parallélisation" est le processus consistant à diviser une grande tâche unique en éléments plus petits et à exécuter chaque petit élément individuellement à l'aide d'une pluralité de ressources informatiques. Dans certaines formes de réalisation, le modèle individuel 504 de tâche peut être réparti sur les paramètres du modèle, c'est-à-dire que chaque ressource informatique doit obtenir la totalité des données d'entraînement mais seulement une partie des paramètres du modèle. Selon une autre possibilité, n'importe quel autre procédé de parallélisation du modèle individuel 504 de tâche permettant au système 201 de fonctionner comme décrit ici peut être utilisé, notamment, sans s'y limiter, une répartition sur les données d'entraînement, c'est-à-dire que chaque ressource informatique doit obtenir tous les paramètres du modèle, mais seulement une partie des données d'entraînement, ou une répartition des données d'entraînement ainsi que des paramètres du modèle. En outre, dans certains scenarii, le modèle individuel 504 de tâche peut être parallélisé sur des ressources informatiques hétérogènes, c'est-à-dire que l'ensemble de ressources individuelles 602 affectées au modèle individuel 504 de tâche est hétérogène. De plus, dans certains scenarii, le modèle individuel 504 de tâche peut être parallélisé sur de multiples sources de ressources informatiques, une partie du modèle individuel 504 de tâche étant par exemple exécutée par le nuage public 208 et une autre partie étant exécutée par le nuage privé 210 ou la ressource de calcul interne 212.

En fonctionnement, dans l'exemple de forme de réalisation, le module de ressources 314 vérifie l'existence de nouveaux modèles individuels 504 de tâches. Quand le module de ressources 314 détecte de nouveaux modèles individuels 504 de tâches auxquels des ressources n'ont pas encore été attribuées, le module de ressources 314 consulte la ressource individuelle 602 pour trouver des ressources informatiques appropriées et affecte des ressources individuelles appropriées 602, inutilisées à ce moment, aux modèles individuels 504 de tâches. Le module de ressources 314 recherche des ressources individuelles 602 qui satisfassent, sans s'y limiter, les exigences de plate-forme du modèle telles que le système d'exploitation et la taille des processeurs et de la mémoire, et le nombre, minimal à maximal, de nœuds spécifiés par le modèle. Dans certaines formes de réalisation, si au moins le nombre minimal de nœuds requis n'est pas disponible, le modèle individuel 504 de tâche reste alors non planifié et sera à nouveau examiné ultérieurement. Dans l'exemple de forme de réalisation, le module de ressources 314 décidera s'il convient de demander davantage de ressources en s'appuyant sur des facteurs tels que, sans s'y limiter, le nombre de demandes alors en file d'attente, les types de modèles demandés, la qualité requise de la solution définitive, les contraintes de coûts et de délais, la qualité obtenues à ce moment par rapport aux contraintes de coût et de délais et une estimation des ressources nécessaires pour exécuter chaque modèle. S'il faut s'attendre à un besoin de ressources supplémentaires, le module de ressources 314 peut demander plus de ressources informatiques 206 aux nuages publics 208 ou au nuage privé 210 pour introduire davantage de ressources individuelles 602 dans la série de ressources disponible. Par exemple, et sans s'y limiter, si le module de ressources 314 évalue qu'il peut répondre aux exigences

de délais imposées par la demande 204 pour trouver une solution d'une grande qualité reposant sur le nombre de ressources individuelles 602 alors impliquées, il n'est pas nécessaire d'impliquer des ressources individuelles supplémentaires 602, car il risque d'en résulter un surcoût. Dans certaines formes de réalisation, le module de ressources 314 utilise une table de correspondance qui contient les mesures de performances évoquées plus haut, créées d'après un historique des performances à l'occasion de problèmes antérieurs similaires. Dans certaines formes de réalisation, le module de ressources 314 peut avoir un nombre maximal de ressources utilisables en même temps, si bien que le module de ressources 314 peut fournir jusqu'à ce nombre maximal. Une fois que des ressources individuelles 602 auront été attribuées au modèle individuel 504 de tâche, le module d'exécution 312 continuera à traiter le modèle individuel 504 de tâche à l'aide de la ressource individuelle 602, comme décrit plus loin.

Par ailleurs, dans l'exemple de forme de réalisation, le module d'exécution 312 utilise des modules d'API 313 pour transmettre les modèles individuels 504 de tâches aux ressources informatiques 206. Le module d'exécution 312 a pour fonction de communiquer avec les différentes ressources informatiques 206 pour assurer des fonctions telles que, sans s'y limiter, la fourniture de nouvelles ressources informatiques, la transmission de modèles individuels 504 de tâches aux ressources informatiques 206 en vue de leur exécution, la réception de résultats à la suite de l'exécution et l'abandon de ressources informatiques devenues inutiles.

En outre, dans l'exemple de forme de réalisation, le module d'exécution 312 présente le modèle individuel 504 de tâche à la ressource individuelle 602 en vue de son exécution. La ressource individuelle 602 est une ou plusieurs ressources informatiques 206

émanant de sources dont des nuages publics 208, des nuages privés 210 et/ou des ressources de calcul internes 212. Pour faciliter la communication avec chaque source de ressource informatique, le module d'exécution 312 utilise des modules d'API 313. A chaque source de ressources informatiques 206 est associée une API. Une API est une classe de communications créée sous la forme d'un protocole pour communiquer avec un programme particulier, p.ex., dans le cas d'un fournisseur de nuage, l'API du fournisseur de nuage crée une méthode de communication avec le fournisseur de nuage et les ressources sur nuage, afin d'assurer des fonctions telles que, sans limitation, la fourniture de nouvelles ressources informatiques, la communication avec les ressources informatiques fournies à un moment et l'abandon de ressources informatiques. Chaque module d'API 313 communique avec une seule source de ressources informatiques telles que, sans s'y limiter, Amazon EC2®, ou une grappe interne de serveurs privés à grande disponibilité. Un module d'API 313 pour une source de ressources informatiques associée doit être inclus dans le système 201 afin que le module de ressources 314 fournisse et affecte des modèles individuels 504 de tâches à cette source de ressources informatiques et afin que le module d'exécution 312 exécute les modèles individuels 504 de tâches à l'aide de cette source de ressources informatiques. Dans certaines formes de réalisation, le modèle individuel 504 de tâche aura de multiples ressources individuelles 602 attribuées par différentes sources et impliquera de multiples modules d'API pour communiquer avec chaque ressource informatique respective.

En fonctionnement, dans l'exemple de forme de réalisation, le module d'exécution 312 vérifie périodiquement les modèles individuels 504 de tâches à la recherche de modèles individuels 504

de tâches auxquels sont affectées de ressources informatiques et qui sont préparés pour être exécutés. Le module d'exécution 312 examine les ressources individuelles 602 pour déterminer quelle source de ressources informatiques a été affectée au modèle individuel 504 de tâche puis, à l'aide de son module d'API correspondant, transmet à la ressource informatique particulière une sous-tâche associée à l'exécution du modèle individuel 504 de tâche. Si, par exemple, au modèle individuel 504 de tâche ont été attribués 10 nœuds Linux, 8 d'une grappe de calcul Linux interne et 2 d'un nuage public, le module d'exécution fait alors intervenir le module d'API associé à la grappe de calcul Linux interne pour exécuter les 8 sous-tâches sur la grappe de calcul Linux interne, et fait aussi intervenir le module d'API associé au prestataire de nuage public pour exécuter les 2 sous-tâches sur le nuage public. Dans certaines formes de réalisation, le module d'exécution 312 présente le modèle individuel 504 de tâche tout entier à une seule ressource individuelle 602.

Par ailleurs, dans l'exemple de forme de réalisation, le module d'exécution 312 teste périodiquement les ressources individuelles 602 pour vérifier l'achèvement des sous-tâches attribuées correspondant au modèle individuel 504 de tâche. Le module d'exécution regroupe des résultats de multiples sous-tâches et renvoie les résultats regroupés au module de planification/optimisation 310 de tâches. Le module d'exécution 312 reçoit les données de résultats 606 directement de la ressource individuelle 602, c'est-à-dire du serveur individuel qui a exécuté une partie du modèle individuel 504 de tâche. Selon une autre possibilité, le module d'exécution 312 reçoit les données de résultats 606 d'un gestionnaire 603 de stockage ou d'un stockage partagé 604, décrit plus loin. Si, par exemple, le modèle individuel

504 de tâche a été attribué à 10 ressources individuelles 602, le module d'exécution 312 distribue alors des sous-tâches à chacune des 10 ressources individuelles 602, puis les teste jusqu'à leur achèvement. Une fois que les données de résultats 606 de la totalité des 10 ressources individuelles 602 sont recueillies, elles sont regroupées et renvoyées au module de planification/exécution 310 de tâches. Dans certains scenarii, le module de planification/exécution 310 de tâches, suivant le type de tâche, analyse le résultat globalisé du modèle individuel 504 de tâche et renvoie le résultat 214 (représenté sur la Figure 3). Dans d'autres scenarii, le module de planification/optimisation 310 de tâches peut analyser le résultat globalisé du modèle individuel 504 de tâche, mais exécute ensuite encore un ou plusieurs modèles individuels 504 de tâches avant de renvoyer un résultat final 214. Le résultat du premier modèle individuel 504 de tâche peut être utilisé dans le ou les modèles individuels de tâches ultérieurs 504. Dans l'exemple de forme de réalisation, le module de planification/optimisation de tâche renvoie le résultat globalisé au module de demande 304 (représenté sur la Figure 3). Selon une autre possibilité, le module de planification/optimisation 310 de tâches renvoie le modèle individuel 504 de tâche à l'utilisateur 202 (représenté sur la Figure 3).

En outre, dans certaines formes de réalisation, le module d'exécution 312 peut surveiller l'état des ressources individuelles 602 pour détecter tout échec associé à la sous-tâche attribuée affectant le modèle individuel de tâche auquel elle a été attribuée. Par exemple, et sans s'y limiter, une erreur de temps de déroulement pendant l'exécution, ou une défaillance du système d'exploitation de la ressource individuelle 602 elle-même. En constatant une panne, le module d'exécution 312 peut recommencer la sous-tâche

liée au modèle individuel 504 de tâche dans la ressource individuelle 602 d'origine ou peut réattribuer la sous-tâche à une autre ressource individuelle possible 602. Dans d'autres formes de réalisation, le module d'exécution 312 peut être conçu sous la forme d'une seconde couche de tolérance de défaut, ce qui permet à un prestataire de services sur nuage de fournir la première couche de tolérance de défaut par l'intermédiaire de ses propres mécanismes exclusifs, et ne mettant en œuvre les mécanismes décrits plus haut que si le module d'exécution 312 détecte une défaillance du mécanisme de tolérance de défaut du prestataire de services sur nuage.

