

12 DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 27.01.14.

30 Priorité : 25.03.13 US 13849716.

43 Date de mise à la disposition du public de la demande : 26.09.14 Bulletin 14/39.

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 :

○ Demande(s) d'extension :

71 Demandeur(s) : GE AVIATION SYSTEMS LLC — US.

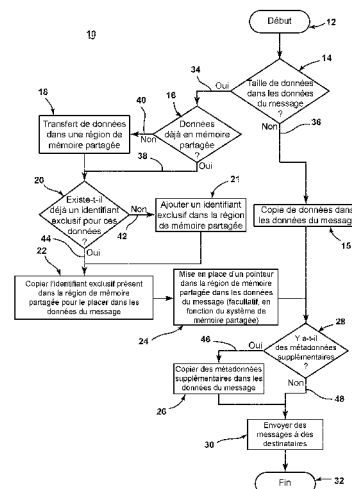
72 Inventeur(s) : OCCHIPINTI BENJAMIN THOMAS, KUCZYNSKI KONRAD ROBERT et BUEHLER ERIC DANIEL.

73 Titulaire(s) : GE AVIATION SYSTEMS LLC.

74 Mandataire(s) : CASALONGA & ASSOCIES.

54 PROCÉDE DE TRANSMISSION HYBRIDE DE MESSAGES A MEMOIRE PARTAGEE.

57 Procédé (10) pour communiquer des données entre au moins deux coprocesseurs dans un environnement informatique unique dans lequel l'un des coprocesseurs est un processeur émetteur et l'autre des coprocesseurs est un processeur récepteur et l'environnement informatique unique comprend une mémoire accessible aux coprocesseurs, le procédé comportant, entre autres, la détermination (14), dans le processeur émetteur, de la taille des données à communiquer, l'ajout (15) des données dans un message et la transmission (30) du message au processeur récepteur si la taille des données est inférieure ou égale (36) à une valeur seuil prédéterminée, et la transmission (24, 30) d'un pointeur au processeur récepteur, le pointeur désignant un emplacement des données dans la mémoire, si la taille des données est supérieure (34) à la valeur seuil prédéterminée.



## **Procédé de transmission hybride de messages à mémoire partagée**

5 L'invention concerne la communication de données, en particulier entre au moins deux coprocesseurs.

10 Les performances des codes pour des applications exécutées dans un environnement informatique à multiprocesseur ou processeur multicoeur peuvent être déterminées, en partie, par la manière dont les éléments de traitement individuels présents dans l'environnement communiquent les uns avec les autres. La mémoire partagée et la transmission de messages sont deux grandes catégories de paradigmes de communication utilisées pour les communications entre processeurs. Il existe des techniques de programmation hybride qui recourent aux deux paradigmes de communication.

15 Un aspect de l'invention concerne un procédé de transfert de données entre deux ou plus de deux coprocesseurs dans un environnement informatique unique dans lequel l'un des coprocesseurs est un processeur émetteur et un autre des coprocesseurs est un processeur récepteur. L'environnement informatique unique comprend une mémoire accessible aux coprocesseurs. Le procédé comporte la détermination, dans le processeur émetteur, de la taille des données à communiquer. Si la taille des données est inférieure ou égale à une valeur seuil prédéterminée, le procédé insère les données dans un message et transmet le message au processeur récepteur. Si la taille des données est supérieure à la valeur seuil prédéterminée, le procédé transmet un pointeur au processeur récepteur, le pointeur indiquant un emplacement des données dans la mémoire.

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 :

5 la Figure 1 est un organigramme illustrant un procédé de transmission de messages hybrides avec mémoire partagée selon une forme de réalisation de l'invention.

