

19 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

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

2 870 367

21 N° d'enregistrement national : 05 02796

51 Int Cl⁷ : G 06 F 12/16, G 06 F 13/16

12

DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 22.03.05.

30 Priorité : 22.03.04 US 10805811.

43 Date de mise à la disposition du public de la demande : 18.11.05 Bulletin 05/46.

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) : DELL PRODUCTS, L.P. — US.

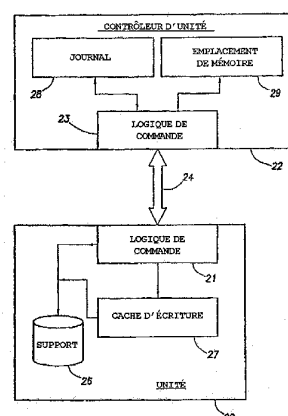
72 Inventeur(s) : MARKS KEVIN T, ALI AHMAD AJ et CLAUSEN ROBERT.

73 Titulaire(s) :

74 Mandataire(s) : DUPUIS LATOUR DOMINIQUE.

54 SYSTEME ET PROCEDE DE RESTAURATION DE DISQUE APRES INCIDENT DISQUE.

57 La présente invention concerne un système et un procédé pour reconstituer une unité (20) dans lesquels le cache d'écriture (27) de l'unité (20) est activé et un enregistrement séparé est maintenu pour les commandes transmises à l'unité (20). Sur une base périodique, l'unité (20) vide les données se trouvant dans son cache (27), et l'historique de commandes est effacé. Lorsque l'unité (20) est reconstituée avec succès, le cache (27) de l'unité (20) est désactivé.



FR 2 870 367 - A1



La présente description concerne d'une manière générale le domaine des systèmes de mémorisation de données, et plus particulièrement, un système et un procédé pour reconstituer une unité à l'aide d'une mémoire cache d'écriture activée.

Au fur et à mesure que la valeur et l'utilisation des informations continuent à croître, les personnes et le personnel commercial cherchent des procédés supplémentaires pour traiter et mémoriser les informations. L'une des options mises à la disposition des utilisateurs concerne les systèmes de traitement d'informations. Un système de traitement d'informations, d'une manière générale, traite, compile, mémorise et/ou communique des informations ou des données à des fins commerciales, personnelles ou à d'autres fins, permettant ainsi aux utilisateurs d'exploiter la valeur des informations. Du fait que les besoins et impératifs en termes de technologie et de traitement d'informations varient d'un utilisateur à un autre ou d'une application à une autre, les systèmes de traitement d'informations peuvent également varier en tenant compte du type d'informations qui sont traitées, de la manière par laquelle les informations sont traitées, de la quantité d'informations traitées, mémorisées ou communiquées, et du degré de rapidité et d'efficacité avec lesquelles les informations peuvent être traitées, mémorisées, ou communiquées. Les variations dans les systèmes de traitement d'informations permettent aux systèmes de traitement d'informations d'être généraux ou configurés pour un utilisateur spécifique ou une utilisation spécifique, telle qu'un traitement de transactions financières, les réservations de vol par avion, la mémorisation de données d'entreprise, ou les communications globales. De plus, les systèmes de traitement d'informations peuvent inclure une variété de composants matériels et logiciels pouvant être configurés pour traiter, mémo-

riser et communiquer des informations et peuvent inclure un ou plusieurs systèmes informatiques, systèmes de mémorisation de données, et systèmes de gestion de réseau, par exemple un ordinateur, un poste de travail informatique individuel, un ordinateur portable, un serveur informatique, un serveur d'impression, un routeur de réseau, un concentrateur de réseau, un commutateur de réseau, un réseau de disques de réseau de mémorisation, un système de réseau redondant de disques indépendants ("RAID") et un commutateur de télécommunications.

Les systèmes de traitement d'informations incluent souvent une ou plusieurs unités groupées en un réseau d'unités. De nombreuses unités incluent un cache d'écriture associé qui peut être sélectivement activé ou désactivé. En ce qui concerne les unités qui incluent un cache d'écriture activé, il existe quelquefois un retard entre l'instant auquel l'unité notifie au contrôleur d'unité que l'écriture s'est déroulée avec succès et l'instant auquel les données sont écrites sur le support de mémorisation de l'unité. Une unité munie d'un cache d'écriture activé va souvent transférer ou mémoriser des données d'écriture dans le cache d'écriture de l'unité. Une fois que les données d'écriture sont transférées avec succès vers le cache, l'unité va transmettre une notification au contrôleur d'unité pour indiquer que la commande d'écriture s'est exécutée avec succès. Cette notification est transmise par l'unité au contrôleur d'unité même si les données de commande d'écriture n'ont pas encore été écrites dans le support permanent et non-volatile de l'unité. La mémorisation des données de commande d'écriture dans le cache d'écriture permet au circuit de commande du contrôleur d'écriture d'optimiser l'ordre dans lequel les informations sont écrites sur le support de l'unité.

