V

Conclusion et perspectives

Dans cette thèse, nous nous sommes intéressés au couplage de codes parallèles pour les applications interactives à grande échelle. Les points suivants ont été abordés :

– Nous avons fourni un travail préliminaire de couplage interactif d'une simulation parallèle avec une visualisation elle aussi parallèle. Deux scénarios ont été étudiés, l'un avec un couplage fort à l'intérieur d'une même boucle de calcul, l'autre mettant en relation deux programmes distincts par un couplage asynchrone.

– Nous avons proposé un nouveau modèle de couplage de code (*FlowVR*), alliant modularité, simplicité, efficacité et extensibilité. Basé sur des composants (*modules*) reliés par un graphe de flux de données, ce modèle sépare la conception des modules de la construction du réseau de l'application. Des objets de filtrages et de synchronisations placés sur ce réseau permettent d'exprimer des schémas de communication collective et des politiques de couplage avancées, respectant les contraintes de cohérences spécifiques aux applications interactives.

– Nous avons présenté une implantation de ce modèle permettant de valider son efficacité au travers de nombreuses applications innovantes et de grande taille, notamment une reconstruction 3D multi-caméras permettant des interactions en temps réel entre le corps de l'utilisateur et les objets et simulations présentes dans l'environnement. Cette application utilise jusqu'à 54 processeurs, 200 processus et 4000 connexions.

– Plusieurs extensions de haut niveau ont été étudiées pour faciliter et standardiser la construction de certaines parties de l'application. La première, *FlowVR Render*, concerne la phase de rendu. En ce basant sur les dernières avancées des cartes graphiques (les shaders) et sur une description modulaire de l'environnement virtuel, nous avons proposé un modèle de rendu parallèle permettant de filtrer et recombiner de manière efficace les données de différents modules afin de fournir aux machines de rendu les informations nécessaires.

– L'efficacité de FlowVR Render a été testée en adaptant l'outil de visualisation scientifique VTK. En comparaison avec l'outil prédominant de rendu parallèle (*Chromium*), FlowVR Render s'est montré performant et permet de mieux passer à l'échelle, à la fois en terme de machines de rendu que de découpage de la description de la scène. FlowVR Render a aussi permis le développement de nouvelles applications exploitant les fonctionnalités avancées des cartes graphiques pour répartir la charge de calcul entre les CPU et les GPU tout en optimisant les transferts sur le réseau (rendu volumique haute qualité, texturage du modèle reconstruit en recombinant les flux vidéos des caméras, vidéos haute-définition sur mur d'image, visualisation interactive de l'état de la grille Grid5000).

– Enfin nous avons exploré la définition d'un modèle de haut niveau pour la boucle de simulation afin d'obtenir un couplage multi-physique gérant les interactions entre de nombreux objets de types différents. La simulation est découpée en modules gérant les mouvements des objets (*animators*) ou calculant les forces d'interactions entre-eux (*interactors*). Un prototype combinant des objets solides, fluides et déformables a été présenté. Alors que la plupart des travaux n'étudient que des environnements impliquant seulement deux types d'objets différents, notre architecture démontre la faisabilité d'interactions plus complexes. De plus, grâce à la parallélisation des calculs et la distribution des tâches, la simulation atteint le temps réel et permet des interactions entre tous les objets et l'utilisateur. L'environnement virtuel ainsi obtenu dépasse les simples collisions rigides classiques et permet d'envisager la construction d'environnements complexes et réalistes.

En ce concentrant sur les applications interactives et les environnements immersifs, les principes et les outils développés l'ont été dans un objectif d'efficacité et de simplicité. Ceci a permis d'éviter la complexité des modèles de couplage classiques (nécessitant de volumineuses spécifications et un important effort d'implantation). Toutefois, de nombreuses fonctionnalités non essentielles (trop complexes ou redondantes) sont par conséquence non supportées. Ainsi, FlowVR ne supporte aucune dynamicité ni résistance aux pannes. De même, FlowVR Render ne supporte pas de niveau de détail dynamique pour les primitives. Cependant, ces lacunes peuvent faire l'objet d'extensions futures intéressantes.

Les applications développées grâce à FlowVR ont démontré de nouvelles interactions complexes exploitant la puissance des plateformes de type grappe. Quelques expérimentations, notamment dans le cadre de la Fête de la Science 2004, ont relié des machines sur des sites distants. L'étape suivante serait le développement d'applications exploitant les nouvelles plateformes de type grille ou intégrant plusieurs environnements immersifs distants de manière collaborative.

Un des problèmes principaux à ce jour lors de la construction d'applications FlowVR évoluées concerne l'allocation des ressources disponibles. Le concepteur du réseau est responsable du placement de chaque module sur les différentes machines. Un mauvais placement peut entraîner un déséquilibre de la charge de travail. Des modèles de performances pourraient être utilisés pour calculer le placement optimal sur des machines données. C'est un problème qui se complexifie rapidement avec la taille de l'application (NP-Complet), avec en plus l'originalité que la latence des calculs est un critère de performance important pour une application interactive. Quelques réponses préliminaires à ces questions ont été apportées lors du stage de DEA de Christophe Sadoine co-encadré avec Denis Tristram.

Ce projet de recherche se poursuit actuellement dans le cadre du projet MOAIS (INRIA, ID-IMAG) par l'intégration de FlowVR et de KAAPI, un outil de parallélisation dynamique basé sur un ordonnancement par vol de tâche et graphe de dépendance de données. L'idée serait d'utiliser KAAPI pour la parallélisation interne de certains modules FlowVR. Il faut alors étudier comment interfacer FlowVR et l'ordonnanceur de KAAPI pour gérer les tâches reliées aux ports de communications : FlowVR doit-il contraindre le placement de ces ports ou au contraire KAAPI peut décider dynamiquement de l'emplacement des données, peut-être en prenant en compte un facteur de coup fourni par FlowVR.