10 Les exemples de formes de réalisation sont décrits en référence aux dessins. Ces dessins illustrent certains détails de formes de réalisation spécifiques qui emploient un module, un procédé ou un programme informatique décrit ici. Cependant, les dessins ne doivent pas être interprétés comme imposant d'éventuelles limites susceptibles de figurer sur les dessins. Le procédé et le programme informatique peuvent être placés sur n'importe quels supports exploitables par un ordinateur pour l'exécution de leur exploitation. Les formes de réalisation peuvent être mises en œuvre à l'aide d'un processeur informatique existant ou par un processeur informatique spécifique installé à cette fin ou à une autre, ou par un système filaire.

20 Comme indiqué plus haut, les formes de réalisation décrites ici peuvent comporter un programme informatique comprenant des supports exploitables par ordinateur destinés à porter ou à stocker des instructions exécutables par ordinateur ou des structures de données. Ces supports exploitables par ordinateur peuvent être n'importe quels supports existants, accessibles à un ordinateur polyvalent ou spécifique, ou à une autre machine à processeur. A titre d'exemple, ces supports exploitables par ordinateur peuvent comprendre une mémoire vive, une mémoire morte, une EPROM, une EEPROM, un CD-ROM ou autre moyen de mémorisation sur disque optique, des moyens de mémorisation sur disque magnétique ou d'autres dispositifs magnétiques de mémorisation, ou tout autre

30

support utilisable pour porter ou stocker un code de programme voulu sous la forme d'instructions ou de structures de données exécutables par ordinateur et accessible à un ordinateur polyvalent ou spécifique ou à une autre machine à processeur. Quand des informations sont transmises ou fournies à une machine via un réseau ou une autre liaison de communication (filaire, radioélectrique, ou à la fois combinée et radioélectrique), la machine perçoit à juste titre la liaison comme un support exploitable par ordinateur. Ainsi, toute liaison de ce type est appelée à juste titre support exploitable par ordinateur. Des combinaisons des moyens ci-dessus entrent également dans le cadre des supports exploitables par ordinateur. Les instructions exécutables par ordinateur comprennent, par exemple, des instructions et des données, qui amènent un ordinateur polyvalent, un ordinateur spécifique ou des processeurs spécifiques à exécuter une certaine fonction ou un certain groupe de fonctions.

Des formes de réalisation seront décrites dans le contexte général d'étapes d'un procédé qui peuvent être mises en œuvre dans une même forme de réalisation par un programme comprenant des instructions exécutables par ordinateur, notamment un code de programme, par exemple sous la forme de modules de programmes exécutés par des machines dans des environnements en réseau. Globalement, les modules de programmes comprennent des routines, des programmes, des objets, des composants, des structures de données, etc., qui ont pour effet technique d'exécuter des tâches particulières ou mettent en œuvre des types particuliers de données abstraites. Les instructions exécutables par ordinateur, les structures de données associées et les modules de programmes constituent des exemples de code de programme pour exécuter des étapes du procédé exposé ici. L'ordre particulier de ces instructions

exécutables ou les structures de données associées constituent des exemples d'actions correspondantes pour exécuter les fonctions décrites dans ces étapes.

5 Les formes de réalisation peuvent être mises en œuvre dans un environnement en réseau à l'aide de liaisons logiques avec un ou plusieurs ordinateurs distants à processeurs. Les liaisons logiques peuvent comprendre un réseau local (LAN) et un réseau étendu (WAN) qui sont présentés ici à titre d'exemple nullement limitatif. Ces environnements de mise en réseau sont courants dans les 10 réseaux informatiques à l'échelle d'un bureau ou d'une entreprise, dans les intranets et l'Internet et peuvent utiliser toutes sortes de protocoles de communication différents. Les spécialistes de la technique comprendront que ces environnements informatiques en réseau englobent ordinairement de nombreux types de configuration 15 de système informatique, dont les ordinateurs personnels, les dispositifs de poche, les systèmes à multiprocesseur, les équipements grand public à microprocesseur ou programmables, les PC en réseau, les mini-ordinateurs, les ordinateurs centraux et autres.