Cette méthodologie est problématique, cependant, dans le cas d'une commande d'écriture qui a été écrite avec succès dans le cache de l'unité, mais qui n'a pas encore été écrite avec succès sur le support non-volatile de l'unité. Dans ce cas, l'unité a notifié au contrôleur d'unité que la commande d'écriture était réussie. Si une unité devient défectueuse (par exemple l'unité perd de la puissance ou est relancée ou le cache d'écriture devient corrompu) dans ces circonstances, les données mémorisées dans le cache d'écriture peuvent être perdues et jamais écrites sur le disque. Le contrôleur d'unité, cependant, ignore la perte de données du fait que le circuit de commande de l'unité a indiqué que la commande d'écriture a été exécutée. Le contrôleur d'unité et l'unité ne sont pas synchronisés et les données dans le cache d'écriture qui n'ont pas été écrites sur disque sont perdues. Pour empêcher cet événement d'anomalie de se produire, les caches d'écriture des unités d'un système RAID sont souvent désactivés. Lorsque le cache d'écriture d'une unité a été désactivé, l'unité ne peut pas temporairement mémoriser les données d'écriture sur un cache, forçant ainsi l'unité à écrire les données directement sur le support de mémorisation non-volatile de l'unité. Dans ce scénario, l'unité ne notifie pas au contrôleur d'unité une écriture réussie jusqu'à ce que l'unité ait écrit les données d'écriture sur son support de mémorisation non-volatile. Lorsque le cache d'écriture d'une unité est désactivé, cependant, ses performances peuvent être négativement affectées, du fait que le circuit de commande de l'unité ne peut pas optimiser le transfert de données à partir du cache vers le support permanent de l'unité.

Les réseaux de mémorisation RAID sont caractérisés par la capacité à restaurer ou reconstituer les informations sur une unité après une anomalie. Par exemple, dans un réseau RAID 5, des informations de parité sont

mémorisées sur les unités du réseau. Si l'une des unités devient défectueuse, elle est reconstituée sur la base des informations de parité mémorisées sur les autres unités du réseau. Lorsque la capacité du support des unités augmente, la restauration d'une unité dure plus longtemps du fait de la quantité accrue d'informations qui doivent être restaurées. Les temps de reconstitution de plusieurs heures ou d'un ou de plusieurs jours ne sont pas non-communs pour des unités ayant des supports capables de mémoriser des dizaines ou des centaines de gigaoctets de données. Pendant qu'une unité du réseau est en cours de reconstitution, de nombreux réseaux RAID s'exécutent dans un mode dégradé. Dans un mode dégradé, les performances du réseau peuvent souffrir du fait des ressources dédiées à reconstituer l'unité. De plus, si le cache de l'unité en cours de reconstitution est désactivé durant la période de reconstitution, le temps nécessaire pour reconstituer l'unité peut être plus long par comparaison au temps nécessaire pour reconstituer une unité qui a un cache d'écriture activé. De plus, dans de nombreux niveaux RAID, si une seconde unité devient défectueuse pendant que le réseau est en mode dégradé, le réseau sera perdu.

Selon la présente description, on décrit un système et un procédé pour reconstituer une unité dans un réseau d'unités. Le cache d'écriture de l'unité en cours de reconstitution est activé. Durant le processus de reconstitution, des commandes envoyées à l'unité sont également enregistrées dans un journal associé au contrôleur d'unité. Une commande de synchronisation est périodiquement envoyée à l'unité. En réponse à la commande de synchronisation, l'unité écrit toutes les données se trouvant dans le cache d'écriture, qui n'ont pas été écrites sur le support non-volatile, sur le support. Après synchronisation, le journal est effacé, lorsque les comman-

des d'écriture ont été exécutées à l'égard de la mémoire non-volatile des réseaux d'unités.

Un avantage du système et procédé décrits ici concerne le temps de reconstitution plus court des unités défectueuses. L'activation du cache d'écriture pour l'unité en cours de restauration permet à une logique de commande de l'unité d'optimiser l'ordre dans lequel les commandes sont écrites sur le support. L'activation du cache d'écriture pour l'unité en cours de restauration permet également à de multiples commandes d'écriture d'être envoyées au disque avant que les données associées aux commandes d'écriture soient écrites sur le support de l'unité. Du fait que le cache d'écriture est activé, l'unité peut être reconstituée plus rapidement qu'une unité dont le cache d'écriture était désactivé durant le processus de reconstitution. Un autre avantage du système et du procédé décrits ici est que la technique décrite inclut un moyen qui envisage la possibilité qu'une unité en cours de reconstitution puisse souffrir d'une anomalie système pendant que le cache d'écriture est activé et durant le processus de reconstitution. Pour compenser cette possibilité, les commandes sont écrites dans un journal et périodiquement synchronisées vis-à-vis de l'unité. Le processus de synchronisation implique le fait de forcer toutes les données se trouvant dans le cache de l'unité qui sont associées aux commandes d'écriture à être écrites sur le support de mémorisation de l'unité. La mémorisation d'un sous-ensemble de commandes potentiellement non-synchronisées dans le journal du contrôleur d'unité évite la possibilité d'avoir à recommencer le processus de reconstitution depuis le début. Au lieu de cela, le processus de reconstitution peut être relancé à partir de ces commandes mémorisées dans le journal dans le cas d'une anomalie durant le processus de reconstitution (par exemple, une perte de puissance). En résultat, le proces-

sus de reconstitution décrit ici inclut un mécanisme de sécurité qui protège vis-à-vis d'une anomalie ultérieure tout en n'affectant pas négativement les performances du processus de reconstitution. D'autres avantages techniques vont mieux apparaître à l'homme du métier en vue de la description et des dessins qui vont suivre.