Un autre travail en cours dans le cadre du projet MOVI (INRIA, GRAVIR-IMAG) concerne le développement d'une couche d'abstraction concernant la partie vision des applications, afin de gérer facilement les différents formats d'images et de données extraites, ainsi que fournir une librairie de filtres et modules parallèles implantant chaque étape du traitement des données. Un des objectifs étant de pouvoir extraire différentes informations en fonction des besoins de l'application (reconstruction 3D, champs de distance, mouvements et suivi des membres, actions sémantiques, etc). Ces problématiques sont étudiées par Clément Ménier, co-encadré par Edmond Boyer.

L'objectif à long terme, en continuité avec les travaux de cette thèse, est à mon sens la construction d'environnements virtuels de plus en plus étendus et complexes, permettant de simuler de manière réaliste des lieux comme un bâtiment complet (objets tous dynamiques, éclairage photo-réaliste, robinets fonctionnels, piscine, etc), des phénomènes complexes comme le fonctionnement du corps humain (apprentissage des gestes chirurgicaux), ou tout simplement recréer les conditions pour que l'utilisateur se croie totalement immergé. A quand la grille capable de créer la matrice...

Bibliographie

[1] G. Abram and L. Treinish. An extended data-flow architecture for data analysis and visualization. In *6th IEEE Visualization Conference (VIS '95)*, 1995.

[2] R. Ahrem, M. G. Hackenberg, U. Karabek, P. Post, R. Redler, and J. Roggenbuck. Specification of MpCCI. Technical Report 12, GMD-SCAI, Sankt Augustin, Germany, 2001.

[3] J. Ahrens, C. Law, W. Schroeder, K. Martin, and M. Papka. A parallel approach for efficiently visualizing extremely large. Technical Report LAUR-001630, Los Alamos National Laboratory, 2000.

[4] B. A. Allan, R. C. Armstrong, A. P. Wolfe, and J. Ray. The CCA core specification in a distributed memory SPMD framework. *Concurrency Computation*, 14 :1–23, 2002.

[5] J. Allard, E. Boyer, J.-S. Franco, C. Ménier, and B. Raffin. Marker-less real time 3D modeling for virtual reality. In *Proceedings of the Immersive Projection Technology Workshop*, Ames, Iowa, May 2004.

[6] J. Allard, E. Boyer, C. Ménier, and B. Raffin. Running large VR applications on a PC cluster : the FlowVR experience. In *Proceedings of the IPT / EGVE Workshop*, Aalborg, Denmark, October 2005.

[7] J. Allard, M. C. Cabral, C. Goudeseune, H. Kaczmarski, B. Raffin, B. Schaeffer, L. Soares, and M. K. Zuffo. Commodity Clusters for Immersive Projection Environments. In *ACM SIGGRAPH 03 Course*, San Diego, California, July 2003.

[8] J. Allard, V. Gouranton, G. Lamarque, E. Melin, and B. Raffin. Softgenlock : Active Stereo and Genlock for PC Cluster. In *Proceedings of the Joint IPT/EGVE'03 Workshop*, Zurich, Switzerland, May 2003.

[9] J. Allard, V. Gouranton, L. Lecointre, S. Limet, E. Melin, B. Raffin, and S. Robert. FlowVR : a Middleware for Large Scale Virtual Reality Applications. In *Proceedings of Euro-Par 2004*, Pisa, Italia, August 2004.

[10] 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*, pages 275–276, Orlando, USA, March 2002.

[11] J. Allard, V. Gouranton, E. Melin, and B. Raffin. Parallelizing Pre-rendering Computations on a Net Juggler PC Cluster. In *Immersive Projection Technology Symposium*, Orlando, USA, March 2002.

[12] J. Allard and B. Raffin. A shader-based parallel rendering framework. In *IEEE Visualization Conference*, Minneapolis, USA, October 2005.

[13] J. Allard and B. Raffin. Distributed physical based simulation for large vr application. In *IEEE VR Conference*, Alexandria, USA, January 2006.

[14] J. Allard, B. Raffin, and F. Zara. Coupling parallel simulation and multi-display visualization on a PC cluster. In *Euro-Par 2003*, Klagenfurt, Austria, August 2003.

[15] R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski. Toward a common component architecture for high-performance scientific computing. In *Proceeding of the 8th IEEE International Symposium on High Performance Distributed Computation*, August 1999.

[16] P. Augerat, C. Goudeseune, H. Kaczmarski, B. Raffin, B. Schaeffer, L. Soares, and M. K. Zuffo. Commodity Clusters for Immersive Projection Environments. ACM SIGGRAPH 02 Course, July 2002.

[17] O. Aumage, L. Bougé, A. Denis, J.-F. Méhaut, G. Mercier, R. Namyst, and L. Prylli. A portable and efficient communication library for high-performance cluster computing. In *IEEE International Conference on Cluster Computing (CLUSTER 2000)*, pages 78–87, Tecnhnische Universität Chemnitz, Saxony, Germany, November 2000.

[18] O. Aumage, G. Mercier, and R. Namyst. MPICH/Madeleine : a true multi-protocol MPI for high-performance networks. In *Proc. 15th International Parallel and Distributed Processing Symposium (IPDPS 2001)*, San Fransisco, April 2001.

