

Des grappes de PC pour la réalité virtuelle

Bruno Raffin

Laboratoire Informatique et Distribution
ENSIMAG - Antenne de Montbonnot
ZIRST, 51, avenue Jean Kuntzmann
38330 Montbonnot Saint Martin, France
Bruno.Raffin@imag.fr

20 décembre 2001

Résumé

Les environnements de réalité virtuelle multi-projecteurs nécessitent des calculateurs puissants. Des grappes de PC équipés de cartes graphiques peuvent répondre à ces besoins. Cependant, l'utilisation de composants initialement non destinés à cet usage nécessite des solutions originales pour répondre aux attentes en terme de fonctionnalités, de portabilité, d'extensibilité et de performance. Nous présentons comment la multi-projection, la stéréoscopie et la recomposition d'images peuvent être réalisés sur grappe de PC. Nous présentons et analysons les différentes plateformes logicielles récemment développées qui supportent les grappes de PC. Nous détaillons en particulier les librairies Open Source VR Juggler et Net Juggler.

1 Introduction

Les environnements de visualisation 3D avancés sont de plus en plus utilisés pour la réalité virtuelle, la visualisation de données complexes ou le travail collaboratif. Ces environnements utilisent plusieurs vidéo-projecteurs pour un affichage stéréoscopique de très haute résolution sur une surface importante. Pour des applications interactives, la puissance de calcul nécessaire peut être conséquente. Chaque vidéo-projecteur doit être alimenté avec plusieurs dizaines d'images par seconde. Dans la majorité des cas cette puissance de calcul est fournie par des calculateurs multiprocesseurs et multi-pipelines graphiques de type SGI Onyx.

Une alternative consiste à recourir à une architecture faible coût articulée autour de PC reliés en réseau (grappe de PC) et équipés de cartes graphiques. L'uti-

lisation de grappes de PC pour le calcul scientifique est de plus en plus courante. Leur présence dans le top 500 des machines les plus puissantes n'est plus exceptionnelle [1]. Les performances des cartes graphiques, tirées par l'industrie des jeux vidéos, sont en très forte croissance et concurrencent aujourd'hui des stations graphiques haut de gamme. Cependant, l'utilisation de composants initialement non destinées aux environnements de visualisation 3D avancés nécessite le développement de solutions originales pour répondre aux attentes des utilisateurs en terme de fonctionnalités, de portabilité, d'extensibilité et de performance. L'association de vidéo-projecteurs implique un niveau de synchronisation particulier pour que les images affichées soient cohérentes (swaplock). La stéréoscopie active impose une synchronisation des signaux vidéo (genlock). Composer un signal vidéo à partir de plusieurs autres pour associer la puissance graphique de plusieurs cartes nécessite du matériel spécifique. Pour que l'utilisation de ces architectures ne soit pas réservée à quelques spécialistes et puisse réellement se démocratiser, il est essentiel de développer des environnements logiciels adaptés. Le programmeur ne doit pas avoir à faire apparaître dans son code des spécificités de l'environnement d'exécution, comme le nombre et la disposition des vidéo-projecteurs, si le calculateur est une Onyx ou une grappe de PC. Cette abstraction ne doit pas non plus se faire au détriment des performances. De telles solutions ont récemment été développées. Nous aborderons les différentes approches en analysant à chaque fois les inconvénients et avantages en terme de portabilité et de performance.

2 La multi-projection

L'utilisation de plusieurs vidéo-projecteurs pour former des surfaces d'affichage de grande taille soulève un certain nombre de problèmes. L'alignement des vidéo-projecteurs et la jointure au bordures, la calibration de la luminosité et des couleurs sont des réglages délicats. Dans beaucoup de cas ils sont effectués à la main en attendant des outils de calibration automatiques efficaces. L'affichage doit aussi être cohérent entre l'ensemble des vidéo-projecteurs. L'oeil ne doit pas percevoir de décalage dans le temps lors de l'apparition d'un objet dont l'image serait partagée sur plusieurs vidéo-projecteurs. La synchronisation nécessaire à ce niveau, le swaplock, est bien maîtrisée sur grappe de PC. Elle nécessite une barrière de synchronisation à un niveau de performance que la plupart des réseaux courants atteignent. En générale, la mise en place du swaplock est transparente pour l'utilisateur [8, 15, 5].

3 La stéréoscopie

L'effet stéréoscopique est réalisé en transmettant à chaque oeil de l'utilisateur une image calculée à partir de points de vues légèrement différentes. Essentiellement deux techniques sont utilisées pour séparer les deux images : la stéréoscopie passive et la stéréoscopie active.