En outre, dans l'exemple de forme de réalisation, le module de ressources 314 assure la tâche d'approvisionnement et de mobilisation de ressources informatiques. En fonctionnement, la demande 204 peut demander au système 201 d'utiliser davantage de ressources informatiques qu'il n'en est alors fourni et disponible. Le module de ressources 314 utilise des modules d'API 313 pour fournir de nouveaux nœuds sur demande, comme décrit plus haut. Le module de ressources 314 abandonne également des ressources informatiques lorsqu'elles ne sont plus nécessaires. Dans certaines formes de réalisation, le module de ressources 314 peut abandonner des ressources parmi les ressources individuelles 602, sur demande ou contre paiement. Par exemple, et sans s'y limiter, le module de ressources 314 peut abandonner un nœud pendant une période de la journée où le pic de demande accroît le coût de la ressource individuelle 602 en fonction de contraintes de délais et de contraintes de coût de la demande 204, et peut réacquérir la ressource individuelle 602 quand a pris fin la période de demande de pointe.

De plus, dans certaines formes de réalisation, le système 201 comporte un gestionnaire de stockage 603 et un stockage partagé 604. Le stockage partagé 604 peut être, sans s'y limiter, un stockage privé ou un stockage sur nuage. Le stockage partagé 604 est accessible pour les ressources informatiques 206 de manière à permettre aux ressources informatiques 206 de stocker les données 606 associées à l'exécution de modèles individuels 504 de tâches. Le stockage partagé 604 peut servir à stocker des données 606 telles que, sans s'y limiter, des informations sur le modèle, des données d'entrée du modèle et des résultats d'exécution. Le stockage partagé 604 peut également être accessible au gestionnaire 603 de stockage, lequel peut intervenir pour refaire passer dans le système 201 des données 606 concernant l'exécution. Le gestionnaire 603 de stockage peut également affecter le stockage partagé 604 à des ressources informatiques 206 et peut affecter le stockage partagé 604 à la demande du module d'exécution 312 ou du module de planification/optimisation 310 de tâches.

La Figure 7 est un schéma de principe d'un exemple de procédé 700 d'identification et de création automatiques de modèle en fournissant des ressources informatiques hétérogènes 206 pour Apprentissage Automatique à l'aide du système 201 (représenté sur la Figure 3). Le procédé 700 est mis en œuvre par au moins un système informatique 120 comprenant au moins un processeur 152 (représenté sur la Figure 1) et au moins un périphérique de mémoire 150 (représenté sur la Figure 1) couplé au(x) processeur(s) 152. Une demande 204 d'exécution est reçue 702.

Par ailleurs, dans l'exemple de forme de réalisation, un ou plusieurs algorithmes est/sont sélectionné(s) 704 dans la bibliothèque 308. Chaque algorithme de la bibliothèque 308 comprend soit un code source soit un code exécutable par machine.

La sélection 704 d'un sous-ensemble d'algorithmes repose au moins partiellement sur une demande d'exécution 204. Une ou plusieurs tâches d'exécution, p.ex. les modèles individuels 504 de tâches, sont identifiées 706 pour exécution. La/chacune des tâches d'exécution comprend au moins un algorithme de la bibliothèque 308.

En outre, dans l'exemple de forme de réalisation, un sous-ensemble de ressources informatiques est déterminé 708 parmi une pluralité de ressources informatiques 206. La pluralité de ressources informatiques 206 comprend au moins une ressource de calcul interne, à savoir le nuage privé 210 et la ressource de calcul interne 212 ou au moins une ressource de calcul d'un tiers, à savoir le nuage public 208 et une pluralité de ressources de calcul d'un tiers, à savoir le nuage public 208. Le système informatique 120 transmet 710 la/au moins une des tâches d'exécution à au moins une ressource informatique 206 du sous-ensemble de ressources informatiques et reçoit 712 un résultat 214 d'exécution.

La Figure 8 est un schéma de principe d'un autre exemple de procédé 800 d'identification et de création automatiques de modèle en fournissant des ressources informatiques hétérogènes 206 pour Apprentissage Automatique à l'aide du système 201 (représenté sur la Figure 3). Le procédé 800 est mis en œuvre par au moins un système informatique 120 comprenant au moins un processeur 152 (représenté sur la Figure 1) et au moins un périphérique de mémoire 150 (représenté sur la Figure 1) couplé au(x) processeur(s) 152. Une demande 204 de tâche comprenant une ou plusieurs tâches individuelles est identifiée 802. Pour la demande 204 de tâche, des besoins d'une ou de plusieurs ressources informatiques sont identifiés 804.

Par ailleurs, dans l'exemple de forme de réalisation, un ensemble de ressources informatiques d'exécution est déterminé 806 parmi une série de ressources informatiques au moins partiellement à partir de la/des ressource(s) informatique(s) nécessaire(s). Les ressources informatiques 206 comprennent au moins une ressource de calcul interne, à savoir le nuage privé 210 et la ressource interne 212 ou au moins une ressource de calcul externe, à savoir le nuage public 208 et une pluralité de ressources de calcul externes, à savoir le nuage public 208. Chaque ressource informatique de la série de ressources informatiques définit une API correspondante qui facilite la communication entre le système 201 (représenté sur la Figure 3) et la ressource informatique. Parmi l'ensemble de ressources informatiques d'exécution, une première ressource informatique est attribuée 808 à une première tâche individuelle constituée par la/les tâches individuelles, p.ex. les modèles individuels 504 de tâches (représentés sur les figures 5 et 6).

En outre, dans l'exemple de forme de réalisation, une pluralité de modules d'interfaçage 313 est identifiée 810. Chaque module d'interfaçage est conçu pour faciliter la communication avec une ou plusieurs ressources informatiques 206 à l'aide de l'API correspondante. Un module d'interfaçage est choisi 812 parmi une pluralité de modules d'interfaçage 313 au moins partiellement sur la base de la facilitation de la communication avec la première ressource informatique. Le système informatique 120 transmet 814 la première tâche individuelle à exécuter à la première ressource informatique à l'aide du premier module d'interfaçage et reçoit 816 un résultat d'exécution. Au sens de la présente invention, l'expression "modules d'interfaçage" se rapporte à des modules d'API.

Les figures 9 à 11 représentent un schéma en trois parties d'un exemple de structure de 900 de tables de base de données pour le système 201 (représenté sur la Figure 2). Chaque élément des figures 9 à 11 représente une table séparée présente dans la base de données 164, et le contenu de chaque élément indique le nom de la table et la structure de la table, dont des noms de champs et des types de données. Les interconnexions entre éléments indiquent au moins une relation entre les deux tables, telle que, sans s'y limiter, un champ commun. En fonctionnement, chaque table est utilisée par un ou plusieurs des organes du système 201 pour suivre et traiter les diverses étapes de l'exécution de la demande 204 (représentée sur la Figure 2) de tâche. Les relations entre les tables et les organes du système 201 sont décrites ci-après.

La Figure 9 est un schéma de principe d'une première partie d'un exemple de structure 900 de tables de base de données pour le système 201 (représenté sur l Figure 3), représentant les principales tables utilisées par le module de demande (représenté sur la Figure 3). Request 916 est une table de niveau haut contenant des informations sur les demandes 204 (représentées sur la Figure 3). Des informations détaillées pour la demande 204 sont stockées dans Request Info 913 et comprennent, sans s'y limiter, des informations sur des critères de performances, des préférences et des limites pour des ressources informatiques, des noms de modèles, des fichiers d'entrée et de sortie, des fichiers à format conteneur, des fichiers de modèles et des informations sur la confidentialité et le cryptage de données. Job 914 est une table contenant des informations relatives à la demande de traitement 204. Job 914 établit un lien entre les informations fournies par Tasks 908 et Request 916 et sert à lancer un traitement plus poussé par le système 201. Une seule demande 204 peut générer une ou plusieurs entrées dans Job 914. Models 903

est une table qui actualise une bibliothèque de modèles d'apprentissage automatique disponibles pour le système 201. Tasks 908 est une table qui actualise des types de tâches pour les divers modèles que le système 201 peut prendre en charge. Task Models 901 est une table qui associe Models 903 avec ses types de tâches respectifs.

En fonctionnement, dans l'exemple de forme de réalisation, l'utilisateur 202 (représenté sur la figure 2) présente une demande 204 au système 201. De nouvelles demandes 302 sont reçues et traitées par le module de demandes 304 (représenté sur la Figure 3). A la réception de la demande 204, le module de demandes 304 crée une nouvelle rangée dans Request 916 et une nouvelle rangée dans Request Info 913. Les informations associées à la demande 204 sont stockées dans Request Info 913. La demande 204 peut indiquer de quel Model 903 l'utilisateur désire l'utilisation. Selon une autre possibilité, l'utilisateur 202 peut indiquer un type de tâche, grâce à quoi le système 201 exécute un ou plusieurs Models 903 associés à ce type de tâche. Le module de demandes 304 crée alors une nouvelle rangée dans Job 914. La création des entrées dans Job 914 sert de voie de communication avec le module de planification/optimisation 310 de tâches (représenté sur la Figure 3). Quand le module de planification/optimisation 310 de tâches détecte de nouvelles entrées dans la table Job 914, le module de planification/optimisation 310 de tâches poursuit le traitement.

La Figure 10 est un schéma de principe d'une deuxième partie d'un exemple de structure 900 de base de données pour le système 201 (représenté sur la Figure 3), représentant les principales tables utilisées par le module de planification/optimisation de tâche (représenté sur la Figure 3). Une table Job Model 915 contient des informations sur les tâches qu'il

faut exécuter pour traiter la demande 204 (représentée sur la Figure 3). Chaque entrée dans Job Model 915 est associée à une seule entrée dans la table Job 914 ainsi qu'à une seule entrée dans Models 903. Une table Job Model Instance 911 contient des informations relatives à des entrées dans Job Model 915, définissant plus précisément les restrictions de Job Model 915 reposant, par exemple et sans s'y limiter, sur les limitations de ressources informatiques liées au modèle particulier et sur les limitations de ressources informatiques liées à la demande 204 (représentée sur la Figure 3). Dans l'exemple de forme de réalisation, Job Model Instance 911 contient une seule rangée pour chaque rangée de Job Model 915. Selon une autre possibilité, une seule rangée de Job Model 915 peut se traduire par de multiples Job Model Instances 911.

En fonctionnement, dans l'exemple de forme de réalisation, le module de planification/optimisation 310 de tâches (représenté sur la Figure 3) détecte l'apparition d'une nouvelle rangée, non traitée, dans Job 914. Le module de planification/optimisation 310 de tâches sélectionne *n* modèles 903 et crée *n* nouvelles rangées dans la table Job Models 915. Chacune de ces nouvelles rangées dans Job Model 915 représente une sous-tâche, affiliée à un modèle individuel parmi Models 903, qui doit être exécutée pour traiter la demande 204. Ensuite, le module de planification/optimisation 310 de tâches crée, dans la table Job Model Instance 911, *n* rangées chacune en corrélation avec l'une des *n* nouvelles rangées de Job Model 915. Le module de planification/optimisation 310 de tâches prend en considération et formule des restrictions pour chaque Job Model 915 lors de la création de Job Model Instance 911. La création des entrées de Job Model Instance 911 sert de voie de communication avec le module de ressources 314 (représenté sur la Figure 3) et le module d'exécution 312 (représenté sur la Figure 3).