Une compréhension plus complète de la présente description et des avantages de celle-ci peut être acquise en se reportant à la description qui va suivre faite en référence aux dessins annexés, sur lesquels les références numériques analogues indiquent des caractéristiques analogues, et sur lesquels :

- la figure 1 est un schéma d'un réseau de mémorisation redondant,

- la figure 2 est une unité d'un réseau de mémorisation et d'un contrôleur d'unité associé, et

- la figure 3 est un ordinogramme d'un procédé de reconstitution d'une unité d'un réseau d'unités.

Dans le but de la présente description, un système de traitement d'informations peut inclure tout dispositif ou ensemble de dispositifs opérationnels pour calculer, classer, traiter, transmettre, recevoir, récupérer, émettre, commuter, mémoriser, afficher, manifester, détecter, enregistrer, reproduire, gérer, ou utiliser toute forme d'informations, intelligence ou de données à des fins commerciales, scientifiques, de commande, ou autre. Par exemple, un système de traitement d'informations peut être un ordinateur individuel, un dispositif de mémorisation de réseau, ou tout autre dispositif approprié et peut varier en taille, forme, performance, fonctionnalité et prix. Le système de traitement d'informations peut inclure une mémoire à accès direct (RAM), une ou plusieurs ressources de traitement telles qu'une unité centrale de traitement (CPU) ou une logique de com-

mande matérielle ou logicielle, une mémoire à lecture seule (ROM), et/ou d'autres types de mémoire non-volatile. Des composants supplémentaires du système de traitement d'informations peuvent inclure une ou plusieurs unités de disques, un ou plusieurs ports réseau pour communiquer avec des dispositifs externes ainsi que divers dispositifs d'entrée et de sortie (E/S), tels qu'un clavier, une souris, et un afficheur vidéo. Le système de traitement d'informations peut également inclure un ou plusieurs bus fonctionnels pour transmettre des communications entre les divers composants matériels.

La figure 1 est un schéma d'un réseau de mémorisation RAID de Niveau 5, qui est indiqué d'une manière générale par la référence numérique 10. Le réseau de mémorisation 10 inclut cinq unités, qui sont quelquefois appelées disques ou volumes. Chacune des quatre unités dans l'exemple de la figure 1 inclut huit bandes ou rangées de données, désignées par Bande 0 à Bande 7. Il doit être reconnu que la configuration du réseau RAID de la figure 1 est simplement une illustration d'un réseau RAID, et qu'un réseau RAID peut être configuré de manière à avoir plus ou moins d'unités ayant plus ou moins de bandes ou rangées. En faisant référence à la Bande 0, des données sont mémorisées dans Unité A, Unité B, et Unité C. Les bits de parité pour Bande 0, qui sont le résultat d'une opération OU-exclusif effectuée sur le contenu de la Bande 0 dans Unité A, Unité B, et Unité C, sont mémorisés dans l'Unité D et appelés P_0 . A titre de second exemple de la structure de données du réseau RAID 10, en faisant référence à la Bande 7, des données sont mémorisées dans Unité B, Unité C, et Unité D. Les bits de parité pour la Bande 7, qui sont le résultat d'une opération OU-exclusif effectuée sur le contenu de la Bande 7 dans Unité B, Unité C, et Unité D, sont mémorisés dans Unité A et appelés P_7 . Si, par exemple, Unité C devenait défec-

tueuse ou sinon était identifiée comme étant une unité dégradée, les données dans chaque bande de l'Unité C sont reconstituées à l'aide des données dans les trois autres unités du réseau RAID 10. Comme représenté sur la figure 1, chacune des unités du réseau de mémorisation est couplée à un hôte. Le réseau RAID 5 10, est un exemple de niveau de RAID à tolérance de panne dans lequel une anomalie d'unité unique peut être supportée et l'unité défectueuse peut être reconstituée. D'autres niveaux RAID à tolérance de panne incluent 1, 4, 5, 6, 10, et 0+1.

Chaque unité d'un réseau de mémorisation communique avec un contrôleur d'unité et est commandé par celui-ci. La figure 2 est un schéma d'une unité 20 et d'un contrôleur d'unité 22. L'unité 20 et le contrôleur d'unité 22 sont couplés l'un à l'autre via un canal 24. Le canal 24 peut fonctionner conformément à un nombre quelconque de protocoles de communications, incluant un bus ou une liaison de communications de type à Interface Système pour Petits Ordinateurs (SCSI) parallèle ou série, un bus ou une liaison de communications de type à Adaptateur de Terminal Analogique (ATA) parallèle ou série, une liaison de communications Fibre Channel, ou une liaison de communications sans fil. L'unité 20 inclut un élément de mémorisation de support 26, une logique de commande 21, et un cache d'écriture 27. La mémorisation de support 26 peut être constituée de quelconques supports adaptés pour mémoriser des informations, incluant des supports magnétiques ou des supports optiques. D'une manière similaire, le cache d'écriture 27 peut être constitué d'un quelconque mécanisme de mémorisation adapté pour mémoriser des informations. Le cache d'écriture 27 est typiquement une mémoire volatile qui est fonctionnelle pour fournir un accès plus rapide à son contenu par comparaison au support de mémorisation 26, qui est typiquement non-volatil.