3.1 La stéréoscopie passive

La stéréoscopie passive utilise un filtrage par polarisation. Deux vidéo-projecteurs émettent sur la même surface chacun une image destinée à un oeil. Chaque faisceau lumineux est polarisé différemment. L'utilisateur porte des lunettes qui séparent les deux faisceaux suivant leur polarisation. La stéréoscopie passive est simple à mettre en oeuvre sur une grappe de PC, quels que soient les cartes graphiques et l'OS utilisé. Il suffit d'associer un PC par vidéo-projecteur, chaque PC ayant à sa charge le calcul de la vue pour l'oeil droit ou l'oeil gauche. Cette solution a l'inconvénient de doubler le matériel (PC et vidéo-projecteurs) à moins d'utiliser des solutions spécifiques (vidéo-projecteurs dédiés au passif, séparateur de signaux vidéo oeil gauche et oeil droit pour ne pas doubler les PC). La stéréoscopie passive peut être obtenue avec des projecteurs CRT, LCD ou DLP.

3.2 La stéréoscopie active

La stéréoscopie active est basée sur une alternance dans le temps de l'émission de l'image gauche et de l'image droite. L'utilisateur porte des lunettes à obturateurs. Lorsque l'image gauche est émise, le volet droit des lunettes est fermé et vice-versa. La fréquence d'alternance doit être suffisamment élevée pour un bon effet stéréoscopique, typiquement de l'ordre de 70 Hz. Cette fréquence n'est pas à la portée de vidéo-projecteurs LCD. Les configurations en stéréoscopie active utilisent donc essentiellement des projecteurs CRT. L'alternance a lieu à chaque nouveau balayage. La stéréoscopie active nécessite un support logiciel spécifique le plus souvent intégré dans les drivers des cartes graphiques. Il est important de vérifier l'existence du support pour la stéréo active lors de l'installation d'une grappe de PC pour la réalité virtuelle. En configuration multi-projecteurs, il est impératif que tous les vidéo-projecteurs affichent l'image correspondant au même oeil en même temps. Cette synchronisation est communément appelée genlock. Certaines cartes graphiques haut de gamme (3Dlabs) peuvent être reliées par un réseau dédié pour le genlock. Une solution logicielle récente, SoftGenLock [3], rend possible la stéréo active et le genlock sous Linux indépendamment des cartes graphiques utilisées ou de l'accessibilité au code source des drivers.

4 La recomposition de signaux vidéo

Pour certaines applications la puissance de rendu graphique nécessaire dépasse les capacités d'une unique carte graphique. Le format de signal vidéo numérique (DVI) et sa disponibilité de plus en plus courante sur les cartes a donné lieu au développement de composants matériels spécifiques pour la recomposition de signaux vidéo [11]. Ce type de boîtier prend en entrée deux ou plus signaux vidéo DVI et les recompose pour former un unique signal. La charge de travail pris en charge par chaque carte se trouve donc diminuée proportionnellement au nombre de cartes associées pour une même sortie vidéo. La recomposition des signaux introduit cependant un retard sur l'affichage de l'image qui peut être préjudiciable dans le cas d'application interactives. Les recherches dans ce domaines s'efforcent de réduire cette latence.

5 Les plateformes logicielles

La démocratisation des grappes de PC pour la réalité virtuelle passe par le développement de plateformes logicielles adaptées. Les éléments importants à considérer sont les suivants :

- Portabilité vers plusieurs OS (Linux, Windows, IRIX)
- Portabilité vers plusieurs architectures de calculateurs (Onyx, grappe de PC, station de travail)
- Portabilité vers plusieurs architectures de visualisation (écran simple, visio-plan, visio-cube, théâtre)
- Performance
- Interopérabilité avec d'autres outils (CAD)
- Utilisation de standards : portage d'applications existantes et pérennité des nouveaux développements.

Plusieurs plateformes proposent des solutions pour les grappes de PC en réalité virtuelle. Nous en étudions les principes en discutant leur positionnement par rapport aux critères énoncés. Les approches sont classées suivant le paradigme de parallélisation adopté.

5.1 Bibliothèques graphiques distribuées

Lors de l'exécution d'une application, des primitives graphiques de bas niveau décrivant la scène sont générées et transmises vers le driver graphique. Le traitement de ces primitives conduit à une image calculée à partir d'un point de vue donné sur la scène 3D.

