Improving Medical Imaging Performance on the
Intel® Xeon® Processor 5500 series

Abstract

Medical Imaging is a hotly contested field where winning products maximize healthcare quality by providing the best images in the shortest amount of time to assure accurate diagnosis and treatment for patients while maximizing the efficiency of staff and facilities. The evolution of medical imaging has resulted in very large data sets and increasingly complex algorithms putting ever increasing demands on processing power and opening many opportunities for performance optimization. This brief describes work done to optimize an image reconstruction algorithm for the Intel® Xeon® processor 5500 series that resulted in a 50x\(^1\) speedup\(^1\) compared to the original code. The discussion includes characterization of the application (computation vs. data access), performance work (threading and cache optimization) and performance issues (threading overhead and data access patterns).

\(^1\) All performance results measured on Intel® Xeon 5560 processor, 2.8GHz, 12GB DDR3, 64-bit Linux
## Contents

Contents........................................................................................................................................... 1  
Introduction........................................................................................................................................ 2  
Introduction to Medical Imaging Modalities..................................................................................... 2  
Image Reconstruction Code............................................................................................................... 6  
Analysis of the Code........................................................................................................................... 7  
Intel® Xeon® 5560 Processor Results1.............................................................................................. 8  
Future Performance Issues................................................................................................................ 8  
Summary........................................................................................................................................... 9  
Additional Resources......................................................................................................................... 9  
Media Credits.................................................................................................................................... 9  
About the Authors............................................................................................................................... 10
Introduction

Medical imaging scanners come in a wide variety of types but most of them share something in common: they produce huge data streams that need to be processed to produce diagnostically useful images. Some of the familiar types of imaging systems (or modalities as they're called) are Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET) and Single Photon Emission Computed Tomography (SPECT).

As you can imagine, image quality is critical to allowing for accurate diagnosis and in many cases the time to obtain the images plays a critical role in delivery of proper care to the patient. The time it takes a scanner to collect data can be quite short compared to the time it takes to process that data into images but it’s often necessary to see the images before letting the patient leave to be sure the right images were obtained. Clearly, this processing time limits the number of imaging studies that can be performed in a day and affects efficiency of staff and facilities (and therefore return on investment). Performance is critical in many ways.

The amount of processing required depends on the particular modality but has typically required server clusters or even special purpose hardware to provide images in a reasonable amount of time. We have recently seen Intel® Xeon processor 5500 series cut the number of servers in half (compared to previous processor generations) but as scanner resolutions increase, the need for processing speed continues to grow.

Introduction to Medical Imaging Modalities

Medical imaging scanners (referred to as modalities in the healthcare industry) don’t produce pictures; they produce huge data streams that need to be processed to reconstruct diagnostically useful images. The specific modality determines the details of the needed reconstruction processing.
CT (or Computed Tomography) scanners are based on an x-ray source on one side of a ring (like a donut) directed toward an array of x-ray detectors on the opposite side. As the donut rotates, the subject is moved through the donut’s hole. The resulting detector data represents the amount of x-ray energy that made it through the body along a sequence of rays between the detector and the x-ray source. This detector data is used to reconstruct images (slices) representative of the subject’s internal structures. CT is particularly sensitive to bones. In Figure 1 we see a cross section that shows ribs, the spine and some internal organs.
MRI (Magnetic Resonance Imaging) involves measurement of energy state changes in tissues in the presence of intense magnetic fields and pulsed radio frequency fields. An array of sensors detects these energy changes and post-processing results in images of internal tissue structures. MRI is very good at distinguishing between different tissue types. Figure 2 shows a cross section of a human skull detailing the brain, eyes, nose, mouth and other structures.
PET and SPECT are nuclear imaging techniques that require the introduction of radioactive agents into the body and produce images based on how these agents are taken up by metabolic processes in tissues and muscles. The details of these two modalities are different but both produce images reflecting metabolic activity and inactivity organs. Figure 3b compares PET...
images of brains from a normal subject and a patient with Parkinson’s disease. Figure 4b shows a SPECT “slice” of a heart, where the “bright” spot indicates cardiac muscle is receiving and metabolizing the radioactive tracer indicating viable heart tissue. The PET and SPECT images both lack detail compared to CT or MRI due to the very noisy nature of the data and benefit from much more computationally intensive reconstruction techniques to deal with the noise.