Quand le module de ressources 314 détecte de nouvelles entrées dans Job Model Instance 911, le module de ressources 314 poursuit le traitement.

La Figure 11 est un schéma de principe représentant une troisième partie de l'exemple de structure 900 de base de données pour le système 201 (représenté sur la figure 3), représentant les principales tables utilisées par le module d'exécution (représenté sur la Figure 3) et le module de ressources (représenté sur la Figure 3). Les informations sur les ressources informatiques 206 (représentées sur la Figure 3) sont actualisées par trois tables, Compute Resources 912, Resource Instance 920 et Instance Resource 919. Compute Resources 912 contient des informations de niveau haut sur des sources de ressources informatiques. Resource Instance 920 fournit des détails concernant chaque ressource informatique individuelle fournie au même instant au système 201 ou autrement disponible pour être utilisée par le système. Instance Resource 919 suit l'affectation de Resource Instances 920. Dans l'exemple de forme de réalisation, chaque Resource Instance 920 a une rangée correspondante dans la table Instance Resource 919 chaque fois que le Resource Instance 920 est attribuée pour exécuter une tâche. Selon une autre possibilité, une rangée de la table Instance Resource 919 est créée quand la Resource Instance 920 commence à exécuter une tâche attribuée.

En fonctionnement, dans l'exemple de forme de réalisation, chaque prestataire de services sur nuage avec lequel le système 201 est conçu pour coopérer a une rangée dans Compute Resources 912. Chaque nuage privé ou ressource interne peut aussi avoir des rangées dans Compute Resources 912. Par exemple, et sans s'y limiter, la table Compute Resources 912 peut avoir une entrée pour Amazon EC2®, Rackspace®, Terremark®, un nuage interne privé et

les ressources de calcul internes. Chaque rangée représente une source de services informatiques avec lesquels le système 201 est conçu pour coopérer. Resource Instance 920 a une rangée pour chaque dispositif informatique individuel fourni au même instant au système 201 ou autrement disponible pour être utilisé par le système. Chaque Resource Instance 920 aura une ressource informatique "parente" 912 associée à celle-ci, en fonction du prestataire de services sur nuage ou autre source d'où provient la Compute Resource 912. Par exemple, et sans s'y limiter, quand le système 201 fournira des serveurs virtuels d'Amazon EC2®, le système 201 créera 10 entrées dans Resource Instance 920, chacun d'elles correspondant à un seul serveur virtuel d'Amazon EC2®. Dans l'exemple de forme de réalisation, pour les ressources sur nuage, ces rangées sont créées et supprimées à mesure que le système 201 fournit et abandonne des serveurs virtuels des Prestataires de Services sur Nuage. Selon une autre possibilité, des rangées peuvent rester dans la table malgré l'abandon du serveur virtuel associé à une rangée.

Par ailleurs, en fonctionnement, dans l'exemple de forme de réalisation, les Resource Instances 920 sont attribuées pour exécuter une tâche, c'est-à-dire qu'elles sont attribuées pour exécuter des Job Model Instances 911. La table Instance Resource 919 suit l'attribution de Resource Instances 920 à des modèles individuels de tâches 911. Quand un nouveau Job Model Instance 911 est ajouté, le module de ressources 314 attribue une Resource instance 920 au Job Model Instance 911 d'après les informations présentes dans Job Model Instance 911. Selon une autre possibilité, le module d'exécution 312 ou le module de ressources 314 crée ou met à jour une rangée d'Instance Resource 919 associée à la Resource instance 920.

Par ailleurs, dans l'exemple de forme de réalisation, le stockage partagé 604 (représenté sur la Figure 7) peut être attribué pour être utilisé par Resource Instances 920. Une table Storage Resources 906 contient des informations de niveau haut sur les fournisseurs de ressources de stockage disponibles pour le système 201. Une table Storage Instances 907 contient des informations sur des exemples individuels de stockages qui ont été fournis par le système 201 ou ont été attribués pour être utilisés par le système. En fonctionnement, l'exécution d'un Job Model Instance 911 peut nécessiter l'utilisation d'un Storage Instance 907. Le gestionnaire 603 de stockage (représenté sur la figure 7) attribue un Storage Instance 907, à savoir un stockage partagé 604 (représenté sur la Figure 7), au Job Model Instance 911 pour qu'il serve pendant l'exécution.

Les systèmes et procédés décrits plus haut donnent des moyens pour fournir automatiquement des ressources informatiques faisant partie d'un ensemble hétérogène de ressources informatiques à des fins d'Apprentissage Automatique. Les formes de réalisation décrites ici reçoivent une demande d'un utilisateur, sélectionnent, dans une base de données de modèles, un sous-ensemble de modèles qui satisfont les exigences de performances indiquées dans la demande de l'utilisateur, et recherchent un unique modèle optimal ou une combinaison optimale d'une série de modèles. Le processus de recherche se déroule en divisant l'espace de modèles en éléments individuels de tâche constitués d'un ou de plusieurs modèles, chaque modèle ayant de multiples exemples individuels utilisant ce modèle. La division de la demande de l'utilisateur en éléments de tâche distincts permet au système de mobiliser de nombreuses sources de ressources informatiques différentes, dont les deux ressources de d'informatique en nuage de divers prestataires de

services sur nuage, ainsi que des nuages privés ou des ressources de calcul internes. Le système mobilise également différents types de ressources informatiques, notamment des ressources informatiques qui diffèrent par le système d'exploitation et l'architecture matérielle sur lesquels elles reposent. La possibilité de mobiliser de multiples sources de ressources informatiques, ainsi que de multiples types de ressources informatiques, donne au système plus de souplesse et une plus grande capacité de calcul. La combinaison de l'automatisation, de la souplesse et de la capacité rend possible l'analyse de grands espaces de recherche là où, auparavant, il fallait un long processus manuel. Le système comporte aussi des fonctions de contrainte aptes à permettre à un utilisateur d'adapter une demande à un besoin spécifique de façon qu'elle puisse être restreinte au type de ressources informatiques qu'elle mobilise ou à la quantité de ressources informatiques qu'elle mobilise.

Un exemple d'effet technique des procédés et systèmes décrits ici comprend au moins un des aspects, suivants : (a) l'utilisateur demandeur n'a pas à se préoccuper des détails de l'affectation de ressources de calcul ; (b) mobilisation de différents types et sources de ressources informatiques pour l'exécution du travail de calcul de l'utilisateur ; (c) mobilisation du calcul partagé, à l'aide de prestataires de services d'informatique en nuage, à la fois en interne et par l'Internet pour traiter des problèmes d'Apprentissage Automatique ou autres problèmes de calcul d'un utilisateur ; (d) souplesse et capacité de calcul accrues offertes aux utilisateurs ; (e) réduction des heures de travail humain par l'automatisation d'un Apprentissage Automatique ou d'autres demandes de calculs d'un utilisateur grâce au recours à une base de données de modèles ; (f) plus grandes possibilités d'adaptation à la

taille des données et à la complexité des calculs pour un problème particulier.

Des exemples de formes de réalisation de systèmes et de procédés pour la fourniture automatisée de ressources informatiques hétérogènes pour Apprentissage Automatique sont décrits en détail plus haut. Les systèmes et procédés décrits ici ne se limitent pas aux formes de réalisation spécifiques décrites ici, mais, au contraire, des éléments des systèmes et/ou des étapes des procédés peuvent être utilisés indépendamment et séparément d'autres éléments et/ou étapes décrits ici. Par exemple, les procédés peuvent aussi être utilisés en combinaison avec d'autres systèmes nécessitant des systèmes et procédés informatiques partagés et ne se limitent pas à une mise en œuvre uniquement avec l'identification et la création automatiques de modèles avec des systèmes et des procédés très adaptables décrits ici. Au contraire, les exemples de formes de réalisation peuvent être mis en œuvre et utilisés dans le cadre de nombreuses autres applications d'extraction de concepts.

Bien que des aspects spécifiques de diverses formes de réalisation puissent être illustrés sur certains dessins et pas sur d'autres, cela ne vise qu'une plus grande commodité. Selon les principes des systèmes et procédés décrits ici, tout détail d'un dessin peut être cité et/ou revendiqué en combinaison avec tout détail de n'importe quel autre dessin.

Liste des repères

120	Système informatique
150	Périphérique de mémoire
152	Processeur
154	Interface de présentation
156	Utilisateur
158	Interface utilisateur de saisie
160	Interface de communication
162	Dispositif de stockage
164	Base de données
200	Environnement d'application
201	Système
202	Utilisateur
203	Problème
204	Demande
206	Ressources informatiques
208	Nuage public
210	Nuage privé
212	Ressource de calcul interne
214	Résultat
304	Module de demandes
308	Bibliothèque
310	Module de planification/optimisation de tâche
312	Module d'exécution
313	Modules d'API
314	Module de ressources
400	Diagramme de flux de données
402	Tâche
404	Informations de demande

406	Résultats de demande
500	Diagramme de flux de données
502	Modèle de tâche
504	Modèle individuel de tâche
506	Présentation
600	Diagramme de flux de données
602	Ressource individuelle
603	Gestionnaire de stockage
604	Stockage partagé
606	Données
700	Procédé
702	Recevoir
704	Sélectionner
706	Identifier
708	Déterminer
710	Transmettre
712	Recevoir
800	Procédé
802	Identifier
804	Identifier
806	Déterminer
808	Attribuer
810	Identifier
812	Sélectionner
814	Transmettre
816	Recevoir
900	Exemple de structure de tables de base de données
901	Task Models
903	Models
905	Parameter

906	Storage Resources
907	Storage instances
908	Tasks
911	Job Model Instance
912	Compute Resources
913	Request Info
914	Job
915	Job Model
916	Request
918	Model Deploy
919	Instance Resource
920	Resource Instance

REVENDICATIONS

1. Système (201) pour calculs partagés, comportant :

un module de planification (310) de tâche conçu pour identifier une demande (204) de tâche comprenant une ou plusieurs exigences de demande, ladite demande (204) de tâche comprenant une ou plusieurs tâches individuelles (504) ;

un module de ressources (314) conçu pour :

déterminer un ensemble de ressources informatiques d'exécution parmi une série de ressources informatiques (206) au moins en partie d'après ladite/lesdites exigences de la demande, à chaque ressource informatique de la série de ressources informatiques étant associée une interface de programmation d'application, la série de ressources informatiques comprenant :

au moins une ressource de calcul interne (210, 212) et au moins une ressource d'informatique en nuage publique (208); ou

une pluralité de ressources d'informatique en nuage publique (208); et

attribuer une première ressource informatique dudit ensemble de ressources informatiques d'exécution à une première tâche individuelle comprenant la ou les tâches individuelles (504) ;

une pluralité de modules d'interfaçage (313), chaque module d'interfaçage de ladite pluralité de modules d'interfaçage (313) étant conçu pour faciliter la communication avec une ou plusieurs ressources informatiques de la série de ressources informatiques (206) à l'aide de l'interface de programmation d'application correspondante ; et

un module d'exécution (312) conçu pour :

identifier un premier module d'interfaçage parmi ladite pluralité de modules d'interfaçage (313) au moins en partie d'après

la facilitation de la communication avec la première ressource informatique ; et

transmettre ladite première tâche individuelle à exécuter à la première ressource informatique à l'aide dudit premier module d'interfaçage.