[19] I. S. Bains and J. A. Doss. ShaderGen – fixed functionality shader generation tool. `http://developer.3dlabs.com/downloads/shadergen/`.

[20] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc 2.0 Users Manual. Technical Report ANL-95/11 - Revision 2.0.29, Argonne National Laboratory, November 2000.

[21] B. Baumgart. A polyhedron representation for computer vision. In *AFIPS National Computer Conference*, pages 589–596, 1975.

[22] P. H. Beckman, P. K. Fasel, W. F. Humphrey, and S. M. Mniszewski. Efficient coupling of parallel applications using PAWS. In *Proceedings of the 7th IEEE International Symposium on High Performance Distributed Computation*, July 1998.

[23] F. Bertails and M.-P. Cani. Animation de chevelures : du temps-réel à la recherche de réalisme. In *Groupe de travail français Animation et Simulation (GTAS'04)*, Reims, Juin 2004.

[24] F. Bertrand, R. Bramley, K. B. Damevski, J. A. Kohl, D. E. Bernholdt, J. W. Larson, and A. Sussman. Data redistribution and remote method invocation in parallel component architectures. In *Proceedings of the 19th International Parallel and Distributed Processing Symposium : IPDPS 2005*, 2005.

[25] F. Bertrand, Y. Yuan, K. Chiu, and R. Bramley. An approach to parallel MxN communication. In *Proceedings of the Los Alamos Computer Science Institute (LACSI) Symposium*, Santa Fe, NM, October 2003.

[26] 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.

[27] L. Blackford, J. Choi, A. Cleary, J. Demmel, I. S. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. Whaley. ScaLAPACK : A portable linear algebra library for distributed memory computers - design issues and performance. In *Proceedings of Supercomputing '96*, 1996.

[28] D. Blythe, editor. *OpenGL® ES Common/Common-Lite Profile Specification, Version 1.0*. The Kronos Group Inc., 2003.

[29] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. Myrinet : A gigabit-per-second local area network. *IEEE Micro*, 15(1) :29–36, February 1995.

[30] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 594–603. ACM Press, 2002.

[31] K. W. Brodlie, D. A. Duce, J. R. Gallop, J. P. R. B. Walton, and J. D. Wood. Distributed and collaborative visualization. *Computer Graphics Forum*, 23(2), 2004.

[32] G. Burns, R. Daoud, and J. Vaigl. LAM : An Open Cluster Environment for MPI. In *Proceedings of Supercomputing Symposium*, pages 379–386, 1994.

[33] W. Cai, F. B. S. Lee, and L. Chen. An auto-adaptive dead reckoning algorithm for distributed interactive simulation. In *PADS '99 : Proceedings of the 13th workshop on Parallel and distributed simulation*, pages 82–89, 1999.

[34] M. Carlson, P. J. Mucha, and G. Turk. Rigid fluid : animating the interplay between rigid bodies and fluid. *ACM Trans. Graph. (Proceedings of ACM SIGGRAPH 04)*, 23(3) :377–384, 2004.

[35] D. Chapman. Jiggle – rigid body demo. http://www.rowlhouse.co.uk/jiggle/.

[36] Y. Chen, D. W. Clark, T. H. A. Finkelstein, and K. Li. Automatic Alignment Of High-Resolution Multi-Projector Displays Using An Un-Calibrated Camera. In *IEEE Visualization 2000*, Salt Lake City, USA, October 2000.

[37] G. Chiola and G. Ciaccio. Efficient Parallel Processing on Low-cost Clusters with GAMMA Active Ports. *Parallel Computing*, 26, 2000.

[38] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-collide : an interactive and exact collision detection system for large-scale environments. In *SI3D '95 : Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–ff. ACM Press, 1995.

[39] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart. The Cave Audio VIsual Experience Automatic Virtual Environement. *Communication of the ACM*, 35(6) :64–72, 1992.

[40] M. Czernuszenko, D. Pape, D. Sandin, T. DeFanti, G. L. Dawe, and M. D. Brown. The immersadesk and infinity wall projection-based virtual reality displays. *SIGGRAPH Comput. Graph.*, 31(2) :46–49, 1997.

[41] Dahlgren, et al. Babel user's guide. Lawrence Livermore National Laboratory, January 2005.

[42] K. Damevski and S. Parker. Parallel remote method invocation and M-by-N data redistribution. In *Proceedings of the 4th Los Alamos Computer Science Institute Symposium*, October 2003.

[43] V. Danjean, R. Namyst, and R. Russell. Integrating kernel activations in a multithreaded runtime system on linux. In *Parallel and Distributed Processing. Proc. 4th Workshop on Runtime Systems for Parallel Programming (RTSPP '00)*, pages 1160–1167, Cancun, Mexico, May 2000. In conjunction with IPDPS 2000.

[44] A. Darte, Y. Robert, and F. Vivien. *Scheduling and Automatic Parallelization*. Birkhäuser, 2000.

[45] J. D. de St. Germain, S. G. Parker, J. McCorquodale, and C. R. Johnson. Uintah : A massively parallel problem solving environment. In *HPDC'00 : Ninth IEEE International Symposium on High Performance and Distributed Computing*, pages 33–42, August 2000.

[46] I. Debled-Rennesson, S. Tabbone, and L. Wendling. Fast polygonal approximation of digital curves. In *Proceedings of International Conference on Pattern Recognition, Cambridge (UK)*, volume 1, pages 461–468, 2004.

[47] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *Proceedings of ACM SIGGRAPH 01*, Annual Conference Series. ACM Press / ACM SIGGRAPH, August 2001. Proceedings of ACM SIGGRAPH 01.