20 Les formes de réalisation peuvent aussi être mises en œuvre dans des environnements informatiques distribués où des tâches sont effectuées par des dispositifs de traitement locaux et distants qui sont en liaison (soit par des liaisons filaires, des liaisons radioélectriques, soit par une combinaison de liaisons filaires ou 25 radioélectriques) par l'intermédiaire d'un réseau de communication. Dans un environnement informatique distribué, des modules de programmes peuvent être situés dans des dispositifs de mémorisation aussi bien locaux que distants.

30 Un exemple de système pour mettre en œuvre l'ensemble ou des parties des exemples de formes de réalisation pourraient

comprendre un dispositif informatique polyvalent sous la forme d'un ordinateur, comportant un processeur, une mémoire système et un bus système, qui couple au processeur divers composants du système, dont la mémoire système. La mémoire système peut  
5 comprendre une mémoire morte (ROM) une mémoire vive (RAM). L'ordinateur peut aussi comprendre un lecteur de disque dur pour lire ou enregistrer des données sur un disque dur magnétique, un lecteur de disque magnétique pour lire ou enregistrer des données sur un disque magnétique amovible, et un lecteur de disque optique  
10 pour lire ou enregistrer des données sur un disque optique amovible tel qu'un CD-ROM ou d'autres supports optiques. Les lecteurs et leurs supports correspondants exploitables par ordinateur permettent un stockage rémanent d'instructions exécutables par ordinateur, de structures de données, de modules de programmes et d'autres  
15 données pour l'ordinateur.

Les effets techniques du procédé exposé dans les formes de réalisation comprennent l'amélioration des performances et l'évolutivité de systèmes informatiques à multiprocesseur. Des messages volontairement lourds peuvent être échangés entre des  
20 coprocesseurs aussi efficacement que des messages volontairement légers. Le délai supplémentaire nécessaire à la transmission d'un message d'un coprocesseur à de multiples processeurs récepteurs est négligeable. Les performances de systèmes de traitement en temps réel avec de grands types de données et un degré élevé de variation  
25 entre les exigences de taille de différents messages, comme dans le cas de systèmes d'imagerie hyperspectrale, peuvent être améliorées par le procédé exposé dans les formes de réalisation.

Dans un environnement informatique unique à au moins deux coprocesseurs, il est souvent nécessaire, au cours du traitement de  
30 données entrées dans l'environnement informatique, de

communiquer des données entre des coprocesseurs. Pour transmettre des données entre des coprocesseurs, un processeur émetteur doit envoyer un message à un processeur récepteur pour communiquer les données ou des informations sur les données au processeur récepteur. Dans un environnement informatique où la transmission de messages est le paradigme de communication, le processeur émetteur peut envoyer un message contenant les données à un ou plusieurs processeurs récepteurs. Selon une autre possibilité, dans un environnement informatique à système de mémoire partagée, le processeur émetteur peut envoyer, à un ou plusieurs processeurs récepteurs, un message qui code l'emplacement en mémoire accessible à tous les coprocesseurs où se trouvent les données. Ensuite, chaque coprocesseur peut accéder directement aux données en mémoire partagée.

La Figure 1 est un organigramme illustrant un procédé de communication de données entre des coprocesseurs selon une forme de réalisation de l'invention qui utilise un hybride d'acheminement de message à mémoire partagée. Pour commencer, lors d'une étape 12, au cours de l'exécution d'une application dans un système à multiprocesseur, il peut être nécessaire à un coprocesseur dans un environnement informatique de communiquer un bloc de données particulier à au moins un autre coprocesseur. Le bloc de données aura une certaine taille finie déterminable, par exemple, par le nombre d'octets de mémoire nécessaires au codage numérique des données.

