**Author**- Suman Bidarahalli

**Supervisors**

**University** – Prof. Dr. Ir. Arjan Kuijper

**IGD** –Dr. Alan Brunton

**Motivation**

Determining material arrangements to control high-resolution multi-material 3D printers for reproducing shape and visual attributes of a 3D model (e.g. spatially-varying color, translucency and gloss) requires large computational effort [1]. Today’s resolution and print tray sizes allow prints with more than 10^12 voxels each filled with one of the available printing materials (today up to 7 materials can be combined in a single printout). Cuttlefish, a 3D printing pipeline [2], processes the input in a serial fashion leading to increased computation time for higher number of models. Distributed computing is one way of achieving better performance for large computations [3] [4].

**Abstract**

In this master thesis, cluster based software architecture [5] will be developed that distributes the computational task to multiple nodes in the cluster and to merge the resulting partial output. The partial output is in form of a *slice:* a per print layer 2D representation of printer parameters for each (x, y) coordinate. The sub-slices are merged to create full slices used to control the printer. The architecture must support streaming, which is required to rapidly start the print before the full computation is finished, as cuttlefish processes the input in small parts and generates chunk wise output [6]. Due to the streaming architecture, not only is the time to start the print reduced but it is also possible to process huge amount of data (i.e. voxels) without having to store it in the main memory [7]. The task distribution optimization should be done on the basis of cost function which would be computed using different parameters like object size, number of objects and number of nodes in the cluster. The architecture shall be integrated into the universal 3D printer driver Cuttlefish, and tested with different cluster size and print objects.

**Tasks**

* Installation
  + Setting up the hardware, for example: connecting all the nodes of the cluster in a common network.
  + Installation of the software needed for clustering, for example: installation of MPI SDK on the nodes.
* Master Component
  + Creating component for computing the cost function for each input model.
  + The cost function should be chosen on the basis of the constraints set in the configuration file.
  + Depending on the cost function, splitting the tasks among the slaves.
  + Send the configuration of the workload to the slaves.
  + Receive and de-serialize the slave output to sub-slices.
  + Merging the sub-slices to full slices which are used to control the printer.
* Slave Component
  + Creating the slave component which waits until the master provides the workload.
  + Serialization of the sub-slices generated by each of the slaves.
  + Sending the serialized data to the master.
* Analysis of the output and performance
  + Evaluate the output of the slaves.
  + Evaluate and compare the cost functions for distribution of different types of models and model size.
  + Evaluate the performance in terms of computational speed up for different cluster size.
  + Evaluate the overhead of cost computation, distribution, serialization and deserialization.

**References**

1. Kiril Vidimče, Szu-Po Wang, Jonathan Ragan-Kelley, and Wojciech Matusik. 2013. OpenFab: a programmable pipeline for multi-material fabrication. ACM Trans. Graph. 32, 4, Article 136 (July 2013), 12 pages. DOI=http://dx.doi.org/10.1145/2461912.2461993
2. "Cuttlefish.De - Home". *Cuttlefish.de*. N.p., 2016. Web. 21 July 2016.
3. ANDREWS, Gregory R. Foundations of multithreaded, parallel, and distributed programming. Reading: Addison-Wesley, 2000. xx, 664 s. ISBN 0-201-35752-6
4. Godfrey, Bill (2002). "A primer on distributed computing"
5. N. Desai, R. Bradshaw, E. Lusk and R. Butler, "Component-based cluster systems software architecture a case study," Cluster Computing, 2004 IEEE International Conference on, San Diego, CA, 2004, pp. 319-326. DOI: 10.1109/CLUSTR.2004.1392629
6. Alan Brunton, Can Ates Arikan, and Philipp Urban. 2015. Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials. ACM Trans. Graph. 35, 1, Article 4 (December 2015), 13 pages. DOI=http://dx.doi.org/10.1145/2832905
7. C. Charles Law, William J. Schroeder, Kenneth M. Martin, and Joshua Temkin. 1999. A multi-threaded streaming pipeline architecture for large structured data sets. In Proceedings of the conference on Visualization '99: celebrating ten years (VIS '99). IEEE Computer Society Press, Los Alamitos, CA, USA, 225-232.