[48] A. Denis, C. Pérez, and T. Priol. Portable parallel CORBA objects : an approach to combine parallel and distributed programming for grid computing. In *Proceedings of the 7th International Euro-Par'01 Conference*, pages 835–844, Manchester, UK, 2001. Springer-Verlag.

[49] A. Denis, C. Pérez, and T. Priol. PadicoTM : An open integration framework for communication middleware and runtimes. In *IEEE International Symposium on Cluster Computing and the Grid (CCGRID2002)*, 2002.

[50] A. Denis, C. Pérez, and T. Priol. Achieving portable and efficient parallel CORBA objects. *Concurrency and Computation : Practice and Experience*, 15(10) :891–909, August 2003.

[51] F. Devernay and B. Holveck. Blinky, 2005. http://www.inria.fr/rapportsactivite/RA2004/movi/uid34.html.

[52] U. Drepper and I. Molnar. The native POSIX thread library for linux, 2003. http://people.redhat.com/drepper/nptl-design.pdf.

[53] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *HWWS '01 : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 9–16. ACM Press, 2001.

[54] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 736–744. ACM Press, 2002.

[55] A. Esnard. Modèle pour la redistribution de données complexes. In *RENPAR'16*, Le Croisic, France, April 2005.

[56] A. Esnard, M. Dussere, and O. Coulaud. A time-coherent model for the steering of parallel simulations. In *Proceedings of Euro-Par 2004*, Pisa, Italia, August 2004.

[57] F. Faure and L. Heïgéas. Animal, 2004. http://www-evasion.imag.fr/Membres/Laure.Heigeas/AnimAL/.

[58] I. Foster and C. Kesselman. Globus : A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11(2) :115–128, 1997.

[59] I. Foster, C. Kesselman, and S. Tuecke. The nexus approach to integrating multithreading and communication. *Journal of Parallel and Distributed Computing*, 37 :70–82, 1996.

[60] N. Foster and R. Fedkiw. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30. ACM Press, 2001.

[61] J. Franco and E. Boyer. Exact Polyhedral Visual Hulls. In *Proceedings of BMVC2003*, 2003.

[62] J.-S. Franco, C. Ménier, E. Boyer, and B. Raffin. A distributed approach for real time 3d modeling. In *Proceedings of the IEEE Workshop on Real Time 3D Sensors and Their Use*, Washington, USA, July 2004.

[63] F. Galilée, J.-L. Roch, G. Cavalheiro, and M. Doreille. Athapascan-1 : On-line Building Data Flow Graph in a Parallel Language. In IEEE, editor, *Pact'98*, Paris, France, oct 1998.

[64] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM 3 User's Guide and Reference Manual*. Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1994.

[65] G. A. Geist, J. A. Kohl, and P. M. Papadopoulos. CUMULVS : providing fault-tolerance, visualization and steering of parallel applications. *International Journal of High Performance Computing Applications*, 11(3) :224–236, August 1997.

[66] O. Génevaux, A. Habibi, and J.-M. Dischler. Simulating fluid-solid interaction. In *Graphics Interface*, pages 31–38. CIPS, Canadian Human-Computer Commnication Society, A K Peters, June 2003. ISBN 1-56881-207-8, ISSN 0713-5424.

[67] P. Geoffray, L. Prylli, and B. Tourancheau. BIP-SMP : High Performance Message Passing over a Cluster of Commodity SMPs. In *Proceedings of Super Computing 99*, Portland, USA, November 1999.

[68] GNOME Project. Orbit2. `http://www.gnome.org/projects/ORBit2/`.

[69] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of ACM SIGGRAPH 98*, pages 447–452. ACM Press, 1998.

[70] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree : A hierarchical structure for rapid interference detection. In *Proceedings of ACM SIGGRAPH 96*, pages 171–180. ACM Press, 1996.

[71] V. Gouranton, S. Limet, S. Madougou, and E. Melin. A scalable cluster-based parallel simplification framework for height fields. In *Eurographics Symposium on Parallel Graphics and Visualization, EGPGV'04*, June 2004.

[72] V. Gouranton, S. Madougou, E. Melin, and C. Nortet. Interactive rendering of massive terrains using PC cluster. In *EuroVis 2005 : Eurographics/IEEE-VGTC Symposium on Visualization*, June 2005.

[73] C. Greenhalgh and S. Benford. Massive : a collaborative virtual environment for teleconferencing. *ACM Trans. Comput.-Hum. Interact.*, 2(3) :239–261, 1995.

[74] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6) :789–828, September 1996.

[75] W. D. Gropp and E. Lusk. *User's Guide for* `mpich`*, a Portable Implementation of MPI*. Mathematics and Computer Science Division, Argonne National Laboratory, 1996. ANL-96/6.

[76] W. Grosso. *Java RMI*. O'Reilly & Associates, 2001.

[77] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. *ACM Trans. Graph. (Proceedings of ACM SIGGRAPH 03)*, 22(3) :871–878, 2003.

[78] O. Hagsand. Interactive multiuser ves in the dive system. *IEEE MultiMedia*, 3(1) :30–39, 1996.

[79] R. W. Hamming. *Digital Filters*. Prentice-Hall, 3rd edition, 1989.

[80] P. Hanrahan and J. Lawson. A language for shading and lighting calculations. In *Proceedings of ACM SIGGRAPH 90*, pages 289–298. ACM Press, 1990.

[81] M. Hereld, I. Judson, and R. Stevens. Introduction to Building Projection-based Tiled Display Systems. *IEEE Computer Graphics and Applications*, 20(4) :22–28, 2000.

