A Design Tool for High Performance Image Processing on Multi core Platforms

ABSTRACT

- Design and implementation of smart vision systems often involve the mapping of complex image processing algorithms into efficient, real-time implementations on multi core platforms. In this paper, we describe a novel design tool that is developed to address this important challenge. A key component of the tool is a new approach to hierarchical dataflow scheduling that integrates a global scheduler and multiple local schedulers. The local schedulers are lightweight modules that work independently. The global scheduler interacts with the local schedulers to optimize overall memory usage and execution time. The proposed design tool is demonstrated through a case study involving an image stitching application for large scale microscopy images.

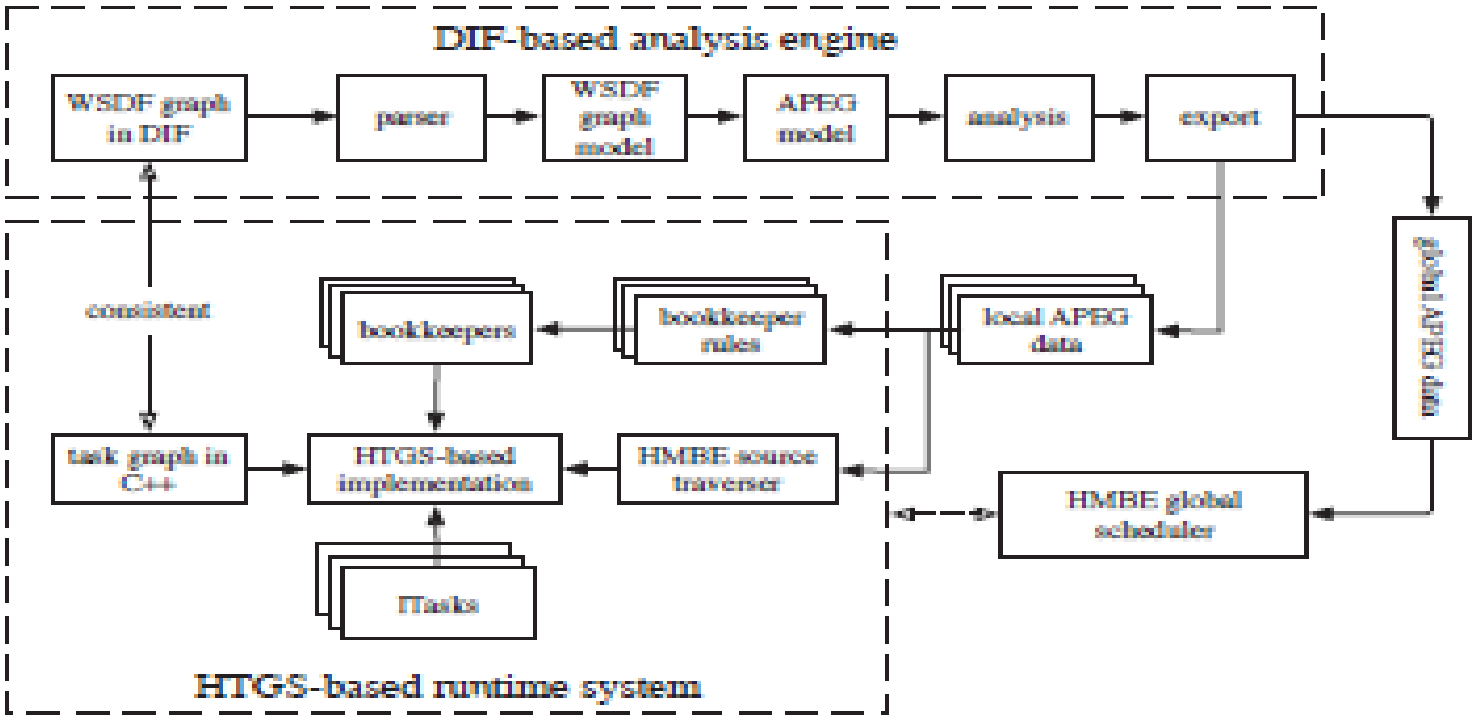
EXISTING SYSTEM

- Making effective use of multicore platforms poses considerable challenges to designers of smart vision systems. These challenges include delivering real-time embedded computer vision performance under complicated data dependencies among application tasks, and the nondeterministic execution characteristics of thread-based programming models.
- Additionally, limited memory availability on embedded platforms along with the data intensive nature of smart vision applications necessitate special attention to memory management.

PROPOSED SYSTEM

- In this paper, we describe a novel design tool that is developed to address this important challenge. A key component of the tool is a new approach to hierarchical dataflow scheduling that integrates a global scheduler and multiple local schedulers. The local schedulers are lightweight modules that work independently.
- The global scheduler interacts with the local schedulers to optimize overall memory usage and execution time. The proposed design tool is demonstrated through a case study involving an image stitching application for large scale microscopy images.

ARCHITECTURE DIAGRAM



SYSTEM SPECIFICATIONS

Hardware Requirements :

- Processor :Intel Pentium IV 1GHz
- RAM :256MB (Min)
- Hard Drive :5GB free space
- Monitor :1024 * 768, High Color inch
- Mouse :Scroll Mouse(Logitech)
- Keyboard :104 keys

CONTD..

Software Requirements :

- OS : Windows XP/7/8
- Front End : Visual Studio 2008
- Back End : SQL Server 2005
- Browser : Any Web Browser

CONCLUSION

- In this paper, we have presented a software tool for design and implementation of multicore image processing systems. This tool consists of two main parts — the DIF-based analysis engine, which applies the Dataflow Interchange Format(DIF) Package, and the HTGS-based runtime system, which builds on the Hybrid Task Graph Scheduler (HTGS). The tool allows system designers to incorporate powerful techniques for performance optimization and memory management while specifying applications at a high level of abstraction and using significant amounts of automation..

CONTD..

- Our experiments demonstrate the ability of our new design tool to provide this high level of abstraction and automation while generating efficient implementations on a diverse set of platforms. Useful directions for future work include extending the hierarchical scheduling techniques developed in this work to heterogeneous platforms, such as CPU/GPU platforms

REFERENCES

- [1] T. Blattner, W. Keyrouz, M. Halem, M. Brady, and S. S. Bhattacharyya, "A hybrid task graph scheduler for high performance image processing workflows," in *Proceedings of the IEEE Global Conference on Signal and Information Processing, 2015, pp. 634–637.*
- [2] J. Wu, T. Blattner, W. Keyrouz, and S. S. Bhattacharyya, "Model-based dynamic scheduling for multicore implementation of image processing systems," in *Proceedings of the IEEE Workshop on Signal Processing Systems, 2017.*
- [3] S. Sriram and S. S. Bhattacharyya, *Embedded Multiprocessors: Scheduling and Synchronization, 2nd ed. CRC Press, 2009, iSBN:1420048015.*
- [4] C. Hsu, M. Ko, and S. S. Bhattacharyya, "Software synthesis from the dataflow interchange format," in *Proceedings of the International Workshop on Software and Compilers for Embedded Systems, Dallas,* Texas, September 2005, pp. 37–49.
- [5] J. Eker and J. W. Janneck, "Dataflow programming in CAL — balancing expressiveness, analyzability, and implementability," in *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers,* 2012, pp. 1120–1124.