La logique de commande 21 de l'unité 20 répond à des commandes provenant du contrôleur d'unité 22 et traite celles-ci, incluant des commandes pour lire des données à partir de l'unité 20 ou écrire des données dans celle-ci, activer ou désactiver le cache d'écriture 27, et forcer toutes les informations dans l'unité d'écriture 27 à être écrites sur le support de mémorisation 26. La logique de commande 21 transmet également des données et des informations de notification au contrôleur d'unité 22. De plus, la logique de commande 21 commande le support de mémorisation 26 et le cache d'écriture 27. Le contrôleur d'unité 22 inclut un élément de logique de commande 23 qui communique avec un journal 28 et commande celui-ci et un emplacement de mémoire 29 du contrôleur d'unité. Des exemples de contrôleur d'unité 22 selon la présente description incluent un contrôleur RAID, un contrôleur ATA/SATA ou SCSI de niveau inférieur, ou les deux.

Le journal 28 est un emplacement de mémorisation d'informations qui mémorise un historique de commandes d'écriture envoyées à l'unité 20. Le contenu du journal 28 peut être contrôlé par la logique de commande 23. La logique de commande 23 peut émettre une commande pour amener le journal 28 à écraser ou vider son contenu. Le journal 28 est de préférence non-volatile de nature de sorte que le contenu du journal 28 sera préservé si se produit une perte soudaine de puissance de l'unité ou du réseau environnant. Le contenu du journal 28 est une liste des plus récentes commandes envoyées à l'unité 20, ainsi qu'un compte de commandes associé à chacune des commandes citées dans la liste. L'emplacement de mémoire 29 peut être un quelconque emplacement de mémoire accessible à la logique de commande 23. Un compte de commandes courant est mémorisé dans l'emplacement de mémoire 29. La valeur du compte de commandes correspond à la commande

d'écriture la plus récente envoyée à l'unité 20. Le compte de commandes dans l'emplacement de mémoire 29 peut être manipulé par la logique de commande 23. En particulier, la logique de commande 23 peut mémoriser un nouveau
5 compte de commandes dans l'emplacement de mémoire 29, récupérer le compte de commandes mémorisé, ou effacer ou réinitialiser le compte de commandes. L'emplacement de mémoire 29 peut comporter une mémoire non-volatile de sorte que le compte de commandes sera préservé si se pro-
10 duit une perte de puissance de l'unité ou du réseau.

La figure 3 est un ordinogramme d'un procédé permettant de reconstituer une unité d'un réseau d'unités. Les étapes de la figure 3 sont exécutées lorsqu'une unité du réseau d'unités doit être reconstituée. Une unité peut
15 nécessiter une reconstitution pour de nombreuses raisons. Les données sur l'unité peuvent être corrompues ou l'unité peut être une unité qui a été ajoutée au réseau d'unités. L'unité peut être reconstituée conformément à une reconstitution automatisée qui identifie les conditions
20 qui commandent la reconstitution de l'unité. En variante, une unité peut être reconstituée après l'ordre manuel d'un administrateur système. A l'étape 30, le cache d'écriture de l'unité est activé, permettant ainsi à l'unité d'utiliser son cache d'écriture pour faciliter
25 l'optimisation des écritures sur le support de l'unité. Le cache d'écriture de l'unité 20 est typiquement activé par une commande envoyée par le contrôleur d'unité 22 à l'unité 20. A titre d'exemple, si l'unité 20 est une unité SCSI, le contrôleur d'unité 22 transmet une commande
30 SELECTION DE MODE à une Page de Mode Cache dans la logique de commande 21 de l'unité 20 et affecte un 1 au bit d'Activation de Cache d'Ecriture dans l'unité. Dans un environnement ATA ou SATA, le contrôleur d'unité envoie une commande ETABLIR CARACTERISTIQUES à l'aide d'un code
35 de sous-commande pour activer le cache d'écriture.

A l'étape 32, les commandes émises à l'unité sont reçues dans le contrôleur d'unité 22. Ces commandes incluent les commandes d'écriture nécessaires pour la reconstitution de l'unité. Lorsque des parties de l'unité sont reconstituées, les commandes peuvent inclure des commandes de charge active pour écrire de nouvelles données dans les parties reconstituées de l'unité. A l'étape 32, la commande reçue est enregistrée dans le journal 28. Lorsque chaque commande est reçue, une liste courante des commandes d'écriture est consignée dans le journal 28. Après l'enregistrement de la commande dans le journal 28, le compte de commandes est incrémenté dans la mémoire 29. A l'étape 38, la commande reçue est transmise à l'unité. Comme on peut le constater par l'ordre des étapes 32 à 38, la commande est enregistrée dans le journal avant que la commande soit transmise à l'unité. De cette manière, un enregistrement de chaque commande est réalisé avant que la commande soit délivrée à l'unité. Lorsque la commande est reçue dans l'unité, la commande, et son contenu, peuvent être mémorisés dans le cache ou mémorisés dans le support de mémorisation de l'unité. Du fait que le cache d'écriture est activé sur l'unité, l'unité peut choisir une méthodologie pour traiter la commande qui réalise l'utilisation la plus efficace du cache de l'unité. Une fois que l'unité a traité avec succès la commande, que ce soit par l'intermédiaire d'un placement de la commande dans le cache ou son support de mémorisation, l'unité va émettre une commande de notification au contrôleur pour indiquer que la commande a été traitée avec succès par l'unité.