2. Système (201) selon la revendication 1, dans lequel ledit module de ressources (314) est en outre conçu pour attribuer une deuxième ressource informatique parmi ledit ensemble de ressources informatiques d'exécution à une deuxième tâche individuelle comprenant la ou les tâches individuelles (504), et dans lequel ledit module d'exécution (312) est en outre conçu pour :

identifier un deuxième module d'interfaçage parmi ladite pluralité de modules d'interfaçage (313) au moins partiellement d'après la facilitation de la communication avec la deuxième ressource informatique, ledit deuxième module d'interfaçage étant distinct dudit premier module d'interfaçage ; et

transmettre ladite deuxième tâche individuelle à exécuter à la deuxième ressource informatique à l'aide dudit deuxième module d'interfaçage.

3. Système (201) selon la revendication 1, dans lequel ladite/lesdites exigences concernant ladite/lesdites ressources informatiques comprend/ comprennent des exigences de sécurité et/ou des exigences de coût.

4. Système (201) selon la revendication 1, dans lequel ledit module d'exécution (312) est en outre conçu pour :

contrôler la première ressource informatique afin de détecter un éventuel échec de ladite première tâche individuelle ; et

présenter ladite première tâche individuelle audit module de ressource (314) pour l'attribution d'une deuxième ressource

informatique parmi ledit ensemble de ressources informatiques d'exécution.

5. Système (201) selon la revendication 1, comportant en outre un gestionnaire de stockage (603) conçu pour gérer une grappe de stockage partagé (604), la grappe de stockage partagé (604) étant accessible à une ou plusieurs des ressources dudit ensemble de ressources informatiques d'exécution.

6. Système (201) selon la revendication 1, dans lequel ladite demande (204) de tâche définit une limitation de calculs, et dans lequel ledit module de ressources (314) est en outre conçu pour identifier ledit ensemble de ressources informatiques d'exécution au moins en partie d'après ladite limite de calculs.

7. Système (201) selon la revendication 1, dans lequel ledit ensemble de ressources informatiques d'exécution comprend un ensemble de ressources informatiques hétérogènes.

8. Système (201) selon la revendication 1, dans lequel ledit module d'exécution (312) est en outre conçu pour :

sélectionner un algorithme parmi une pluralité d'algorithmes (308) au moins en partie d'après la première ressource informatique ; et

attribuer ledit algorithme à ladite première tâche individuelle.

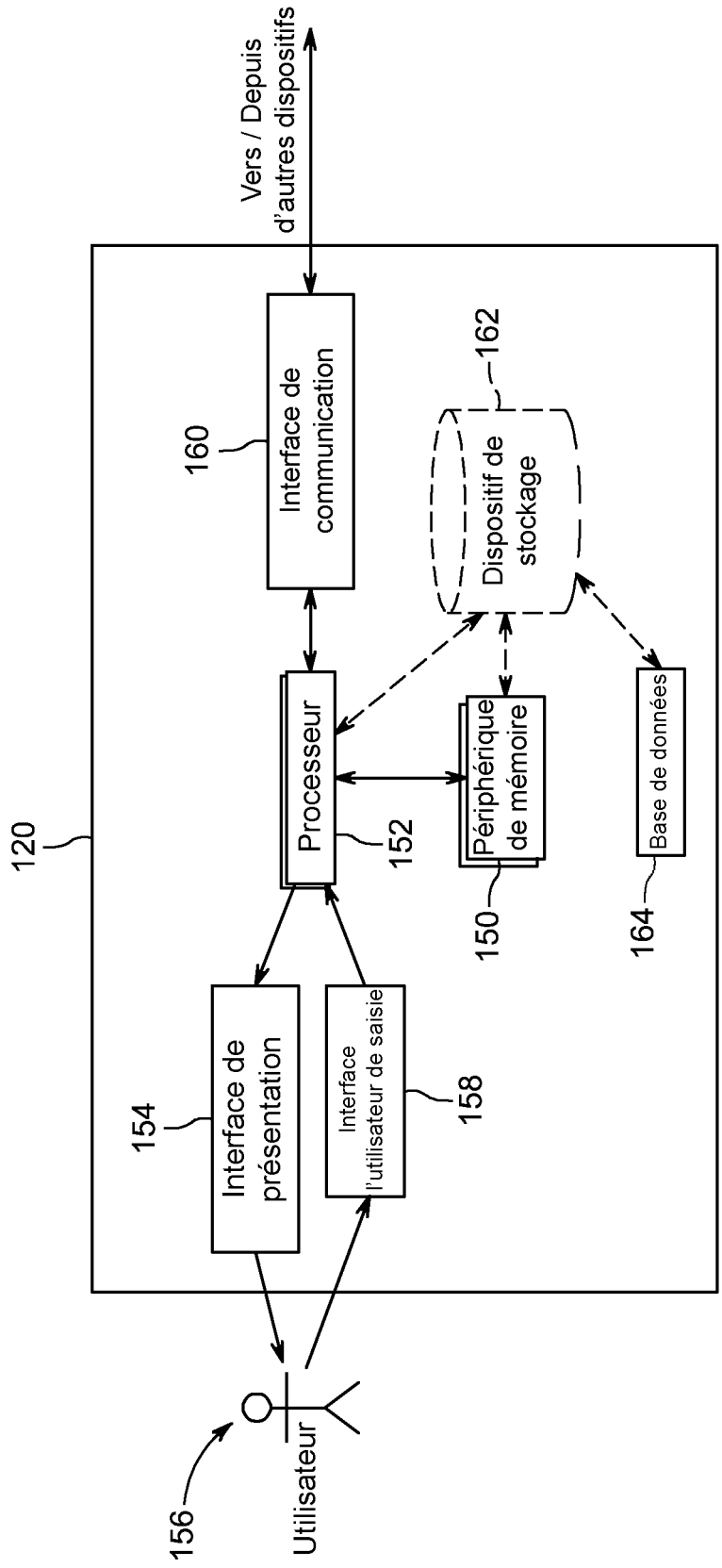

Vers / Depuis
d'autres dispositifs
Interface de
communication
160
162
Dispositif de
stockage
120
Processeur
152
Périphérique
de mémoire
150
Base de données
164
Interface de
présentation
154
Interface
l'utilisateur de saisie
158
Utilisateur
156

FIG. 1

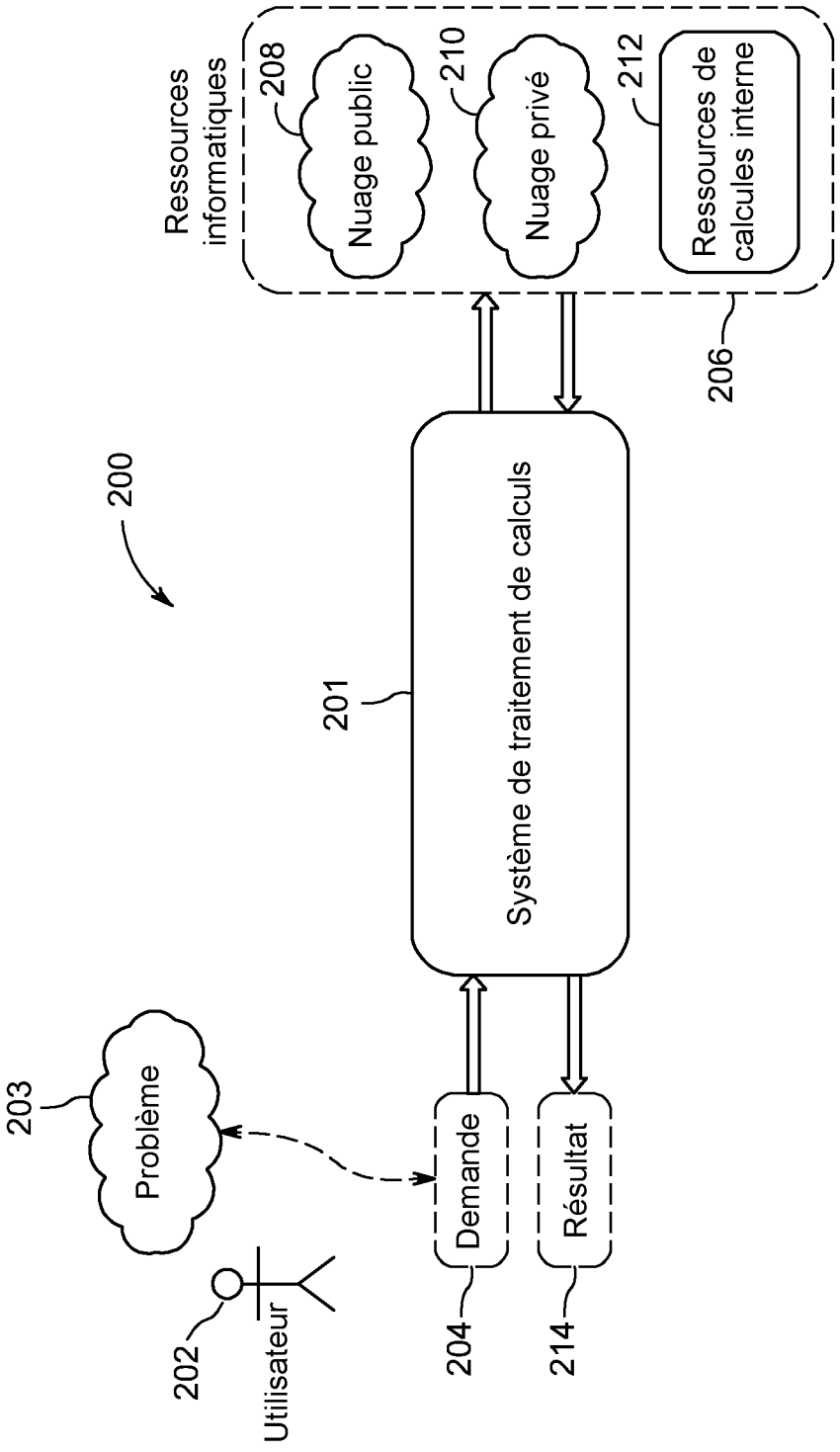

Ressources informatiques
208
Nuage public
210
Nuage privé
212
Ressources de calcules interne
206
200
201
Système de traitement de calculs
203
Problème
Demande
Résultat
204
214
202
Utilisateur

