Abstract—In this paper, we develop the data intensive computing method for the large amounts of tomographic data from ANKA. Based on the large scale data facility (LSDF), an automatic workflow is built up for the combined tomography beamline of ANKA. In this workflow, a dedicated reconstruction method is still missing. Different from the existing reconstruction system, a compressive sampling-based algebraic reconstruction technique with more precise forward model is implemented on the parallel computing architecture: Hadoop cluster. Good results and high computing performance are reported for the 3D X-ray CT reconstruction. With this reconstruction structure, LSDF is not only able to efficiently organize data, but also can provide reconstructed results to the users in near real time.

Index Terms—Data Intensive Computing; Large Scale Data Facility; Computed Tomography; Algebraic Reconstruction Technique; Compressive Sampling

I. INTRODUCTION

Data intensive computing has emerged as a topic to address the growing need of storing, managing, accessing, and processing large volumes of scientific and social data. Its application includes various fields that need to process large amounts of data. A typical application is computed tomography (CT) [1], which is known as a noninvasive technology to image the inner structure of objects. ANKA is a synchrotron light source located at Karlsruhe Institute of Technology (KIT) [2]. From its Topo-Tomo beamline [3], a series of projection data is recorded by a high-velocity camera with high image resolution in many angles. Especially for ultra fast tomography, higher spatial and time resolution has been developed to reduce the impact of the movement of biological organisms. The 3D tomographic scanning could be performed in less than one second. It’s a challenging task to store and process the data of a high-speed and high-throughput imaging beamline. Currently, most storage and processing operations are directly steered by the users and high latency exists from data acquisition to visualization. Thus there is a need to make the immense quantities of data accessible for long periods of time with high bandwidth and automatically provide reconstruction results to users in real time.

The Large Scale Data Facility (LSDF) [4], at the same campus of KIT as ANKA, is designed to cope with the increasing requirements of data intensive scientific experiments. It is a distributed storage facility at petabyte scale to support storage, archives, data bases and meta data repositories. Moreover, a computing cluster is connected for data intensive application with a high speed dedicated network infrastructure. The combination of the tomography beamline of ANKA with the LSDF will greatly enhance the storage and analysis efficiency. An automatic workflow of LSDF is designed to manage the large amounts of tomographic data from ANKA, showed in Fig. 1.

At the ANKA side, projections are recorded continuously by the detector or camera in tomography experiments. Then the data flows through the data acquisition system “DAQ” and “Online Analysis” to ensure its validity. After the “Pre-Archival” step, where the experiment and user meta-data are added, the data is prepared for storage and processing in LSDF. At the LSDF side, the data set is firstly stored into the LSDF data storage resources, and at the same time the related meta-data is written in a database, providing information like the data set location, user meta-data, experiment configuration, processing and monitoring meta-data in LSDF, etc. In the next step of “Schedule Processing”, the execution framework for data intensive applications - LAMBDA [5] regularly queries the monitoring meta-data. Any new data set prepared for processing can be detected automatically and the Creation of a processing procedure is activated, which is done by LAMBDA according to the processing description in the meta-data. The procedure is converted to a script that is executable on computer. Afterwards, the script is submitted to a computing cluster, where the data set is distributed into different nodes for parallel processing. Finally, “Result Analysis” evaluates the quality of results, stores them back into LSDF storage in NeXus file format. Possible failures or errors are reported in this step and handled by users or developers. The users can access the raw data set and reconstruction volume in LSDF storage by the data browser interface KDM [6].

This automatic data processing workflow of LSDF is currently under construction. My work in this paper focuses on the reconstruction and parallel computing of one data set from ANKA. For reconstruction, a dedicated algorithm is still missing. For the ultra fast tomography imaging, fewer projections are actually obtained because of the high rotation speed of the probe. This change may cause a decline in recon-
truction image quality. However the current data processing system cannot handle the quality decline effectively. The immense quantities of data obtained in that way are processed using the filtered back-projection (FBP) method running on the graphics processing unit (GPU) [7], which displays high computational speed for 3D CT image reconstruction [8], [9]. Even though the FBP method is still the approach widely used, its defects are also obvious. Firstly, the FBP method is based on the radon transform model [10], in which the ray is a straight line going through the object. This model is no longer appropriate when the sensor geometry in the camera is considered. Moreover, FBP method can only be applied when the object is densely sampled. Otherwise, severe aliasing artifacts such as sharp streaks could appear.