A l'étape 40, après la transmission par l'unité de la notification afin d'indiquer une réception réussie de la commande, il est déterminé si le processus de reconstitution est achevé. Si le processus de reconstitution de l'unité n'est pas achevé, il est déterminé à

l'étape 42 si le compte de commandes a atteint une valeur maximale prédéterminée. Une fois que le compte de commandes atteint la valeur maximale prédéterminée, c'est-à-dire qu'une fois que le nombre maximal prédéterminé de commandes sont enregistrées dans le journal, une série d'étapes sont exécutées pour synchroniser le journal avec le contenu du support de mémorisation de l'unité. La valeur maximale prédéterminée peut être un nombre quelconque approprié qui n'est pas supérieur au nombre de commandes qui peuvent être mémorisées dans le journal. La valeur maximale prédéterminée ne doit pas être établie à une valeur tellement faible que le journal et le support de mémorisation sont synchronisés tellement souvent que les étapes de synchronisation interfèrent avec l'exécution efficace du processus de reconstitution. La valeur prédéterminée ne doit pas être établie à une valeur tellement élevée qu'un nombre excessif de commandes sont mémorisées dans le journal. Le fait d'avoir un nombre excessif de commandes dans le journal est contraire au but de suivi d'un nombre limité de commandes destinées à être réexécutées en cas de perte ultérieure de l'unité en cours de reconstitution. Si la valeur prédéterminée du compte de commandes n'a pas été atteinte, l'ordinogramme se poursuit à l'étape 32 par la réception de la commande suivante dans le contrôleur d'unité.

S'il est déterminé à l'étape 42 que la valeur prédéterminée du compte de commandes a été atteinte, les commandes enregistrées dans le journal sont synchronisées avec le contenu du support de mémorisation de l'unité. Le contrôleur d'unité envoie une commande à l'étape 44 pour imposer que toutes les données mémorisées dans le cache qui sont associées aux commandes d'écriture soient écrites sur le support. La commande sera une commande reconnue par la logique de commande de l'unité. Dans un environnement SCSI, le contrôleur d'unité envoie une commande

SYNCHRONISER. Dans un environnement ATA ou SATA, le contrôleur d'unité envoie une commande VIDER. Cette commande amène toutes les données associées aux commandes d'écriture dans le cache à être vidées sur le support de
5 mémorisation de l'unité. Après cette étape, le cache de l'unité ne contient plus de commandes d'écriture qui n'ont pas été écrites dans le support de mémorisation de l'unité. A l'étape 46, après que l'étape 44 ait été exécutée avec succès, le journal est effacé et le compte de
10 commandes est remis à zéro. L'ordinogramme continue à l'étape 32 par la réception dans le contrôleur d'unité de commandes supplémentaires émises à l'unité.

S'il est déterminé à l'étape 40 que la reconstitution est achevée, l'ordinogramme continue à l'étape 48
15 par l'envoi d'une commande pour forcer toutes les commandes d'écriture provenant du cache pour que ces commandes d'écriture puissent être exécutées par rapport au support de mémorisation de l'unité. On va reconnaître que l'étape 48 est identique à l'étape 44. A l'étape 50, après l'exécution réussie de l'étape 48, le journal est effacé et le
20 compte de commandes est remis à zéro. On va reconnaître que l'étape 50 est identique à l'étape 46. A l'étape 52, le cache d'écriture de l'unité est désactivé. Dans un environnement SCSI, le contrôleur de commande envoie une
25 commande de SELECTION DE MODE à la Page de Mode Cache dans la logique de commande 21 de l'unité 22 et attribue au bit d'Activation de Cache d'Ecriture de l'unité la valeur 0. Dans un environnement ATA ou SATA, le contrôleur d'unité envoie une commande ETABLIR CARACTERISTIQUES à
30 l'aide d'un code de sous-commande pour désactiver le cache d'écriture. La désactivation du cache d'écriture met l'unité dans un état dans lequel les commandes d'écriture envoyées à l'unité de mémorisation ne peuvent pas être mises en cache dans l'unité, garantissant ainsi que tou-

tes les écritures dans l'unité sont écrites dans le support de mémorisation non-volatile de l'unité.

Le système et le procédé décrits ici permettent d'optimiser la reconstitution à cache d'écriture activé
5 d'une unité tout en protégeant l'unité vis-à-vis d'une anomalie ultérieure durant le processus de reconstitution. L'enregistrement d'une commande d'écriture dans le journal fournit une ressource pour donner la liste des commandes d'écriture les plus récemment envoyées. Si le
10 cache d'écriture d'une unité perd de la puissance durant une reconstitution de l'unité, le journal va contenir une liste des commandes qui peuvent ne pas avoir été écrites sur la mémoire non-volatile de l'unité. Ainsi, le contenu du journal peut être utilisé en tant que ressource pour
15 éviter la nécessité de recommencer le processus de reconstitution en cas d'anomalie de perte de puissance du cache de l'unité en cours de reconstitution.