FIG. 2

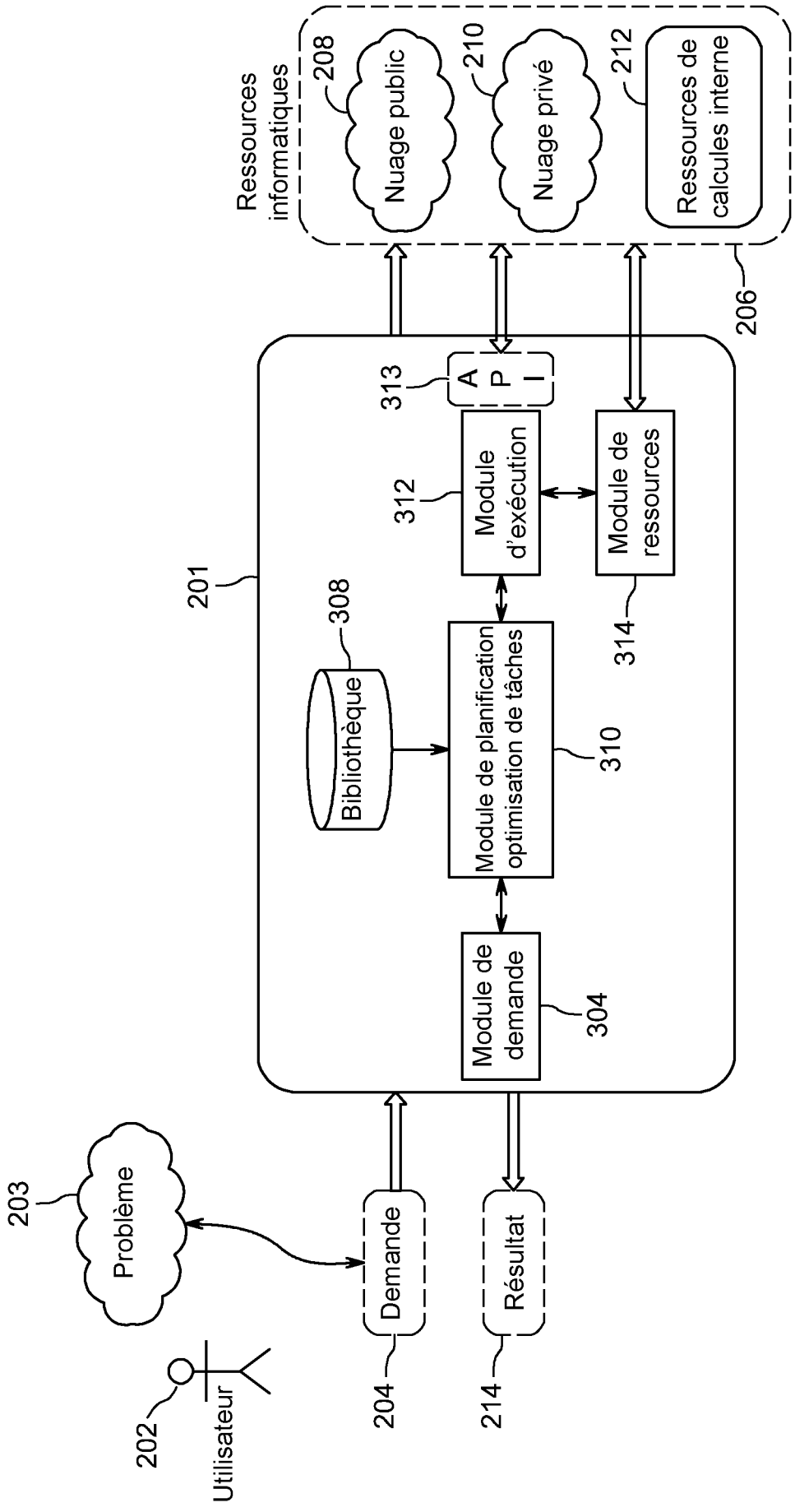

Ressources informatiques
208
Nuage public
210
Nuage privé
212
Ressources de calcules interne
206
313
API
312
Module d'exécution
Module de ressources
201
308
Bibliothèque
Module de planification optimisation de tâches
310
314
Module de demande
304
203
Problème
Demande
Résultat
204
214
202
Utilisateur

FIG. 3

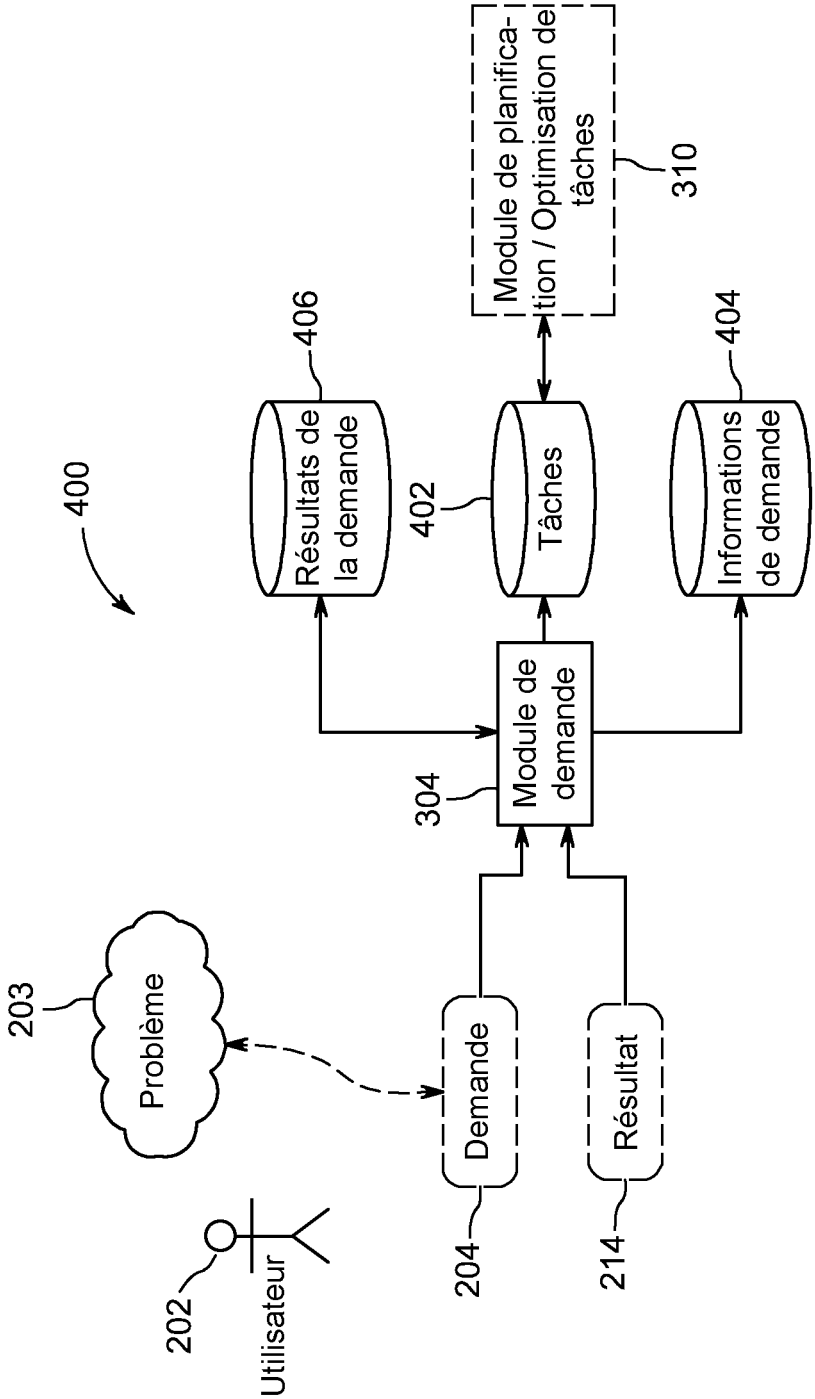

400
Module de planification / Optimisation des tâches
310
406
Résultats de la demande
402
Tâches
404
Informations de demande
304
Module de demande
203
Problème
Demande
Résultat
204
214
202
Utilisateur

FIG. 4

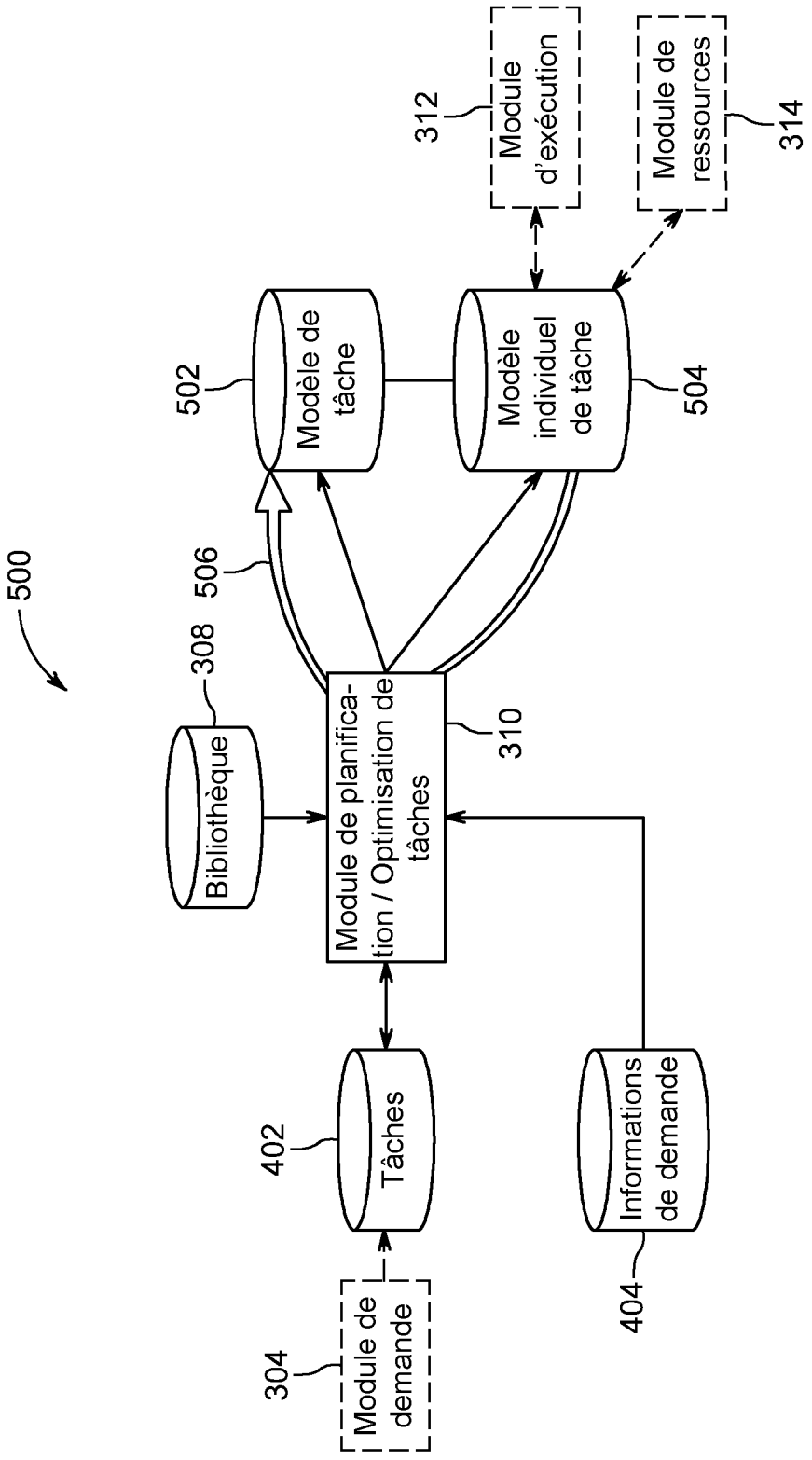

500
312
Module d'exécution
314
Module de ressources
502
Modèle de tâche
Modèle individuel de tâche
504
506
308
Bibliothèque
Module de planification / Optimisation de tâches
310
402
Tâches
Informations de demande
404
304
Module de demande