Thus in the automatic workflow we resort to another entirely different tomographic reconstruction method from FBP called algebraic reconstruction technique (ART) [1]. A more precise forward model is introduced, in which the ray width and distance are taken into account. Based on the forward model, an equation system is built up in terms of measured projections and then solved by the reconstruction algorithm to obtain the object image. Its combination with compressive sensing (CS) theory [11], [12] gives an inspiration to the CT reconstruction from the incomplete projection data. With the CS theory substantially reduced projections are actually needed to erect a smaller equation system from which a sparsest solution is found with high quality. However, the ART is one kind of iterative reconstruction approach that computes the object image through a loop of steps. If execution is sequential, the 3D reconstruction could take tens or even hundreds of hours. A general idea to speed up this processing procedure is computing in parallel. In the case of LSDF, the connected computing resources will play an important role in accelerating the reconstruction procedure and providing result to users in near real time. In this paper, the CT reconstruction algorithm of CS-based algebraic technique is firstly introduced simply. Afterwards, the performance of 3D computed tomography reconstruction on the LSDF computing resources is investigated.

II. INFRASTRUCTURE: LSDF AND COMPUTING CLUSTER

A. Hardware

The LSDF hardware infrastructure, hosted at the Steinbuch Centre for Computing (SCC) at KIT, provides a compute cluster of 58 HP ProLiant DL1000 nodes with dual quad-core Intel processors and 36 GB memory and two 1 TB harddisks to allow data intensive computing close to the data. To offer flexibility to a large extend, these resources can be operated either as a Cloud infrastructure by using virtual machines or as a Hadoop [13] compute cluster. For Cloud computing, all virtual machines can be fully customized by the user to use specialized software environments and licenses for the data analysis. For more general use cases MapReduce-enabled analysis provided by the Hadoop framework can be utilized. The LSDF Hadoop installation offers the MapReduce [14] job submission component and a HDFS filesystem comprising 110 TB disk capacity. For direct user access the Hadoop filesystem is made available via a FUSE [15] mount. For access to large scale data each cluster nodes has access to a shared filesystem where the LSDF scientific data is stored. This is a GPFS [16] filesystem running on high-end storage hardware (DDN SFA10000), and mounted from all the processing nodes via the NFS protocol.

B. Software

All compute cluster nodes are running the Scientific Linux 5 [17] operating system. In addition to the base system, almost arbitrary software components can dynamically be configured from the runtime environment, using the Environment Modules [18] tool. This tool is used to define customized environment variables and scripts for each software component deployed in the cluster to enable/disable it dynamically. The user, or jobs by the user, only has to load the required modules by executing simple commands. Thereby, each user and job gets a clean runtime environment and results are reproducible at all time. Other advantages are that different versions of the same software package can be supported at the same time and that all supported softwares are managed centrally as part of the cluster administration and not by the users themselves.

III. CS-BASED ART METHOD

A. ART forward model

In computed tomography, the radon transform [10] is widely used to illustrate the creation of projections from the object image. A ray in this forward model is a straight line going through the object, and projection measured with this ray is obtained from integral of a function over the straight line. However this ray style does not match the reality of rays. A more accurate ray model will integrate the practical geometry of sensors into the forward imaging, which leads to a higher quality reconstruction of the object image. Different from the radon transform, a new ray model in ART possesses two features: 1) a ray is of certain width $\tau$; 2) a gap exists between the neighbor rays. To illustrate this, shaded rays are showed in Fig. 2. Generally, ART assumes that the cross section
of the object consists of an array of unknowns representing the X-ray absorption coefficients of the object, denoted by \( s_j, (j = 1, 2, \ldots, N) \), as shown in Fig. 2. A sum version of line integral is used here to calculate these projections. The intersection area represents the contribution of image cell \( s_j \) to the \( i^{\text{th}} \) ray projection. It is expressed as a weighting factor \( a_{ij} \), defined as
\[
a_{ij} = \frac{\text{intersection-area}}{\text{cell-area}}. \tag{1}
\]