Lors d'une étape 14, le processeur émetteur peut déterminer la taille des données à communiquer et comparer la taille à un seuil prédéterminé. Si, en 36, la taille des données à communiquer est inférieure ou égale au seuil prédéterminé, le processeur émetteur peut alors, au cours une étape 15, copier les données dans un

message qui sera communiqué à un ou plusieurs processeurs récepteurs. Inversement, en 34, si la taille des données à communiquer est supérieure au seuil prédéterminé, le processeur émetteur peut alors déterminer, au cours d'une étape 16, si les données se trouvent en mémoire partagée.

Dans une forme de réalisation de l'invention, le seuil prédéterminé peut être codé en tant que paramètre statique comme dans un fichier de configuration. Le fichier de configuration peut être un fichier de texte ASCII codé avec des paramétrages initiaux couramment employés pour des applications, des serveurs et des systèmes d'exploitation et souvent écrit en langage de balisage tel que XML. Selon une autre possibilité, le seuil prédéterminé peut être communiqué à l'environnement informatique en tant que paramètre au moment de l'instanciation d'une application, voire être établi comme variable d'environnement du système d'exploitation de l'environnement informatique. Dans tous ces exemples est codé un paramètre statique qui identifie le seuil prédéterminé afin de déterminer si un bloc de données d'une taille particulière sera communiqué plus efficacement par mémoire partagée ou par acheminement de message.

Dans une autre forme de réalisation de l'invention, le seuil prédéterminé peut être adaptable. Le processeur émetteur peut insérer un horodatage reposant sur un signal d'horloge global connu de tous les coprocesseurs quand il communique un message, puis le processeur récepteur peut analyser les délais pour déterminer s'il faut modifier le seuil prédéterminé. Les facteurs susceptibles d'affecter les délais et de modifier le seuil optimal comprennent la saturation du bus, la recherche de personnes, les interruptions et les vitesses d'horloges d'unités centrales.

En 40, si le processeur émetteur détermine que les données à communiquer sont stockées à un emplacement de mémoire inaccessible à un ou plusieurs processeurs récepteurs, le processeur émetteur peut alors, au cours d'une étape 18, transférer les données dans une région de mémoire partagée. Le processeur émetteur peut sauter l'étape de transfert de données dans une région de mémoire partagée si, en 38, le processeur émetteur détermine que les données à communiquer se trouvent déjà dans une région de mémoire partagée.

Le processeur émetteur peut ensuite, lors d'une étape 20, déterminer s'il existe un identifiant exclusif ; ainsi, celui-ci a été créé et attribué aux données à communiquer aux processeurs récepteurs. L'identifiant exclusif est un code attribué à un objet informatique tel qu'un bloc de données destiné à permettre à des environnements informatiques distribués d'identifier des informations d'une manière exclusive. Dans le contexte de la présente invention, l'identifiant exclusif est attribué aux données à communiquer de telle sorte qu'un processeur récepteur qui accède aux données via une mémoire partagée puisse déterminer si les données sont valides. Des données invalides peuvent résulter de problèmes de synchronisation et d'expiration par suite desquels de nouvelles données peuvent écraser d'anciennes données dans la région de mémoire partagée avant qu'un processeur récepteur n'accède aux anciennes données correspondant à l'identifiant exclusif. L'identifiant exclusif peut être l'une des mises en œuvre d'identifiants universellement exclusifs (UUID) bien connus dans la technique de conception de logiciels. Selon une autre possibilité, l'identifiant exclusif peut être un horodatage pour coder un instant concernant les données, notamment celui où elles ont été

enregistrées dans la région de mémoire partagée ou, en soi, l'instant où l'identifiant exclusif a été créé.

5 En 42, si le processeur émetteur détermine qu'il n'existe pas d'identifiant exclusif pour les données à communiquer situées en mémoire partagée, le processeur émetteur peut alors, au cours d'une étape 21, créer un identifiant exclusif et l'ajouter dans la région de mémoire partagée, à l'emplacement des données. Le processeur émetteur peut sauter l'étape d'ajout d'un identifiant exclusif aux données dans une région de mémoire partagée si, en 44, le  
10 processeur émetteur détermine qu'un identifiant exclusif pour les données à communiquer existe déjà et se trouve dans une région de mémoire partagée.