Sur une grappe de PC, une approche consiste à intercepter ces primitives et à les diffuser aux nœuds (ou PC) de la grappe munis de cartes graphiques. Des points de vue différents et complémentaires sont aussi transmis vers ces nœuds. Ils calculent chacun une image correspondant à une partie de l'image globale affichée par l'ensemble des vidéo-projecteurs. Plusieurs approches sont basées sur le standard OpenGL [9, 12]. L'intérêt premier est la portabilité. Toute application 3D utilisant OpenGL peut être exécutée sur une grappe de PC sans être modifiée, ni même recompilée dans certains cas. Cette solution à l'inconvénient d'être centralisée. L'ensemble des codes menant jusqu'à la génération des primitives graphiques est exécuté sur le même nœud, qui doit aussi émettre des données vers les nœuds graphiques. Pour les applications complexes, la puissance du nœud central peut s'avérer insuffisante. D'autre part, les primitives graphiques sont des informations très détaillées sur la scène (des triangles). Le volume de données à transmettre pour décrire une scène peut donc être extrêmement important. Pour prévenir l'apparition d'un goulot d'étranglement au niveau du réseau, le protocole WireGL implante des techniques de cache et de compression de données. Malgré cela, les performances pour les applications graphiques complexes et changeant rapidement sont limitées par le réseau, même avec des réseaux très haute performance de type Myrinet.

5.2 Graphes de scène distribués

Pour limiter la quantité d'information à transmettre sur le réseau, une autre approche consiste à distribuer une description de la scène de plus haut niveau basée sur un graphe de scène [16, 10]. Le seuil d'apparition du goulot d'étranglement au niveau du réseau est repoussé. Par contre, la portabilité est moindre dans la mesure où il n'existe pas encore de graphe de scène standard.

5.3 Évènements de contrôle distribués

La quantité d'information à transmettre sur le réseau devient indépendante de la complexité graphique de la scène à partir du moment où on transmet non plus des données graphiques, mais des événements de contrôle. L'application est dupliquée sur chaque nœud graphique. Chaque événement qui peut modifier les données de l'application, un clic de souris par exemple, est intercepté et diffusé à l'ensemble des nœuds graphiques. Les images sont ainsi calculées à partir de données cohérentes. La quantité de données communiquées est faible et indépendante de la complexité graphique de la scène. Cette approche est bien adaptée aux applications exigeantes en terme d'interactivité. Elle n'impose pas non plus l'investissement dans un réseau haut débit. Dans bien des cas un réseau Fast Ethernet est suffisant. La portabilité dépend du contexte logiciel dans lequel cette solution est mise en

œuvre [13, 8, 9, 5].

5.3.1 VR Juggler et Net Juggler

VR Juggler [7, 4] est une plateforme logicielle Open Source développée par l'équipe de Carolina Cruz-Neira au VRAC, USA. VR Juggler définit une abstraction de l'environnement de réalité virtuel, tout en assurant un accès direct aux libraires graphiques ou graphes de scènes. Une application VR Juggler peut ainsi être basée sur OpenGL ou Performer [14]. La gestion des périphériques n'apparaît qu'à travers une couche d'abstraction. Elle est indépendante des interfaces physiques (traqueurs, configuration de l'espace de visualisation par exemple). Typiquement, l'application déclare des interfaces abstraites de type bouton ou position. L'application consulte ces interfaces pour décider de modifications éventuelles de la scène. Au moment du lancement dans un environnement matériel donnée, VR Juggler instancie l'application à partir de données incluses dans des fichiers de configuration. Par exemple, ces fichiers contiennent le nombre et la disposition des surfaces de projection. VR Juggler en déduit les différents points de vue à appliquer à l'application. Si une interface de position a été déclarée, les fichiers de configuration doivent informer VR Juggler de l'interface physique à lui associer. Cela peut être un traqueur ou des touches de clavier qui simulent un traqueur. Par ailleurs, VR Juggler autorise la reconfiguration de l'application et l'extraction de données de performance à l'exécution. Il intègre un mode simulant sur un simple PC un environnement de visualisation complexe tel qu'un visio-cube. Seuls les fichiers de configuration et non l'application doivent être modifiés pour activer ce mode. VR Juggler assure la portabilité des applications sur un large éventail de plateformes, SGI Onyx, PC Linux ou Windows et stations de travail, mais pas sur grappes de PC. VR Juggler étant encore jeune (1998) et pas (encore) un standard, l'interopérabilité avec d'autres outils est pour le moment réduite. La communauté VR Juggler est cependant très active et les installations utilisant cet outil de plus en plus nombreuses.