FIG. 5

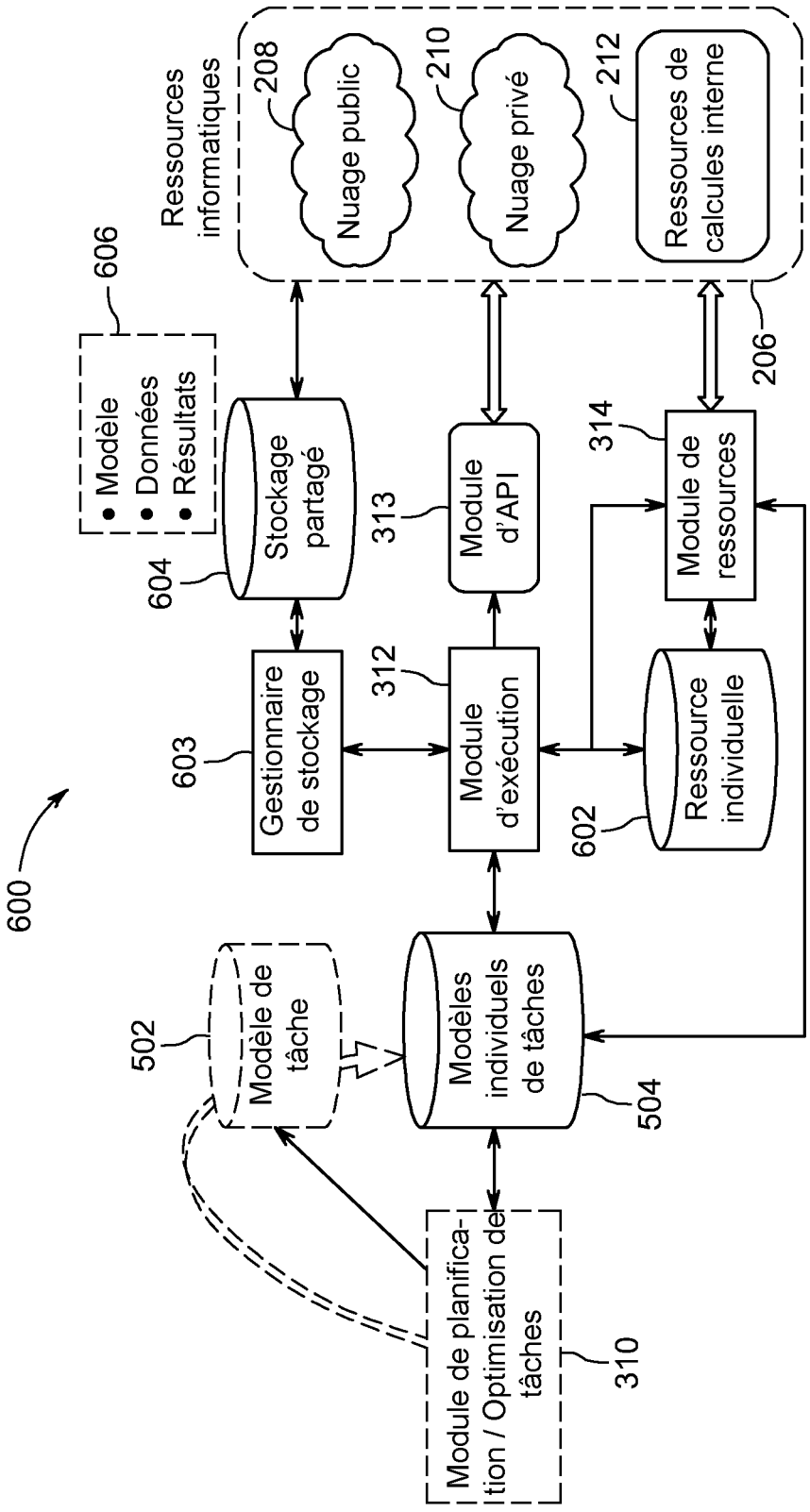

600
Ressources informatiques
208
Nuage public
210
Nuage privé
212
Ressources de calcules interne
206
606
• Modèle
• Données
• Résultats
604
Stockage partagé
313
Module d'API
314
Module de ressources
603
Gestionnaire de stockage
312
Module d'exécution
602
Ressource individuelle
502
Modèle de tâche
Modèles individuels de tâches
504
Module de planification / Optimisation de tâches
310

FIG. 6

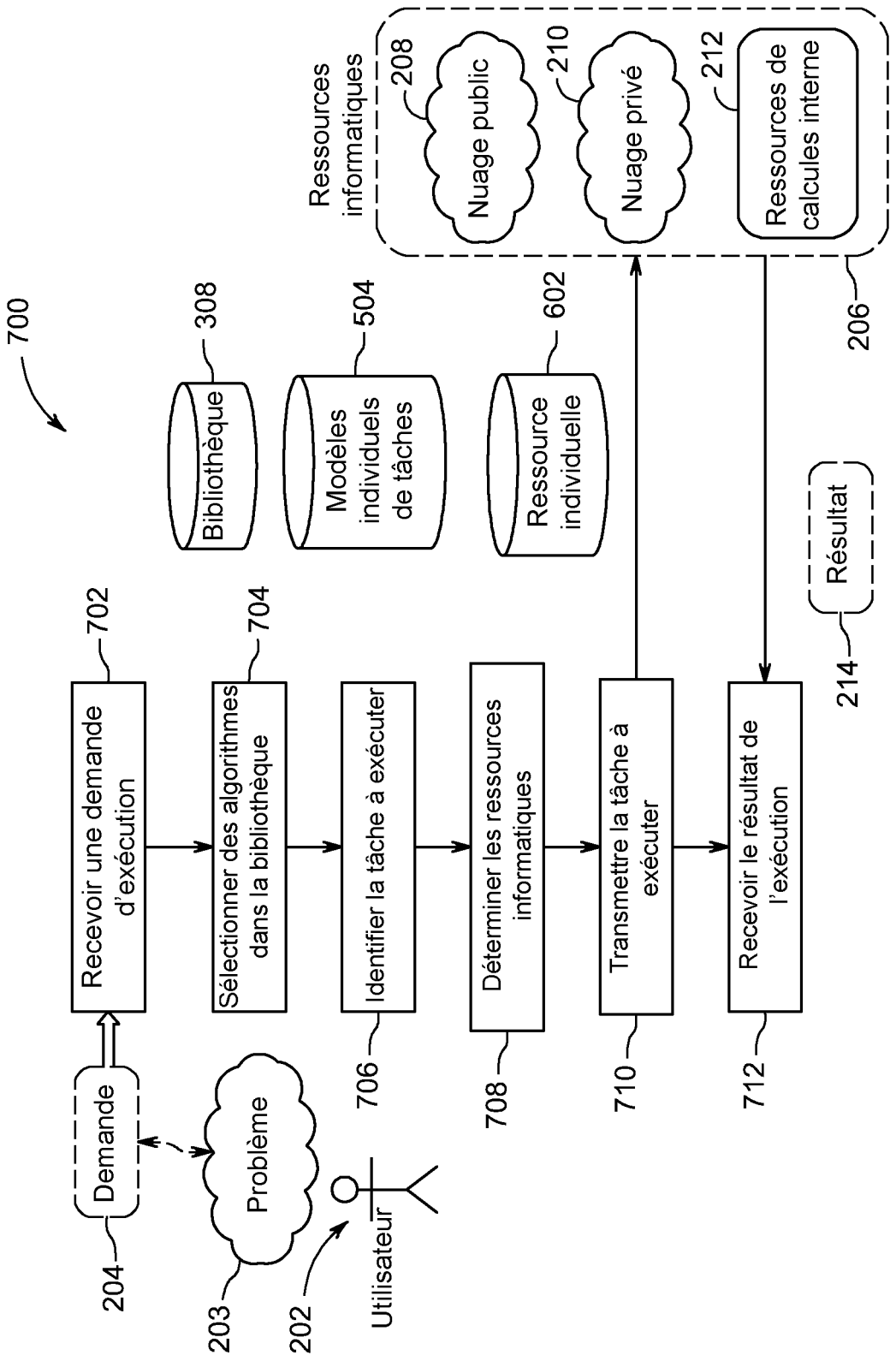

700
Recevoir une demande d'exécution
Sélectionner des algorithmes dans la bibliothèque
Identifier la tâche à exécuter
Déterminer les ressources informatiques
Transmettre la tâche à exécuter
Recevoir le résultat de l'exécution
Demande
Problème
Utilisateur
Bibliothèque
Modèles individuels de tâches
Ressource individuelle
Ressources informatiques
Nuage public
Nuage privé
Ressources de calcules interne
Résultat
202
203
204
206
208
210
212
214
308
504
602
702
704
706
708
710
712