[82] G. Hesina, D. Schmalsteig, 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*, pages 74–81, London, December 1999.

[83] B. Houston, C. Bond, and M. Wiebe. A unified approach for modeling complex occlusions in fluid simulations. In *Proceedings of the ACM SIGGRAPH 03 conference on Sketches & applications*, pages 1–1. ACM Press, 2003.

[84] G. Humphreys, M. Houston, and R. Ng. Chromium : A stream processing framework for interactive rendering on clusters. In *Computer Graphics Proceedings*, Annual Conference Series. ACM Press / ACM SIGGRAPH, Aug 2002. Proceedings of ACM SIGGRAPH 02.

[85] G. Humphreys, M. Houston, R. Ng, S. Ahern, R. Frank, P. Kirchner, and J. T. Klosowski. Chromium : A Stream Processing Framework for Interactive Graphics on Clusters of Workstations. In *Proceedings of ACM SIGGRAPH 02*, pages 693–702, 2002.

[86] IEEE. Standard for Distributed Interactive Simulation – Application Protocols (IEEE Std 1278.1). IEEE Comp. Soc., 1995.

[87] IEEE. Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules (IEEE Std 1516). IEEE Comp. Soc., 2000.

[88] R. M. T. II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser. VRPN : a device-independent, network-transparent vr peripheral system. In *ACM Symposium on Virtual Reality Software & Technology*, 2001.

[89] Institute for Simulation and Training. Communication architecture for distributed interactive simulation (CADIS). Final Draft IST-TR-93-20, University of Central Florida, Orlando, Florida, 1993.

[90] Intel. Open Source Computer Vision Library. `http://www.intel.com/research/mrl/research/opencv/`.

[91] S. Jafar, T. Gautier, A. W. Krings, and J.-L. Roch. A checkpoint/recovery model for heterogeneous dataflow computations using work- stealing. In L. Springer-Verlag, editor, *Euro-Par'2005*, Lisboa, Portogal, august 30 – september 2 2005.

[92] C. Just, A. Bierbaum, A. Baker, and C. Cruz-Neira. VR Juggler : A Framework for Virtual Reality Development. In *2nd Immersive Projection Technology Workshop (IPT98)*, Ames, USA, May 1998.

[93] T. Kamachi, T. Priol, and C. René. Data distribution for parallel CORBA objects. In *Euro-Par'2000 : Parallel Processing*, pages 1239–1250, Munchen, Germany, August 2000. Springer-Verlag.

[94] A. Kanitsar, T. Theussi, L. Mroz, M. Sramek, A. V. Bartroli, B. Csebfalvi, J. Hladuvka, D. Fleischmann, M. Knapp, R. Wegenkittl, P. Felkel, S. Roettger, S. Guthe, W. Purgathofer, and M. E. Groller. Christmas tree case study : computed tomography as a tool for mastering complex real world objects with applications in computer graphics. In *Proceedings of IEEE Visualization'02*, pages 489–492. IEEE Computer Society, 2002.

[95] N. Karonis, B. Toonen, and I. Foster. MPICH-G2 : A Grid-Enabled Implementation of the Message Passing Interface. *Journal of Parallel and Distributed Computing*, 63(5) :551–563, 2003.

[96] K. Keahey, P. Beckman, and J. Ahrens. Ligature : Component architecture for high-performance applications. In *1st NASA Workshop on Performance-Engineered Information Systems*, 1998. To appear in the International Journal of High-Performance and Scientific Applications.

[97] K. Keahey and D. Gannon. PARDIS : A parallel approach to CORBA. In *6th International Symposium on High Performance Distributed Computing (HPDC '97)*, pages 31–39, Portland, Oregon, USA, August 1997. IEEE.

[98] K. Keahey and D. Gannon. Developing and evaluating abstractions for distributed supercomputing. *Cluster Computing*, 1(1) :69–79, 1998.

[99] M. Keckeisen and W. Blochinger. Parallel Implicit Integration for Cloth Animations on Distributed Memory Architectures. In *Proceedings of the Eurographics Parallel Graphics and Visualization Ssymposium*, Grenoble, France, June 2004.

[100] E. Kilgariff and R. Fernando. The GeForce 6 series GPU architecture. In M. Pharr and R. Fernando, editors, *GPU Gems 2 : Programming Techniques for High-Performance Graphics and General-Purpose Computation*, chapter 30, pages 471–491. Addison-Wesley, 2005.

[101] J. T. Klosowski, M. Held, J. S. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1) :21–36, 1998.

[102] S. Krishnan and D. Gannon. XCAT3 : A framework for CCA components as OGSA services. In *Proceedings of HIPS 2004, 9th International Workshop on High-Level Parallel Programming Models and Supportive Environments*, April 2004.

[103] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on PAMI*, 16(2) :150–162, February 1994.

[104] N. A. Le Khac. *Définition et évaluation d'INUKTITUT : une interface pour l'environnement de programmation parallèle asynchrone Athapascan*. PhD thesis, ID / IMAG, Institut National Polytechnique de Grenoble, France, March 2005.

[105] K. Li, H. Chen, Y. Chen, D. W. Clark, P. Cook, S. Damianakis, G. Essl, A. Finkelstein, T. Funkhouser, A. Klein, Z. Liu, E. Praun, R. Samanta, B. Shedd, J. P. Singh, G. Tzanetakis, and J. Zheng. Early Experiences and Challenges in Building and Using A Scalable Display Wall System. *IEEE Computer Graphics and Application*, 20(4) :671–680, 2000.