Net Juggler [2, 5, 6], développé au LIFO, Université d'Orléans, porte VR Juggler sur les grappes de PC. Net Juggler duplique l'application sur les nœuds de la grappe, intercepte et distribue les événements de contrôle. Net Juggler supporte aussi le mécanisme de configuration de VR Juggler avec des fichiers de configuration adaptés à la grappe. Par exemple, si un traqueur est utilisé, les fichiers doivent spécifier sur quel nœud de la grappe il est connecté. Net Juggler se charge ensuite de configurer chaque nœud. La reconfiguration de la grappe peut avoir lieu en cours d'exécution. Net Juggler redirige l'ordre de reconfiguration vers le nœud approprié. Ainsi, Net Juggler fournit une image VR Juggler unique de la grappe de PC. L'application VR Juggler n'a pas à être modifiée. L'utilisateur n'a pas à

accéder à chaque nœud individuellement pour le configurer.

6 Conclusion

Les grappes de PC graphiques présentent de multiples avantages. Le coût est faible. La taille de la grappe et la puissance de chaque nœud sont facilement modulables suivant les besoins. La prédominance de l'architecture PC sur le marché favorise une évolution rapide des logiciels et du matériel. Démocratiser leur utilisation pour la réalité virtuelle passe par le développement de plateformes logicielles qui masquent à l'utilisateur la spécificité de l'architecture tout en assurant de bonnes performances. Les solutions logicielles telles que WireGL, Syzygy, VR Juggler et Net Juggler vont dans ce sens. Il est probable que rapidement les grappes de PC deviennent le calculateur standard pour les environnements de réalité virtuelle.

Références

- [1] Top 500 Supercomputers, www.top500.org.
- [2] Net Juggler. netjuggler.sourceforge.net.
- [3] SoftGenLock. softgenlock.sourceforge.net.
- [4] VR Juggler. www.vrjuggler.org.
- [5] J. Allard, V. Gouranton, L. Lecointre, E. Melin, and B. Raffin. Net Juggler: Running VR Juggler with Multiple Displays on a Commodity Component Cluster. In *IEEE VR*, Orlando, USA, March 2002.
- [6] J. Allard, V. G. E. Melin, and B. Raffin. Parallelizing Pre-rendering Computations on a Net Juggler PC Cluster. In *IPTS*, Orlando, USA, March 2002.
- [7] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, and C. Cruz-Neira. VR Juggler: A Virtual Platform for Virtual Reality Application Development. In *IEEE VR 2001*, Yokohama, Japan, March 2001.
- [8] M. Bues, R. Blach, S. Stegmaier, U. Häfner, H. Hoffmann, and F. Haselberger. Towards a Scalable High Performance Application Platform for Immersive Virtual Environments. In J. D. B. Fröhlich and H.-J. Bullinger, editors, *Immersive Projection Technology and Virtual Environments 2001*, pages 165–174, Stuttgart, Germany, May 2001. Springer.
- [9] Y. Chen, H. Chen, D. W. Clark, Z. Liu, G. Wallace, and K. Li. Software Environments for Cluster-based Display Systems. www.cs.princeton.edu/omnimedia/papers.html, 2001.

- [10] G. Hesina, D. Schmalstieg, A. Fuhrmann, and W. Purgathofer. Distributed Open Inventor: A practical Approach to Distributed 3D graphics. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, London, December 1999.
- [11] G. Humphreys, M. Eldridge, I. Buck, G. Stoll, M. Everett, and P. Hanrahan. WireGL: A Scalable Graphics System for Clusters. In *Proceedings of SIGGRAPH 2001*, 2001.
- [12] G. Humphreys and P. Hanrahan. A Distributed Graphics System for Large Tiled Displays. In *Proceedings of IEEE Visualization'99*, 1999.
- [13] D. Pape, C. Cruz-Neira, and M. Czernuszenko. *CAVE User's Guide*. Electronic Visualization Laboratory, University of Illinois at Chicago, 1997.
- [14] J. Rohlf and J. Helman. IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. In *Computer Graphics (SIGGRAPH 94)*, pages 381–394. ACM Press, July 1994.
- [15] R. Samanta, J. Zheng, T. Funkhouser, K. Li, and J. P. Singh. Load Balancing for Multi-Projector Rendering Systems. In ACM, editor, *Proceedings of SIGGRAPH/Eurographics Workshop on Graphics Hardware*, pages 107–144, Los Angeles, USA, August, 1999.
- [16] B. Schaeffer. A Software System for Inexpensive VR via Graphics Clusters. www.isl.uiuc.edu/ClusteredVR/ClusteredVR.htm.