Une fois que le processeur d'émission a déterminé l'identifiant exclusif pour les données en mémoire partagée à  
15 communiquer, le processeur émetteur peut, lors d'une étape 22, copier l'identifiant exclusif dans la région de mémoire partagée pour le placer dans un message à envoyer au(x) processeur(s) émetteur(s). De plus, lors d'une étape 24, le processeur émetteur peut placer un pointeur dans le message, le pointeur étant une  
20 valeur indiquant une adresse en mémoire partagée qui situe les données à communiquer. Bien que le fait de placer un pointeur dans le message à communiquer soit une forme de réalisation préférée de l'invention, un environnement informatique spécifique tel que le type de système à mémoire partagée permet d'éviter de  
25 communiquer le pointeur directement du processeur émetteur au processeur récepteur. Par exemple, l'environnement informatique peut être configuré de façon que la mémoire partagée soit divisée d'une manière où l'emplacement des données à communiquer puisse être vérifié par le processeur récepteur en effectuant une opération

arithmétique modulaire sur l'identifiant exclusif présent dans le message.

Le message à communiquer à au moins un processeur récepteur peut avoir une copie des données du message comme lors  
5 d'une étape 15 ou peut contenir un pointeur d'un emplacement en mémoire partagée comme lors d'une étape 24. Quoi qu'il en soit, le processeur émetteur peut déterminer, lors d'une étape 28, si des métadonnées supplémentaires doivent être transmises dans le message. Si, en 46, le processeur émetteur décide d'ajouter des  
10 métadonnées supplémentaires dans le message, le processeur émetteur peut alors, au cours d'une étape 26, copier les métadonnées supplémentaires dans le message. Le processeur émetteur peut sauter l'étape de copie des métadonnées supplémentaires dans le message si, en 48, le processeur émetteur  
15 détermine qu'il n'y a pas de métadonnées disponibles ou nécessaires pour le message.

Les métadonnées peuvent être n'importe quel jeu de données qui fournit des informations supplémentaires pour décrire les données à communiquer et pour permettre une communication des  
20 plus efficaces de ces données. Par exemple, les métadonnées peuvent contenir des informations de délais qui caractérisent le canal nécessaire pour transmettre les données du processeur émetteur à un processeur récepteur particulier. Dans un environnement informatique où les données peuvent représenter une  
25 image hyperspectrale formatable de plusieurs manières différentes en fonction de caractéristiques telles que le nombre de bandes spectrales, la résolution, les dimensions de l'image et l'ordre des bandes, les métadonnées peuvent coder le format spécifique de l'image hyperspectrale. Des informations supplémentaires  
30 caractérisant l'optique servant à acquérir des images, telles que le

nombre F-stop et la sensibilité ISO, peuvent être codées comme métadonnées. Par ailleurs, l'historique du traitement, dont les transformées appliquées à l'image hyperspectrale, peut être mémorisé sous la forme de métadonnées.

5           Le processeur émetteur peut ensuite, lors d'une étape 30, transmettre le message au(x) processeur(s) récepteur(s). Le procédé se termine par une étape 32 où chaque processeur récepteur peut décoder le message pour soit déterminer directement les données si elles sont codées dans le message, soit extraire les données de leur  
10 emplacement en mémoire partagée si le message contient un identifiant et un pointeur exclusifs.

          De nombreux environnements informatiques peuvent bénéficier de l'invention décrite ci-dessus. Bien que les entités de traitement pour l'émission et la réception soient appelées  
15 coprocesseurs, l'invention s'applique tout autant à des processus, des processus légers au niveau de noyaux et des fibres. Parmi les mises en œuvre matérielles convenant particulièrement à l'invention figurent les processeurs graphiques (GPU) et les processeurs multicoeurs, en raison du grand parallélisme de leur architecture de  
20 traitement.