FIG. 7

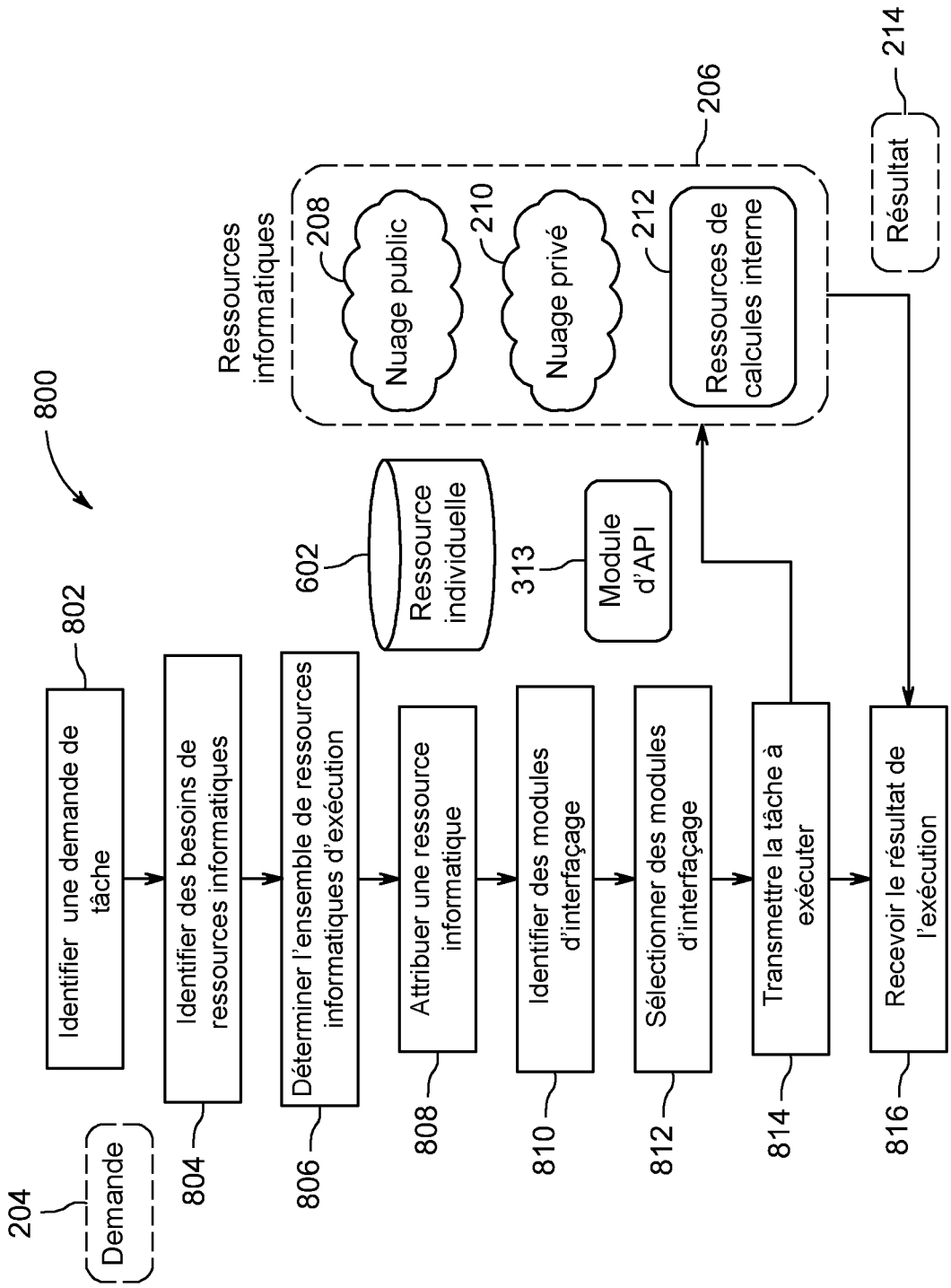

800
802
Identifier une demande de tâche
804
Identifier des besoins de ressources informatiques
806
Déterminer l'ensemble de ressources informatiques d'exécution
808
Attribuer une ressource informatique
810
Identifier des modules d'interfaçage
812
Sélectionner des modules d'interfaçage
814
Transmettre la tâche à exécuter
816
Recevoir le résultat de l'exécution
204
Demande
602
Ressource individuelle
313
Module d'API
Ressources informatiques
206
208
Nuage public
210
Nuage privé
212
Ressources de calcules interne
214
Résultat

FIG. 8

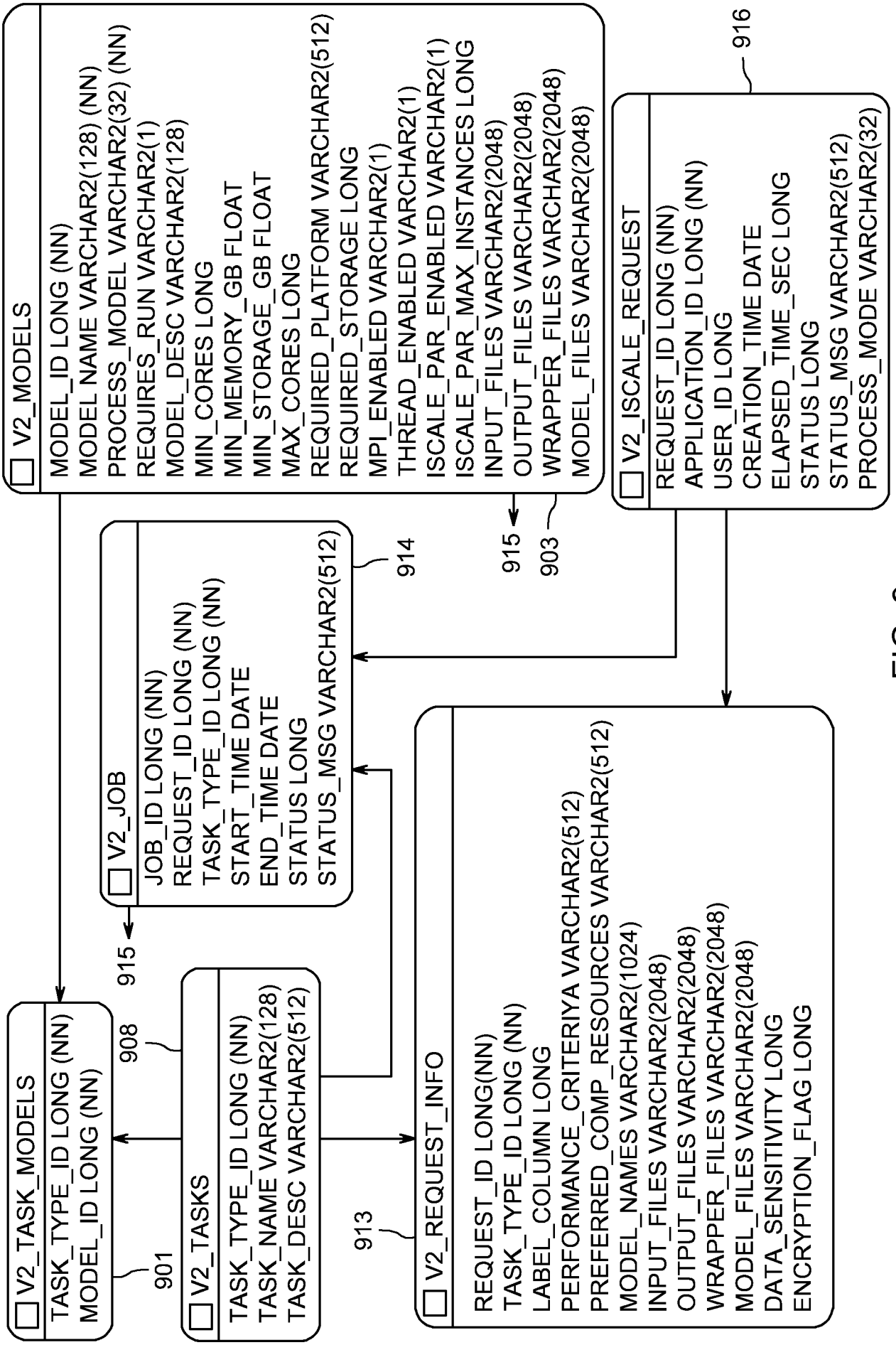

V2_MODELS
MODEL_ID LONG (NN)
MODEL_NAME VARCHAR2(128) (NN)
PROCESS_MODEL VARCHAR2(32) (NN)
REQUIRES_RUN VARCHAR2(1)
MODEL_DESC VARCHAR2(128)
MIN_CORES LONG
MIN_MEMORY_GB FLOAT
MIN_STORAGE_GB FLOAT
MAX_CORES LONG
REQUIRED_PLATFORM VARCHAR2(512)
REQUIRED_STORAGE LONG
MPI_ENABLED VARCHAR2(1)
THREAD_ENABLED VARCHAR2(1)
ISCALE_PAR_ENABLED VARCHAR2(1)
ISCALE_PAR_MAX_INSTANCES LONG
INPUT_FILES VARCHAR2(2048)
OUTPUT_FILES VARCHAR2(2048)
WRAPPER_FILES VARCHAR2(2048)
MODEL_FILES VARCHAR2(2048)
V2_ISCALE_REQUEST
REQUEST_ID LONG (NN)
APPLICATION_ID LONG (NN)
USER_ID LONG
CREATION_TIME DATE
ELAPSED_TIME_SEC LONG
STATUS LONG
STATUS_MSG VARCHAR2(512)
PROCESS_MODE VARCHAR2(32)
916
V2_JOB
JOB_ID LONG (NN)
REQUEST_ID LONG (NN)
TASK_TYPE_ID LONG (NN)
START_TIME DATE
END_TIME DATE
STATUS LONG
STATUS_MSG VARCHAR2(512)
914
915
903
915
V2_TASK_MODELS
TASK_TYPE_ID LONG (NN)
MODEL_ID LONG (NN)
908
901
V2_TASKS
TASK_TYPE_ID LONG (NN)
TASK_NAME VARCHAR2(128)
TASK_DESC VARCHAR2(512)
913
V2_REQUEST_INFO
REQUEST_ID LONG(NN)
TASK_TYPE_ID LONG (NN)
LABEL_COLUMN LONG
PERFORMANCE_CRITERIYA VARCHAR2(512)
PREFERRED_COMP_RESOURCES VARCHAR2(512)
MODEL_NAMES VARCHAR2(1024)
INPUT_FILES VARCHAR2(2048)
OUTPUT_FILES VARCHAR2(2048)
WRAPPER_FILES VARCHAR2(2048)
MODEL_FILES VARCHAR2(2048)
DATA_SENSITIVITY LONG
ENCRYPTION_FLAG LONG