Let \( p_i \) be the projection measured with the \( i^{\text{th}} \) ray, and it equals
\[
p_i = \sum_{j=1}^{N} a_{ij} s_j, \quad i = 1, 2, \ldots, M, \tag{2}
\]
where \( M \) is the total number of rays. These \( M \) projection equations build up a equation system, which can be simply written as a matrix form
\[
p = \mathbf{A}s, \tag{3}
\]
where \( s = \{s_1, s_2, \ldots, s_N\} \) is the image vector; \( p = \{p_1, p_2, \ldots, p_M\} \) is the set of all measured projections; \( \mathbf{A} \) is viewed as the system transform matrix of object image to the measurement space, and determined by the forward model. The size of \( \mathbf{A} \) is \( M \times N \).
\[
\mathbf{A} = \begin{bmatrix}
a_{11} & a_{12} & \ldots & a_{1N} \\
a_{21} & a_{22} & \ldots & a_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
a_{M1} & a_{M2} & \ldots & a_{MN}
\end{bmatrix}. \tag{4}
\]

Reconstruction algorithm aims to solve the image vector from the equation \( \mathbf{A}s = \mathbf{p} \) given the system transform matrix and practical measurements. For this purpose, the iterative method is commonly used [19].

B. CS-based ART algorithm

Compressive sampling-based algebraic reconstruction technique integrates the concept of compressive sampling into the CT reconstruction field. CS theory tells that one can restore an image from less samples than traditional image recovery methods [12]. With the CS theory, a decreased number of projections are actually needed for reconstruction with high quality. The condition to apply CS successfully is that the desired image should have a sparse representation in a convenient transform domain. It is the fact for many natural images [20]. Again \( s \) is the \( N \times 1 \) column vector in \( \mathbb{R}^N \) representing the object image. An expansion of \( s \) in an orthonormal basis \( \Psi \) can be written as follows:
\[
s = \Psi x, \tag{5}
\]
where \( \Psi \) is the \( N \times N \) matrix with \( N \) column vectors \( \{\Psi_i\}_{i=1}^{N} \) in \( \mathbb{R}^N \). \( x \) is also an \( N \times 1 \) column vector, viewed as the representation of \( s \) in the basis \( \Psi \). If all but a few of the elements of \( x \) are zero or almost zero, we can say that \( x \) is a sparse representation of \( s \).

For parallel-beam CT imaging, ART forward model gives a discrete linear system showed in equation (6). Substituting equation (5) in (3), projection measurements \( y \) can be written as
\[
y = \mathbf{A}s = \mathbf{A}\Psi x. \tag{6}
\]
System matrix \( \mathbf{A} \) is obtained from the imaging model of CT. The problem is how to determine the basis \( \Psi \) to find the sparse representation \( x \). The often used method is called Total Variation (TV) [21]–[23], which is related to the sparse gradient of object image. Instead of solving equation (5) directly, CS-based ART algorithm is to find the solution \( x \) when the \( x \) is the sparsest in equation (5).

IV. IMPLEMENTATION

Based on one experimental data set, we try to demonstrate the CS-based ART algorithm and its performance on computing cluster of LSDF for 3D CT reconstruction. Our study includes two terms. First, the algorithm has to be proved to be of high-quality reconstruction for the fast tomography of ANKA, and secondly the LSDF connected with data intensive computing resources can effectively process the 3D reconstruction and provide results to users in near real time. In this section, detailed setups about our implementation are described.

A. Experimental data set

The experimental data set used in our implementation is achieved from Topo-Tomo beamline of ANKA. It is an intensively sampled result of an insect in 1502 directions between 0° and 180°, and exported as a series of TIF images of size 1024 \times 1024, taking up 3.3 GB of storage. In order to apply the CS-based ART approach and prove this approach is feasible when the object is sampled sparsely, only a sub-data of the experimental data set is needed. The way we use is to decrease the angular resolution of the data set. Thus a sub-data set in 60 angles is extracted uniformly from these 1502 ones, ensuring that the information in full directions is included in the subset at the maximum extend. Consequently, the size of one sinogram decreases to 1024 \times 60, only 4% of the original data 1024 \times 1502. Table I shows the major parameters about the original data set and sub-data set. The
size of reconstruction image is set as $512 \times 512$. CS-based ART algorithm is applied to every cross-section in the sub-data set and executed on Hadoop cluster. A 3D volume of size $512 \times 512 \times 1024$ is achieved finally. The reconstruction result from CS-based ART algorithm is of high quality, which is reported in the next chapter.