On doit comprendre que le système et le procédé décrits ici ne sont pas limités à l'architecture précise
20 décrite sur les dessins de la présente description. Au lieu de cela, le système et le procédé de la présente invention peuvent être utilisés dans une quelconque architecture de système informatique adaptée qui implique l'utilisation d'une alimentation en énergie redondante.
25 On doit également comprendre que le système et le procédé décrits ici ne sont pas limités dans leur application à un processeur spécifique ou une famille de processeurs ou à l'application d'une commande spécifique au processeur. Au lieu de cela, le système et le procédé décrits ici
30 peuvent être utilisés avec tout processeur capable de moduler sa consommation d'énergie par l'intermédiaire de la modulation d'une ou de plusieurs de ses horloges internes. Bien que la présente invention ait été décrite en détail, on doit comprendre que divers changements, substitutions,
35 et modifications peuvent être réalisés sur

celle-ci sans s'écarter du domaine et de la portée de la présente invention.

REVENDICATIONS

1. Procédé pour traiter les commandes de reconstitution envoyées à une unité (20), l'unité (20) ayant une mémoire non-volatile et un cache (27), caractérisé en ce qu'il comporte les étapes consistant à :
- 5 activer le cache (27) de l'unité (20),
 enregistrer dans un premier emplacement de mémoire (29) les commandes de reconstitution envoyées à l'unité (20), et
- 10 amener périodiquement l'unité (20) à vider les données mises en cache associées aux commandes de reconstitution dans sa mémoire non-volatile,
- dans lequel chaque commande envoyée à l'unité (20) est au moins temporairement enregistrée dans l'emplacement de mémoire (29) durant la période où le cache
- 15 (27) de l'unité (20) est activé.
2. Procédé pour traiter les commandes de reconstitution envoyées à une unité (20) selon la revendication 1, caractérisé en ce qu'il comporte de plus l'étape consistant à désactiver le cache (27) de l'unité (20) après la reconstitution réussie de l'unité (20).
- 20 3. Procédé pour traiter les commandes de reconstitution envoyées à une unité (20) selon la revendication 2, caractérisé en ce que l'étape consistant à amener l'unité (20) à vider les données mises en cache dans sa mémoire non-volatile comporte les étapes consistant à :
- 25 maintenir un comptage du nombre de commandes mémorisées dans le premier emplacement de mémoire (29), et
- 30 amener l'unité (20) à vider les données dans sa mémoire non-volatile lorsque le comptage du nombre de commandes mémorisées dans le premier emplacement de mémoire (29) atteint un seuil prédéterminé.
- 35

4. Procédé pour traiter les commandes de reconstitution envoyées à une unité (20) selon la revendication 3, caractérisé en ce qu'il comporte en outre l'étape consistant à effacer le premier emplacement de mémoire (29) et le compte après le vidage réussi des données à partir du cache (27) vers la mémoire non-volatile.

5. Réseau de mémorisation (10), caractérisé en ce qu'il comporte :

10 de multiples unités (20), dans lesquelles chaque unité (20) comporte,

un support de mémorisation non-volatile (26) pour mémoriser des données, et

un cache d'écriture (27) pour mettre en cache des données associées à des commandes d'écriture reçues par l'unité (20), et

15 des contrôleurs d'unité (22), où chaque contrôleur d'unité (22) est associé à une unité (20) du réseau de mémorisation (10) et est couplé à celle-ci, où chaque contrôleur d'unité (22) comporte une première mémoire, où la première mémoire est fonctionnelle pour mémoriser un historique (28) de commandes d'écriture transmises depuis chaque contrôleur d'unité (22) à son unité (20) associée, et

20 dans lequel chaque contrôleur d'unité (22) est fonctionnel pour traiter la reconstitution de son unité (20) associée en :

activant le cache d'écriture (27) de l'unité (20),

30 enregistrant chaque commande d'écriture envoyée à l'unité (20) dans la première mémoire,

amenant périodiquement l'unité (20) à vider les données dans le cache d'écriture (27) de l'unité (20), et

désactivant le cache d'écriture (27) de l'unité (20) lors de l'exécution réussie de la reconstitution de l'unité (20).

5 6. Réseau de mémorisation (10) selon la revendication 5, caractérisé en ce que chaque contrôleur d'unité (22) est fonctionnel pour effacer sa première mémoire (29) après l'indication du vidage réussi des données du cache d'écriture (27) de l'unité (20) associée.

10 7. Réseau de mémorisation (10) selon la revendication 5, caractérisé en ce que chaque contrôleur d'unité (22) comporte une seconde mémoire pour enregistrer le nombre de commandes mémorisées dans la première mémoire.

15 8. Réseau de mémorisation (10) selon la revendication 7, caractérisé en ce que chaque contrôleur d'unité (22) est fonctionnel pour amener son unité (20) associée à vider les données du cache d'écriture (27) lorsque le
20 nombre de commandes mémorisées dans la première mémoire atteint un seuil prédéterminé.