FIG. 9

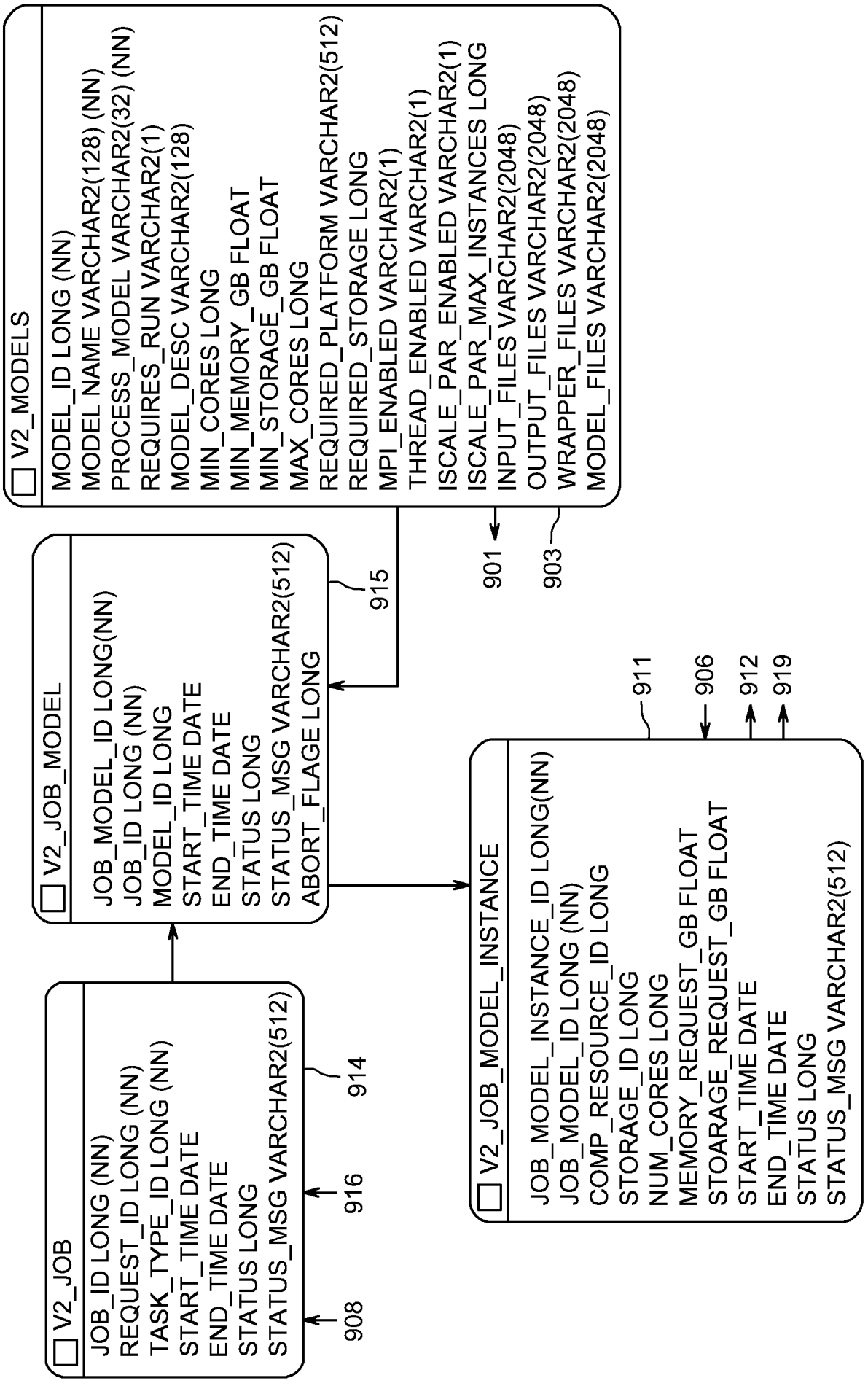

V2_MODELS
MODEL_ID LONG (NN)
MODEL NAME VARCHAR2(128) (NN)
PROCESS_MODEL VARCHAR2(32) (NN)
REQUIRES_RUN VARCHAR2(1)
MODEL_DESC VARCHAR2(128)
MIN_CORES LONG
MIN_MEMORY_GB FLOAT
MIN_STORAGE_GB FLOAT
MAX_CORES LONG
REQUIRED_PLATFORM VARCHAR2(512)
REQUIRED_STORAGE LONG
MPI_ENABLED VARCHAR2(1)
THREAD_ENABLED VARCHAR2(1)
ISCALE_PAR_ENABLED VARCHAR2(1)
ISCALE_PAR_MAX_INSTANCES LONG
INPUT_FILES VARCHAR2(2048)
OUTPUT_FILES VARCHAR2(2048)
WRAPPER_FILES VARCHAR2(2048)
MODEL_FILES VARCHAR2(2048)
V2_JOB_MODEL
JOB_MODEL_ID LONG(NN)
JOB_ID LONG (NN)
MODEL_ID LONG
START_TIME DATE
END_TIME DATE
STATUS LONG
STATUS_MSG VARCHAR2(512)
ABORT_FLAGE LONG
V2_JOB
JOB_ID LONG (NN)
REQUEST_ID LONG (NN)
TASK_TYPE_ID LONG (NN)
START_TIME DATE
END_TIME DATE
STATUS LONG
STATUS_MSG VARCHAR2(512)
V2_JOB_MODEL_INSTANCE
JOB_MODEL_INSTANCE_ID LONG(NN)
JOB_MODEL_ID LONG (NN)
COMP_RESOURCE_ID LONG
STORAGE_ID LONG
NUM_CORES LONG
MEMORY_REQUEST_GB FLOAT
STOARAGE_REQUEST_GB FLOAT
START_TIME DATE
END_TIME DATE
STATUS LONG
STATUS_MSG VARCHAR2(512)
901
903
915
914
916
908
911
906
912
919

FIG. 10

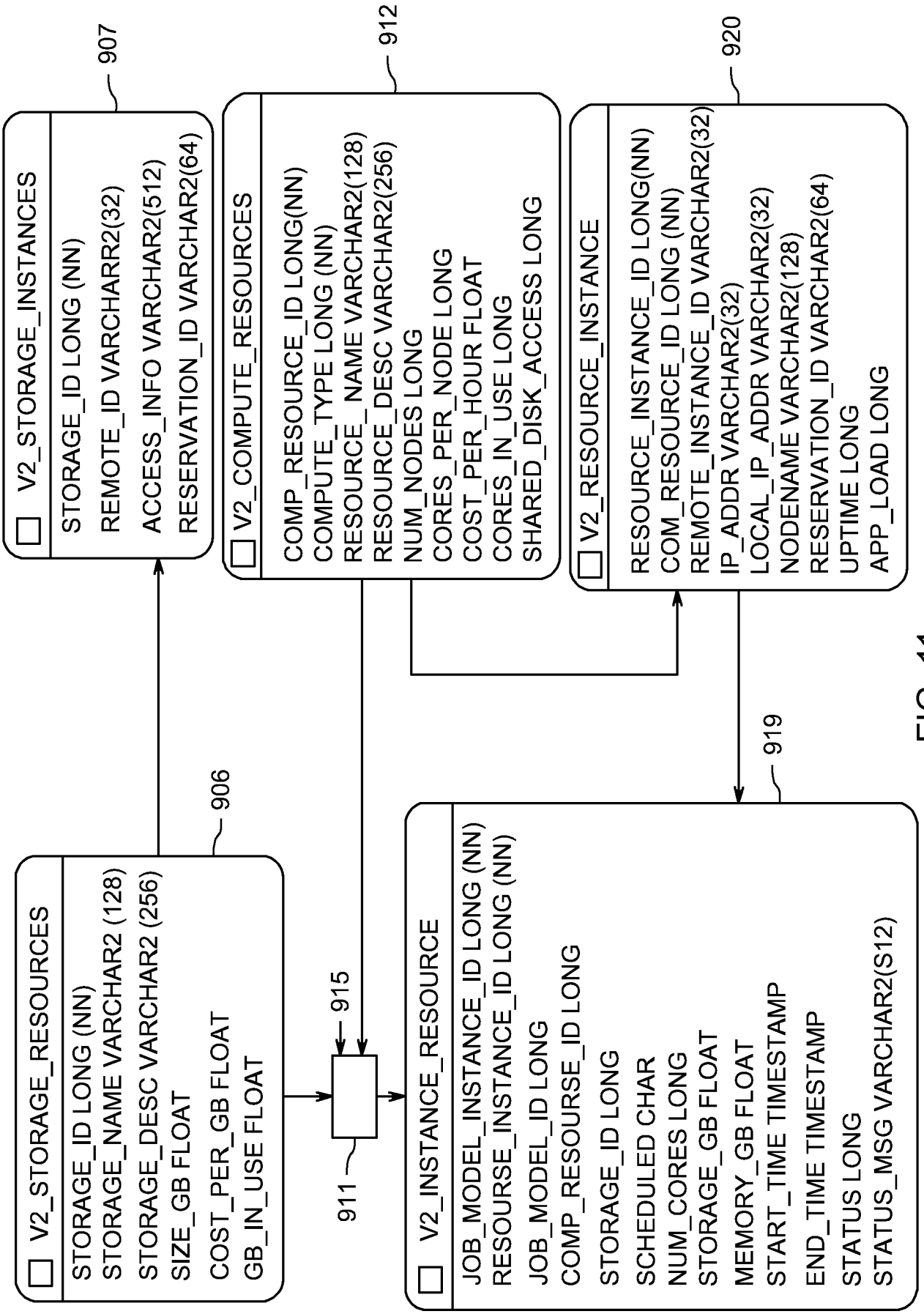

V2_STORAGE_INSTANCES
STORAGE_ID LONG (NN)
REMOTE_ID VARCHARR2(32)
ACCESS_INFO VARCHAR2(512)
RESERVATION_ID VARCHAR2(64)
907
V2_COMPUTE_RESOURCES
COMP_RESOURCE_ID LONG(NN)
COMPUTE_TYPE LONG (NN)
RESOURCE_ NAME VARCHAR2(128)
RESOURCE_DESC VARCHAR2(256)
NUM_NODES LONG
CORES_PER_NODE LONG
COST_PER_HOUR FLOAT
CORES_IN_USE LONG
SHARED_DISK_ACCESS LONG
912
V2_RESOURCE_INSTANCE
RESOURCE_INSTANCE_ID LONG(NN)
COM_RESOURCE_ID LONG (NN)
REMOTE_INSTANCE_ID VARCHAR2(32)
IP_ADDR VARCHAR2(32)
LOCAL_IP_ADDR VARCHAR2(32)
NODENAME VARCHAR2(128)
RESERVATION_ID VARCHAR2(64)
UPTIME LONG
APP_LOAD LONG
920
V2_STORAGE_RESOURCES
STORAGE_ID LONG (NN)
STORAGE_NAME VARCHAR2 (128)
STORAGE_DESC VARCHAR2 (256)
SIZE_GB FLOAT
COST_PER_GB FLOAT
GB_IN_USE FLOAT
906
915
911
V2_INSTANCE_RESOURCE
JOB_MODEL_INSTANCE_ID LONG (NN)
RESOURSE_INSTANCE_ID LONG (NN)
JOB_MODEL_ID LONG
COMP_RESOURSE_ID LONG
STORAGE_ID LONG
SCHEDULED CHAR
NUM_CORES LONG
STORAGE_GB FLOAT
MEMORY_GB FLOAT
START_TIME TIMESTAMP
END_TIME TIMESTAMP
STATUS LONG
STATUS_MSG VARCHAR2(S12)
919

FIG. 11