**B. Data parallel computing**

Our experimental data set consists of 1024 slices, each reconstructed using the same algorithm. This reconstruction work is time-consuming if it is performed sequentially. With the Google MapReduce programming model [14] of Hadoop, we can distribute the work around the cluster. In our implementation, 37 nodes in the Hadoop cluster are available. Each node in the cluster has the ability to process up to six mapping tasks in parallel. Thus we divide the whole work, 1024 slice reconstructions, into a number of mapping tasks, which are distributed to different nodes automatically by Hadoop. Each mapping task handles a similar number of slices sequentially, but the executions of different mapping tasks are parallel. When the amount of mapping tasks is less than or equal to the nodes, only one mapping task is executed on each node. If the total amount of tasks is one, it means all the slice reconstructions are processed sequentially. Fig. 3 demonstrates the relationship of different slices, mapping tasks and nodes. In order to find out the parallel performance of Hadoop cluster in processing 3D CT reconstruction, two types of measurement ways are designed as follows.

![Diagram of Data Parallel Computing for 3D CT Reconstruction](image)

Fig. 3: Data parallel computing for 3D computed tomography reconstruction

1) **One mapping task per node**: Let the number of mapping tasks be less than 37, so that only one mapping task at most is executed on one node. Thus the number of mapping tasks is equal to that of nodes actually used. In this measurement, the implementation times are recorded according to different number of nodes. Good speedup factors are supposed to be obtained compared with the sequential execution of these 1024 reconstructions, as all the mapping tasks run completely in parallel. In this implementation we will see that the parallel computing architecture of Hadoop cluster are really suitable for the 3D CT reconstruction.

2) **Multiple mapping tasks per node**: The computing power of Hadoop cluster can not be fully exploited if each node processes only one task. As maximum six mapping tasks can run on one node simultaneously, we design to measure the implementation time according to the number of mapping tasks per node. In this measurement, 37 nodes are all used. The total number of mapping tasks is 37 multiplied by the number of tasks per node. In this way, the speedup factor increases further. It is expected that the whole reconstruction job can be finished in several minutes.

Times in seconds taken by the whole job is measured directly, from which other two essential factors, speedup and efficiency, are generally calculated to evaluate the performance of parallel computing. The speedup and efficiency factors are defined as follows:

\[
\text{speedup}(n) = \frac{\text{time-serial}}{\text{time-parallel}(n)} \tag{7}
\]

\[
\text{efficiency} = \frac{\text{speedup}(n)}{n} \tag{8}
\]

where $n$ is the number of nodes participating in parallel computing. When assessing the parallel computing performance, these factors are good indicators of computing speed and efficiency increase. Measurements are given in the next chapter.

**V. RESULTS AND ANALYSIS**

**A. CS-based ART reconstruction results**

Fig. 4 shows the reconstruction results of one slice from the experimental data set, including the reconstruction image from CS-based ART using sub-data set with the sinogram size $1024 \times 60$ showed in Fig. 4b and other two compared images showed in Fig. 4a and Fig. 4c respectively. Fig. 4a is the referenced reconstruction image which is obtained from FBP algorithm using the original data set with the sinogram size $1024 \times 1502$. On the contrary, Fig. 4c comes from the same reconstruction algorithm of FBP but using the sub-data set. As we can see, even though the size of the sub-data set is only 4% of the original data set, the quality of image from CS-based ART algorithm is still comparable with that of the referenced reconstruction. However the image from FBP method shows many reconstruction artifacts. Thus the CS-based ART algorithm is proved to be a good solution to address the serious aliasing problem of FBP in case of incomplete projection data for ultra fast tomography.
B. Parallel computing performance

Table II gives the times in seconds needed to reconstruct the 3D volume when different amounts of nodes are used. For each case, the time is measured for ten times, and the average is showed in the table. When only one node is used, the reconstruction process is sequential which takes 39727s (around 11 hours) for all slices. When more nodes are used, the time decreases rapidly and high speedup and efficiency factors are obtained.

Fig. 5 shows three diagrams in which the measured time, speedup and efficiency factor are plotted respectively, and compared with the expected counterparts. The expected time is calculated by time-sequential/n, where n is the number of nodes. The expected speedup factor is equal to the number of nodes n. As we can see that the measured time and speedup factor are quite closed to the expected ones, which means the parallel computing is of high performance when different nodes are used to accomplish the 3D reconstruction job. The reason is that the computing resources of each node is dedicated to one mapping task and thus the parallel overhead is little.