 9. Réseau de mémorisation (10) selon la revendication 5, caractérisé en ce que la première mémoire est
25 non-volatile.

 10. Procédé pour reconstituer le support de mémorisation (26) d'une unité (20), dans lequel l'unité (20) est en communication avec un cache d'écriture (27) qui
30 peut être sélectivement activé, le procédé étant caractérisé en ce qu'il comporte les étapes consistant à :

 activer le cache d'écriture (27) de l'unité (20),
 transmettre une ou plusieurs commandes à l'unité (20),

écrire la ou les plusieurs commandes sur un journal (28), et

forcer l'unité (20) à vider les données du cache d'écriture (27) dans le support de mémorisation (26).

5

11. Procédé pour reconstituer le support de mémorisation (26) d'une unité (20) selon la revendication 10, caractérisé en ce qu'il comporte en outre l'étape consistant à effacer le journal (28).

10

12. Procédé pour reconstituer le support de mémorisation (26) d'une unité (20) selon la revendication 11, caractérisé en ce qu'il comporte en outre l'étape consistant à désactiver le cache d'écriture (27) de l'unité (20) après la reconstitution réussie du support (26) de l'unité (20).

15

13. Procédé pour reconstituer un support (26) d'une unité (20) selon la revendication 10, caractérisé en ce qu'il comporte les étapes consistant à :

20

délivrer un compte de commandes envoyées à l'unité (20), et

dans lequel l'étape consistant à forcer l'unité (20) à vider les données du cache d'écriture (27) est exécutée une fois que le compte de commandes envoyées à l'unité (20) atteint une valeur prédéterminée.

25

14. Procédé selon la revendication 13, caractérisé en ce qu'il comporte en outre l'étape consistant à effacer le journal (28) après le vidage réussi des données du cache d'écriture (27).

30

15. Procédé selon la revendication 13, caractérisé en ce qu'il comporte en outre l'étape consistant à dé-

sactiver le cache d'écriture (27) après la reconstitution réussie du support (26) de l'unité (20).

5 16. Procédé selon la revendication 10, caractérisé en ce qu'il comporte en outre l'étape consistant à effacer le journal (28) après que l'unité (20) ait vidé toutes les données du cache d'écriture (27) et transmis un message indiquant que les données mises en cache ont été écrites sur le support (26).

10

 17. Procédé selon la revendication 10, caractérisé en ce que le journal (28) est constitué d'une mémoire non-volatile associée à un contrôleur d'unité (22).

15 18. Contrôleur d'unité (22) fonctionnel pour communiquer avec une unité (20) par l'intermédiaire d'un canal de communications (24), caractérisé en ce qu'il comporte :

20 une première mémoire pour enregistrer des commandes transmises depuis le contrôleur d'unité (22) à l'unité (20) durant la période où l'unité (20) est en cours de reconstitution,

 une seconde mémoire pour mémoriser un compte des commandes enregistrées dans la première mémoire,

25 dans lequel, durant la période où l'unité (20) est en cours de reconstitution, le contrôleur d'unité (22) est fonctionnel pour activer le cache d'écriture (27) de l'unité (20) et amener l'unité (20) à vider les données du cache d'écriture (27) de l'unité (20) lorsque
30 le compte des commandes atteint un seuil prédéterminé.

 19. Contrôleur d'unité (22) selon la revendication 18, caractérisé en ce que le contrôleur d'unité (22) est fonctionnel pour désactiver le cache d'écriture (27)

21

de l'unité (20) après la reconstitution réussie de l'unité (20).

5 20. Contrôleur d'unité (22) selon la revendication 19, caractérisé en ce que la première mémoire constitue une mémoire non-volatile.

