Thesis Proposal:
Data-Parallel Transcoding for the 3D Internet

René Schubotz, Airbus Group CTO TX5AI

March 10, 2014

1 Background and Overview

In order to build 3D models, people usually use some form of Digital Content Creation (DCC) tool. This can be a modeling system like Maya or 3D Studio Max, and also a CAD system or simulation package. They all usually allow exporting the internal representation to some form of 3D data file.

Current declarative Web3D technologies use such 3D data embedded in (X)HTML pages and optional 3D data files referenced by the embedded part. The 3D data files can reference further 3D data files and therefore build an asset hierarchy that is rendered within the browser.

In an aerospace enterprise setting, such Web3D technologies can help to further integrate different engineering domains into Web-based collaborative engineering environments. This, however, requires tooling to analyze, optimize and publish potentially very large 3D data sets.

2 Problem Definition

We investigate the idea of transcoding 3D assets in parallel, and develop an approach that scales reasonably well on commodity clusters. Since we are more interested in decreasing the time for processing a single datum than increasing the throughput of the overall system, we propose a data-parallel approach.

Here, the X3D/XML document is divided in some number of chunks, and each processing element would work on the chunks independently. As the chunks are processed, the results are merged.

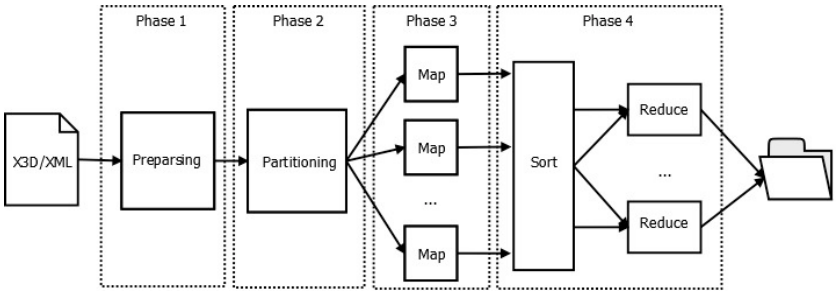

Figure 1: Data-parallel Transcoding Architecture

The main contributions of this thesis are

- data-parallel transcoding approach for declarative Web3D
- experimental evaluation and performance analysis

3 Approach

We assume availability of a 3D asset conversion pipeline to import various 3D model formats in a uniform manner. After model import, various post processing tools, e.g. frequently-needed operations such as computing normal and tangent vectors, allow for optimizing the data. We further expect an export interface which writes the processed 3D assets in X3D/XML format to the regular file system.

In the following, we provide an architecture blueprint (cf. Figure 1) that enables data-parallel transcoding of potentially very large scene graphs.

Phase 1. We use a *preparsing* step to determine some logical structure of the X3D/XML document. The resulting skeleton is used to partition the scene graph document into appropriate chunks and to guide the parallel transcoding.

Phase 2. Once the preparsing is complete, we are able to *partition* the scene graph document into appropriate fragments or chunks.

Phase 3. The generated chunks are *distributed* as key-value pairs across multiple computing nodes that perform the same operation concurrently.

Phase 4. After data-parallel processing, the intermediate results are *merged* into a hyperlink graph of Web3D resources for presentation and sharing on the Web.

In the scope of this thesis, we are interested in decomposing scene graphs with respect to some application-driven level of granularity, eg. with respect to certain node properties such as type, label or any other attribute value.