<table>
<thead>
<tr>
<th>Nodes n</th>
<th>Time (s)</th>
<th>Speedup</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39727</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6681</td>
<td>5.94</td>
<td>0.9910</td>
</tr>
<tr>
<td>12</td>
<td>3415</td>
<td>11.63</td>
<td>0.9692</td>
</tr>
<tr>
<td>18</td>
<td>2326</td>
<td>17.07</td>
<td>0.9486</td>
</tr>
<tr>
<td>24</td>
<td>1747</td>
<td>22.73</td>
<td>0.9473</td>
</tr>
<tr>
<td>30</td>
<td>1400</td>
<td>28.37</td>
<td>0.9459</td>
</tr>
<tr>
<td>36</td>
<td>1180</td>
<td>33.66</td>
<td>0.9350</td>
</tr>
</tbody>
</table>

According to the second measurement described in the previous chapter, more than one mapping tasks are set for each node. As showed in Table III, the average time of ten measurements in each line is given. When the number of mapping tasks increases from one to six, the time took by the 3D reconstruction decreases from 1107s to 330s with the speedup factor going up to 120. Different from the first measurement way, several mapping tasks in one node share the same computing resources, such as memory and processing cores. As a result, slower growth of computing performance is displayed when compared with the first measurement way. This becomes obvious when the times and speedup factors are compared with their expected counterparts. As showed in Fig. 6, the distance between the measured line and the expected counterpart becomes larger with the increase of mapping tasks per node. However still a great speedup factor 120 is achieved finally, and only 330 seconds (less than 6 minutes) is needed to accomplish the 3D reconstruction. The results suggest that with the data intensive computing facility, 3D CT reconstruction can be finished in a near instantaneous time.

<table>
<thead>
<tr>
<th>Mapping tasks per node</th>
<th>Mapping tasks</th>
<th>Time (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>1107</td>
<td>35.85</td>
</tr>
<tr>
<td>2</td>
<td>74</td>
<td>650</td>
<td>61.08</td>
</tr>
<tr>
<td>3</td>
<td>111</td>
<td>488</td>
<td>81.39</td>
</tr>
<tr>
<td>4</td>
<td>148</td>
<td>417</td>
<td>95.20</td>
</tr>
<tr>
<td>5</td>
<td>185</td>
<td>360</td>
<td>110.2</td>
</tr>
<tr>
<td>6</td>
<td>222</td>
<td>330</td>
<td>120.0</td>
</tr>
</tbody>
</table>

VI. DISCUSSION AND CONCLUSION

In this paper, the data intensive computing method is designed for the large amounts of tomographic data from ANKA. Based on the large scale data facility at KIT, an automatic workflow is built up to combine the tomography beamline of ANKA. In this workflow, this paper focuses on the parallel computing of 3D computed tomography reconstruction. A dedicated reconstruction algorithm is presented for tomographic data set which includes fewer projections, such as data from ultra fast tomography. It is based on a more precise forward model of ART algorithm and compressive sampling theory.
An experimental data set from the Topo-Tomo beamline of ANKA is used and shows good reconstruction result. Even much less projection data is used, the CS-based ART can achieve a comparable reconstruction result with the referenced reconstruction, reducing largely the artifacts in the FBP reconstruction.

With the LSDF and the connected computing cluster, data parallel computing of 3D CT reconstruction is performed in distributed mapping tasks executed in parallel. Two measurement types are used. The first one suggests that the measured time and speedup factor are of good scale in case of one mapping task per node. The time needed for 3D reconstruction with 1024 slices decreases quickly. In the second measurement, the scale becomes worse. However, when maximum
amount of mapping tasks per node is set, a time less than six minutes is actually needed for the data intensive computing of 3D CT reconstruction. Still the sequential reconstruction takes more than ten hours. The speedup factor reaches up to 120 with 37 nodes available. A big advantage of this system is its good scalability in processing such big data-sets. When more cluster nodes are used, better time performance can be achieved.

The data intensive computing of 3D computed tomography reconstruction on the computing cluster of LSDF shows good result and high speed. The reconstruction framework will be integrated into the LSDF workflow for the real-time processing of data sets from tomography beamline of ANKA.

VII. ACKNOWLEDGEMENTS

This work is supported by the China Scholarship Council (CSC).

REFERENCES