1/3

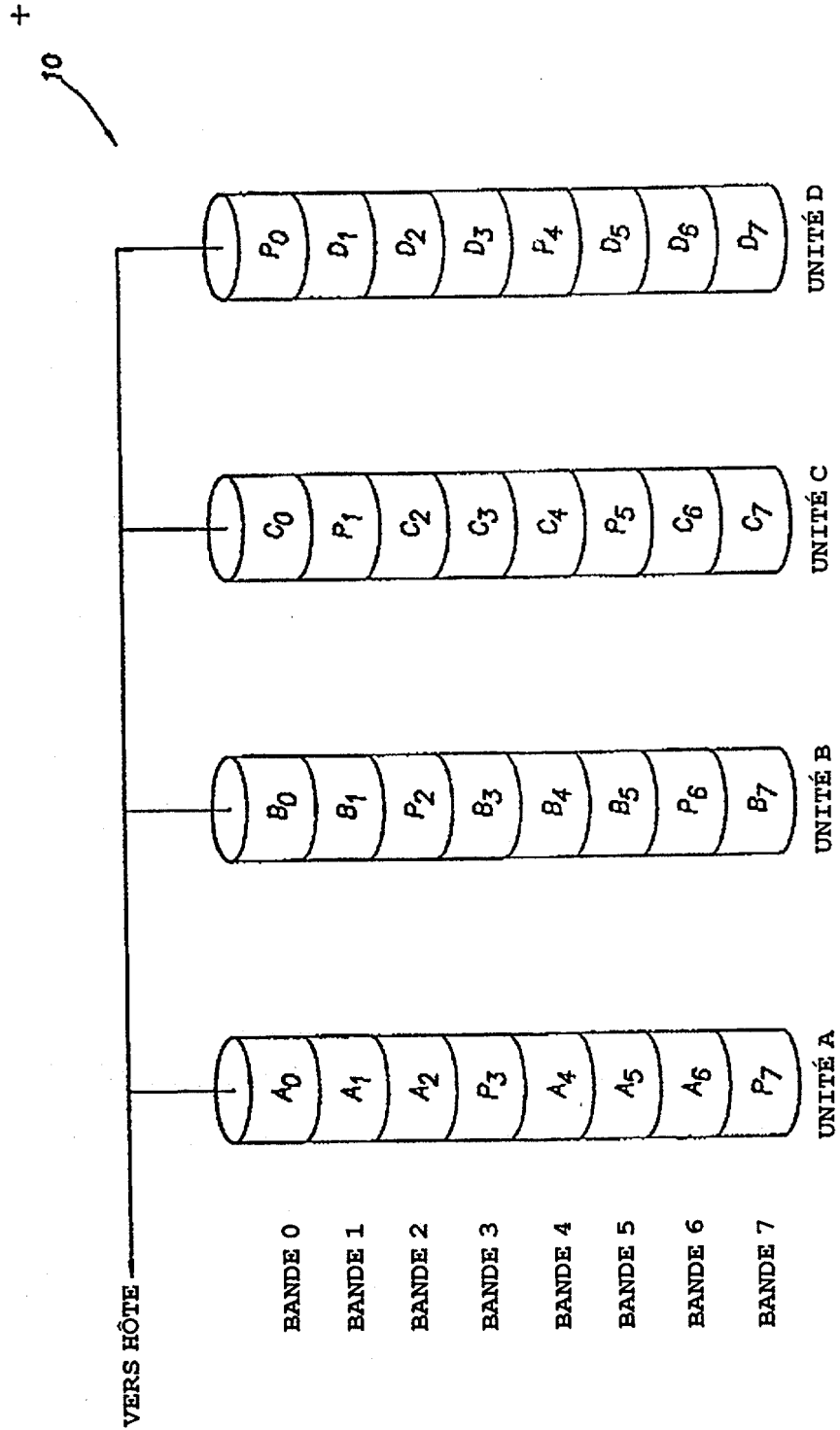


Figure 1

2/3

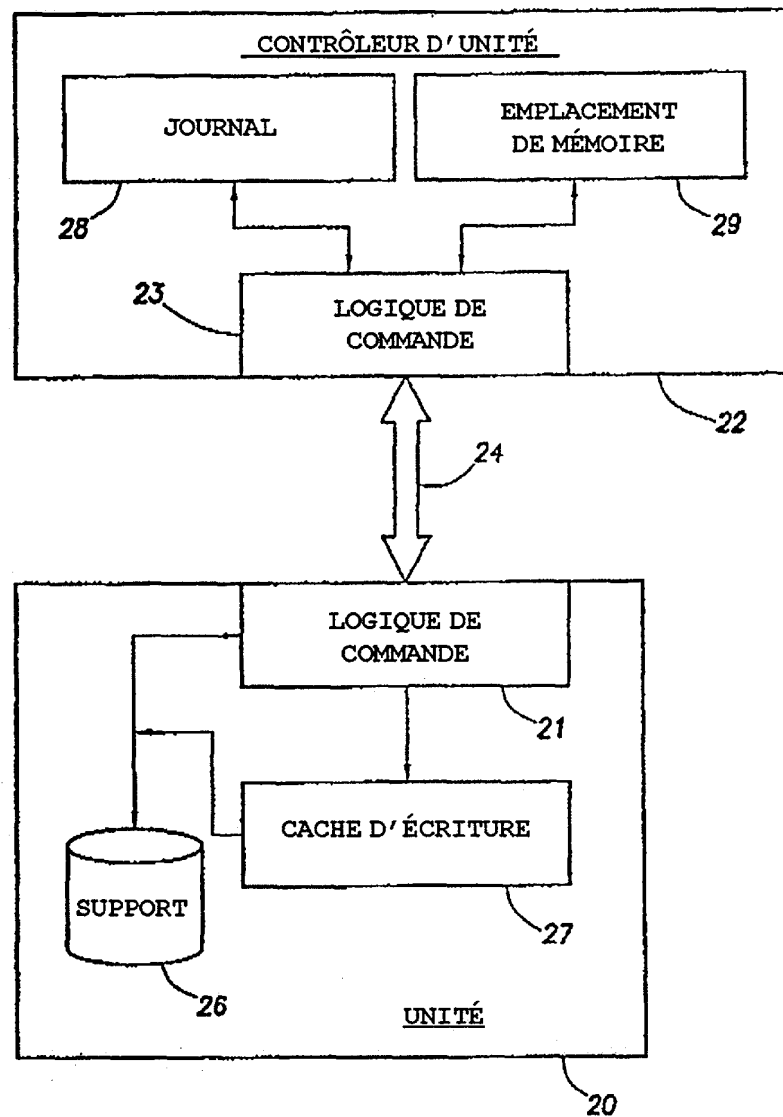


Figure 2

3/3

Figure 3