[106] K.-C. Lin. Dead reckoning and distributed interactive simulation. In *Proceedings of SPIE Conference (AeroSense'95)*, April 1995.

[107] J. Löwy. *COM & .NET Component Services*. O'Reilly & Associates, 2001.

[108] K.-L. Ma, J. S. Painter, C. D. Hansen, and M. F. Krogh. Parallel volume rendering using binary-swap compositing. *IEEE Computer Graphics and Applications*, 14 :59–68, July 1994.

[109] D. Margery, B. Arnaldi, A. Chauffaut, S. Donikian, and T. Duval. Openmask : {Multi-Threaded | Modular} animation and simulation {Kernel | Kit } : a general introduction. In Simon Richir, Paul Richard, and Bernard Taravel, editors, *VRIC 2002 Proceedings*, pages 101–110. ISTIA Innovation, June 2002.

[110] W. R. Mark, R. S. Glanville, K. Akeley, and M. J. Kilgard. Cg : a system for programming graphics hardware in a c-like language. In *Proceedings of ACM SIGGRAPH 03*, pages 896–907. ACM Press, 2003.

[111] J. E. Marvie, J. Perret, and K. Bouatouch. Remote interactive walkthrough of city models. In *Proceedings of Pacific Graphics*, volume 2, pages 389–393. IEEE Computer Society, October 2003. Short Paper.

[112] C. Meerwald. Free JAVA & C++ ORBs. http://cmeerw.org/freeorbwatch/.

[113] Message Passing Interface Forum. *MPI : A Message–Passing Interface Standard*. University of Tennessee, Knoxville, Tennessee, 1994.

[114] Message Passing Interface Forum. *MPI-2 : Extensions to the Message–Passing Interface*. University of Tennessee, Knoxville, Tennessee, April 1997.

[115] Microsoft. Distributed component object model. http://www.microsoft.com/com/.

[116] A. Mohr and M. Gleicher. Non-invasive, interactive, stylized rendering. In *SI3D '01 : Proceedings of the 2001 Symposium on Interactive 3D graphics*, pages 175–178. ACM Press, 2001.

[117] A. Mohr and M. Gleicher. HijackGL : reconstructing from streams for stylized rendering. In *NPAR '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 13–ff. ACM Press, 2002.

[118] S. Molnar, M. Cox, D. Ellsworth, and H. Fuchs. A Sorting Classification of Parallel Rendering. *IEEE Computer Graphics and Applications*, 14(4) :23–32, July 1994.

[119] K. Moreland, B. Wylie, and C. Pavlakos. Sort-last parallel rendering for viewing extremely large data sets on tile displays. In *Proceedings of the IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*, 2001.

[120] C. Morin, R. Lottiaux, G. Vallée, P. Gallard, G. Utard, R. Badrinath, and L. Rilling. Kerrighed : a single system image cluster operating system for high performance computing. In *Euro-Par 2003*, Klagenfurt, Austria, August 2003.

[121] J. D. Mulder, J. J. van Wijk, and R. van Liere. A survey of computational steering environments. *Future Generation Computer Systems*, 15(1), 1999.

[122] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross. Interaction of fluids with deformable solids. *Journal of Computer Animation and Virtual Worlds (CAVW)*, 15 :159–171, 2004.

[123] M. Naef, E. Lamboray, O. Staadt, and M. Gross. The Blue-C distributed scene graph. In *Proceedings of IEEE Virtual Reality 2003*, pages 275–276, March 2003.

[124] R. Namyst and J.-F. Méhaut. PM2 : Parallel multithreaded machine. a computing environment for distributed architectures. In *Parallel Computing (ParCo'95)*, pages 279–285. Elsevier, 1995.

[125] C. Niederauer, M. Houston, M. Agrawala, and G. Humphreys. Non-invasive interactive visualization of dynamic architectural environments. In *SI3D '03 : Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 55–58. ACM Press, 2003.

[126] J. Nieplocha, R. J. Harrison, and R. J. Littlefield. Global arrays : A nonuniform memory access programming model for high-performance computers. *The Journal of Supercomputing*, 10(2) :169–189, 1996.

[127] NovodeX AG. NovodeX. `http://www.novodex.com`.

[128] Object Management Group. The common object request broker : Architecture and specification, v. 2.0, July 1995. Document formal/97-02-25.

[129] Object Management Group. Data parallel CORBA, November 2001. Document ptc/01-11-09.

[130] Object Management Group. The common object request broker : Architecture and specification, v. 3.0, July 2002. Document formal/02-06-01.

[131] Object Management Group. CORBA components specification, v. 3.0, June 2002. Document formal/02-06-65.

[132] Ohio Supercomputer Center. Xmpi – a run/debug GUI for MPI. http://www.lam-mpi.org/software/xmpi/.

[133] M. Olano and A. Lastra. A shading language on graphics hardware : the pixelflow shading system. In *Proceedings of ACM SIGGRAPH 98*, pages 159–168. ACM Press, 1998.

[134] E. Olson. Cluster Juggler–PC cluster virtual reality. Master's thesis, Iowa State University, 2002.

[135] OMG CCM Implementers Group. CORBA component model tutorial, April 2002. Document ccm/2002-04-01.

[136] Open Software Foundation. OSF DCE application environment specification, 1992.

[137] ORL. omniorb. http://omniorb.sf.net.

[138] D. Pape, C. Cruz-Neira, and M. Czernuszenko. *CAVE User's Guide*. Electronic Visualization Laboratory, University of Illinois at Chicago, 1997.

[139] S. Parker and C. Johnson. SCIRun : A scientific programming environment for computational steering. In *Supercomputing '95*. IEEE Press, 1995.