### Liste des repères

	10	Procédé de communication de données entre des coprocesseurs
5	12	Etape initiale du procédé 10
	14	Etape de détermination de la taille des données à communiquer et de comparaison de la taille avec un seuil prédéterminé
	15	Etape de copie des données dans un message
10	16	Etape de détermination de ce que les données se trouvent ou non en mémoire partagée
	18	Etape de transfert des données dans une région de mémoire partagée
	20	Etape de détermination de l'existence d'un identifiant exclusif
15	21	Etape de création d'un identifiant exclusif et d'ajout de celui-ci dans la région de mémoire partagée, à l'emplacement des données
	22	Etape de copie de l'identifiant exclusif présent dans la région de mémoire partagée pour le placer dans un message
20	24	Etape de mise en place d'un pointeur dans le message
	26	Etape de copie des métadonnées supplémentaires dans le message
25	28	Etape de détermination de la nécessité de transmission de métadonnées supplémentaires dans le message
	30	Etape d'émission du message
	32	Etape finale du procédé 10
30	34	La taille des données à communiquer est supérieure au seuil prédéterminé

- 36 La taille des données à communiquer est inférieure ou égale au seuil prédéterminé
- 38 Les données à communiquer se trouvent déjà dans une région de mémoire partagée
- 5 40 Les données à communiquer sont enregistrées dans un emplacement en mémoire qui est inaccessible à un ou plusieurs processeurs récepteurs
- 42 Pour les données à communiquer, il n'existe pas d'identifiant exclusif situé en mémoire partagée
- 10 44 Un identifiant exclusif pour les données à communiquer existe déjà et se trouve dans une région de mémoire partagée
- 46 Ajouter des métadonnées supplémentaires dans le message
- 48 Pas de métadonnées supplémentaires disponibles ou nécessaires pour le message
- 15

## REVENDICATIONS

1. Procédé pour communiquer des données entre au moins deux coprocesseurs dans un environnement informatique unique dans lequel l'un des au moins deux coprocesseurs est un processeur émetteur et l'autre des au moins deux coprocesseurs est un processeur récepteur et l'environnement informatique unique comprend une mémoire accessible aux au moins deux coprocesseurs, le procédé comportant

la détermination (14), dans le processeur émetteur, de la taille des données à communiquer,

si la taille des données est inférieure ou égale (36) à une valeur seuil prédéterminée, l'ajout (15) des données dans un message et la transmission (30) du message au processeur récepteur, et

si la taille des données est supérieure (34) à la valeur seuil prédéterminée, la transmission (24, 30) d'un pointeur au processeur récepteur, le pointeur désignant un emplacement des données dans la mémoire.

2. Procédé selon la revendication 1, dans lequel l'étape de transmission d'un pointeur au processeur récepteur est précédée d'une étape (18) de transfert des données à un emplacement dans la mémoire correspondant au pointeur si les données se sont pas déjà (16, 40) à l'emplacement.

3. Procédé selon la revendication 1, dans lequel l'étape (24, 30) de transmission d'un pointeur au processeur récepteur comprend l'étape (22) de transmission d'un identifiant exclusif au processeur récepteur (22), l'identifiant exclusif étant pré-ajouté aux données dans la mémoire à l'emplacement indiqué par le pointeur (21).

4. Procédé selon la revendication 3, dans lequel l'identifiant exclusif est un horodatage.

5. Procédé selon la revendication 1, dans lequel la valeur seuil prédéterminée est codée dans un fichier de configuration.

5 6. Procédé selon la revendication 1, dans lequel la valeur seuil prédéterminée repose sur le délai nécessaire à la transmission d'un message du processeur émetteur au processeur récepteur.

10 7. Procédé selon la revendication 1, dans lequel le processeur émetteur transmet le message à plus d'un processeur récepteur.

1/1