[140] S. Pettifer, J. Cook, J. Marsh, and A. West. Deva3 : architecture for a large-scale distributed virtual reality system. In *VRST '00 : Proceedings of the ACM symposium on Virtual reality software and technology*, pages 33–40. ACM Press, 2000.

[141] T. Pohl, N. Thuerey, F. Deserno, U. Ruede, P. Lammers, G. Wellein, and T. Zeiser. Performance Evaluation of Parallel Large-Scale Lattice Boltzmann Applications on Three Supercomputing Architectures. In *Proceedings of SC2004*, November 2004.

[142] K. Proudfoot, W. R. Mark, S. Tzvetkov, and P. Hanrahan. A real-time procedural shading system for programmable graphics hardware. In *Proceedings of ACM SIGGRAPH 01*, pages 159–170. ACM Press, 2001.

[143] C. Pérez, T. Priol, and A. Ribes. A parallel CORBA component model for numerical code coupling. In *Proceedings of the 3th International Workshop on Grid Computing*, pages 88–99, Baltimore, Maryland, November 2002. Springer-Verlag.

[144] P. Rajlich. Cave Quake 3 Arena. http://www.visbox.com/cq3a.

[145] D. Rantzau, K. Frank, U. Lang, D. Rainer, and U. Wössner. COVISE in the CUBE : An environnement for analyzing large and complex simulation data. In *2nd Immersive Projection Technology Workshop (IPT98)*, Ames, USA, May 1998.

[146] D. Reiners, G. Voss, and J. Behr. OpenSG basic concepts. In *OpenSG 2002 Symposium*, Darmstadt, 2002.

[147] G. Reitmayr and D. Schmalstieg. An open software architecture for virtual reality interaction. In *ACM Symposium on Virtual reality software & Technology*, pages 47–54, 2001.

[148] C. René and T. Priol. MPI code encapsulating using parallel CORBA objects. In *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computing*, pages 3–10, Redondo Beach, California, USA, August 1999. IEEE.

[149] G. Robertson, S. K. Card, and J. D. Mackinlay. The cognitive coprocessor architecture for interactive user interfaces. In *UIST '89 : Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology*, pages 10–18. ACM Press, 1989.

[150] J.-L. Roch and *et al.* Athapascan : API for Asynchronous Parallel Programming. Technical report, INRIA Rhône-Alpes, projet APACHE, feb 2003.

[151] S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart hardware-accelerated volume rendering. In *VISSYM '03 : Proceedings of the symposium on Data visualisation 2003*, pages 231–238. Eurographics Association, 2003.

[152] J. Rohlf and J. Helman. IRIS Performer : A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics. In *Computer Graphics (ACM SIGGRAPH 94)*, pages 381–394. ACM Press, July 1994.

[153] R. J. Rost. *OpenGL® Shading Language*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[154] M. Roth, G. Voss, and D. Reiners. Multi-threading and clustering for scene graph systems. *Computers & Graphics*, 28(1) :63–66, 2004.

[155] T. Saito and J. Toriwaki. New algorithms for euclidean distance transformations of an n-dimensional digitised picture with applications. *Pattern Recognition*, 27(11) :1551–1565, 1994.

[156] B. Schaeffer and C. Goudeseune. Syzygy : Native PC Cluster VR. In *IEEE VR Conference*, 2003.

[157] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit An Object-Oriented Approach To 3D Graphics, 3rd Edition*. Kitware, Inc., 2003.

[158] Scientific Computing and Imaging Institute. SCIRun : A Scientific Computing Problem Solving Environment. http://software.sci.utah.edu/scirun.html.

[159] M. Segal and K. Akeley. *The OpenGL™ Graphics System : A Specification, Version 1.0*. Silicon Graphics, 1992.

[160] C. Shaw, M. Green, J. Liang, and Y. Sun. Decoupled simulation in virtual reality with the mr toolkit. *ACM Trans. Inf. Syst.*, 11(3) :287–317, 1993.

[161] S. Singhal and M. Zyda. *Networked Virtual Environments - Design and Implementation*. ACM SIGGRAPH Series. ACM Press Books, 2000.

[162] L. Smarr and C. E. Catlett. Metacomputing. *Communications of the ACM*, 35(6) :44–52, 1992.

[163] R. Smith et al. Open dynamics engine. http://www.ode.org.

[164] J. M. Squyres and A. Lumsdaine. A Component Architecture for LAM/MPI. In *Proceedings, 10th European PVM/MPI Users' Group Meeting*, number 2840 in Lecture Notes in Computer Science, pages 379–387, Venice, Italy, September / October 2003. Springer-Verlag.

[165] J. Stam. Stable Fluids. In *Proceedings of ACM SIGGRAPH 99*, pages 121–128, August 1999.

[166] J. Stam. Interacting with smoke and fire in real time. *Communications of the ACM*, 43(7) :76–83, 2000.

[167] W. R. Stevens. *UNIX Network Programming – Networking APIs : Sockets and XTI*, volume 1. Prentice-Hall, 2nd edition, 1998.

[168] G. Stoll, M. Eldridge, D. Patterson, A. Webb, S. Berman, R. Levy, C. Caywood, M. Taveira, S. Hunt, and P. Hanrahan. Lightning-2 : A High-Performance Display Subsystem for PC Clusters. In *Proceedings of ACM SIGGRAPH 01*, 2001.

[169] P. S. Strauss. Iris inventor, a 3d graphics toolkit. In *OOPSLA '93 : Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications*, pages 192–200. ACM Press, 1993.

[170] Sun Microsystems. Entreprise JavaBeans^{TM} specification, version 2.0, August 2001. http://java.sun.com/products/ejb/docs.html.

[171] I. E. Sutherland. The ultimate display. In *Proceedings of the IFIPS Congress*, volume 2, pages 506–508, New York City, NY, USA, May 1965.

[172] C. Szyperski. *Component Software – Beyond Object-Oriented Programming*. Addison-Wesley / ACM Press, 1998.

[173] P. Terdiman. OPCODE : Optimized collision detection. http://www.codercorner.com/Opcode.htm.

[174] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : Principles and practice of background maintenance. In *Proceedings of the 7th International Conference on Computer Vision*, pages 255–261, Kerkyra, Greece, 1999.

[175] University of Minnesota. PowerWall. http://www.lcse.umn.edu/research/powerwall/powerwall.html.

[176] University of North Carolina. Collision detection systems. http://www.cs.unc.edu/~geom/collide/.

[177] C. Upson, T. Faulhaber Jr., D. Kamins, D. H. Laidlaw, D. Schlegel, L. Vroom, R. Gurwitz, and A. van Dam. The Application Visualization System : A computational environment for scientific visualization. *IEEE Computer Graphics and Applications*, 9(4) :30–42, 1989.

[178] L. G. Valiant. A Bridging Model for Parallel Computation. *Communications of the ACM*, 33(8) :103–111, August 1990.

[179] T. von Eicken, D. Culler, S. C. Goldstein, and K. E. Schauser. Active messages : a mechanism for integrated communication and computation. In *19th International Symposium on Computer Architecture*, May 1992.

[180] VRCO. Trackd. http://www.vrco.com/trackd/Overviewtrackd.html.

[181] C. Wang, J. Gao, and H.-W. Shen. Parallel Multiresolution Volume Rendering of Large Data Sets with Error-Guided Load Balancing. In *Proceedings of the Eurographics Parallel Graphics and Visualization Ssymposium*, pages 23–30, Grenoble, France, June 2004.

[182] D. Winer. XML-RPC specification, 1999. http://www.xmlrpc.com/spec.

[183] World Wide Web Consortium. SOAP 1.2. http://www.w3.org/TR/soap/.

[184] F. Zara. Simulation physique de textiles sur grappe de processeurs. In *AFIG 2001*, Limoges, November 2001.

[185] F. Zara, F. Faure, and J.-M. Vincent. Physical cloth simulation on a PC cluster. In X. P. D. Bartz and E. Reinhard, editors, *Fourth Eurographics Workshop on Parallel Graphics and Visualization 2002*, Blaubeuren, Germany, September 2002.

[186] K. Zhang, K. Damevski, V. Venkatachalapathy, and S. G. Parker. SCIRun2 : A CCA framework for high performance computing. In *9th International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS'04)*, pages 72–79, 2004.

[187] S. Zhou, W. Cai, B.-S. Lee, and S. J. Turner. Time-space consistency in large-scale distributed virtual environments. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 14(1) :31–47, January 2004.

Figures couleur

Résumé

FlowVR : calculs interactifs et visualisation sur grappe

Cette thèse combine le calcul haute performance à la réalité virtuelle pour permettre la conception de méthodes de couplage de composants parallèles à l'intérieur d'applications distribuées et interactives.

Un nouveau modèle de couplage est présenté, conçu selon des critères de modularité, simplicité, efficacité et extensibilité. La construction des applications repose sur une séparation entre la programmation de modules parallèles réutilisables et la définition de l'application sous forme de graphe de flux de données contenant des mécanismes de filtrage et de synchronisations, permettant d'exprimer des schémas de communication collective et des politiques de couplage avancées.

Ce travail sur le couplage interactif est complété par une extension haut niveau concernant le rendu distribué. En exploitant une description modulaire de la scène 3D en primitives indépendantes basées sur l'utilisation de shaders, des réseaux de filtrage permettent de combiner plusieurs flux pour acheminer efficacement les informations aux machines de rendu. Ce système est très extensible et permet la création de nouvelles applications exploitant la puissance des cartes graphiques pour décharger certains calculs des processeurs et réduire les transferts réseau.

De nombreuses applications nouvelles sont ainsi développées, combinant des algorithmes de vision parallélisés immergeant l'utilisateur dans l'environnement virtuel, et des interactions avec des objets contrôlés par des simulations physiques distribuées (poterie, collisions, fluides).

Mots-clés Couplage de codes parallèles, simulations interactives, rendu distribué, réalité virtuelle, grappes de PC.

Abstract

FlowVR : interactive computations and cluster-based visualization

This thesis combines high performance computing and virtual reality to design parallel components coupling methods inside distributed interactive applications.

A new coupling model designed with modularity, simplicity, efficiency and extensibility criteria is presented. Applications are constructed following a two phases process : programming reusable parallel components, and defining the application as a data-flow graph with filtering and synchronisation mechanisms, expressing advanced collective communication schemes and coupling policies.

An extension of this work is presented for distributed rendering. Using a modular scene description based on independent shader-based primitives, filtering networks are designed to efficiently combine multiple graphics streams up to the rendering nodes. This framework scales well and allow the creation of new applications, exploiting the graphics cards' power to remove computations from CPUs which reducing network communications.

Several innovative applications are created using these tools. They includes parallel computer vision algorithms, immersing the user inside a virtual world, which can then interact with objects controlled by several distributed physical simulations (carving, collisions, fluids).

Keywords Parallel code coupling, interactive simulations, distributed rendering, virtual reality, PC clusters.