Figure 4a

Figure 4b

SPECT Scanner and Reconstructed Image of Human Heart Demonstrating Blood Perfusion

Image Reconstruction Code
The initial single threaded implementation of a new algorithm was taking significantly longer to execute compared to conventional reconstruction. This code was analyzed and optimized for Intel® architecture. All work was performed on a server platform with two Intel® Xeon® Processors X5560 running at 2.8GHz and with 12GB memory. The operating system was 64-bit Linux®.

**Analysis of the Code**

Inspection of the code showed it was structured into sequential segments that performed operations on many small subsets of a larger dataset (several gigabytes). The subsets were extracted non-uniformly from the large dataset according to some complex calculations. The code segments performed only a few simple arithmetic operations on the data subsets. We expected memory access to be a limiting factor due to the non-uniform addressing (not prefetcher or cache friendly) and it was unclear by inspection what cache efficiency might be.

Using the Intel® VTune™ Performance Analyzer, we determined that the main code segments accounted for about 90% of the execution time (total clocks). In these “hot spots” CPI (clocks per instruction) was very high. Taking a closer look at cache performance we found that L2 cache misses were reasonable. So what was causing the high CPI? Digging deeper we found we were suffering a 50% page walk penalty. This made sense since our working set was much larger than the span of the Data TLB with 4kB pages. The Intel Xeon Processor X5560’s support for 2MB huge pages makes it possible to address 500 times as much data per DTLB entry, significantly reducing the number of page walks necessary to address large data sets. Implementing 2MB huge pages made page walks insignificant.

Further examination of the data indices showed not only a non-uniform stride, but data accesses scattered all over the large dataset. We rearranged the data so indices always increased, and also to maximize cache reuse from subset to subset. This allowed the processor’s data prefetchers to work better in addition to improving cache reuse. The combined effect of huge pages and data reordering provided a 2x performance speedup.

To take maximum advantage of the target platform, we switched from the GNU GCC* compiler to the Intel® C/C++ compiler. This change resulted in an additional 1.2x speedup.

We developed a data parallel threading model by splitting the number of data elements among the available threads. It was necessary to synchronize across threads in several places. We wanted to use lightweight synchronization so we utilized very simple spin locks.
### Intel® Xeon 5560 Processor Results

<table>
<thead>
<tr>
<th>Action taken</th>
<th>Speedup¹.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huge pages + data reordering</td>
<td>2x</td>
</tr>
<tr>
<td>Intel® C/C++ compiler</td>
<td>1.2x</td>
</tr>
<tr>
<td>Multi-threading (16 threads)</td>
<td>10x</td>
</tr>
<tr>
<td>Performance tuning hotspots</td>
<td>1.7x</td>
</tr>
<tr>
<td>Overall</td>
<td>~50x</td>
</tr>
</tbody>
</table>

Table 1

Table 1 summarizes what we were able to achieve on the Intel Xeon 5560 processor.

- Implementing huge pages and reordering data accesses resulted in a 2x speedup.
- Switching from GNU GCC* to the Intel C/C++ compiler gave another 20% performance boost.
- The data parallel threading scaled well and gave a 10x speedup with 16 threads on Intel Xeon Processor X5560 (SMT enabled).
- Additional coding changes at hotspots gained another 1.7x speedup.
- Overall, a speedup of about 50x was realized.

### Future Performance Issues

For the data sets available to this work, we noticed a leveling off in scalability as the number of threads increased beyond 8 to 12 threads. We were able to explain this by looking at the work done per thread and comparing that to the thread synchronization overhead. Although a very simple spin-lock was used to synchronize threads, the number of computations done by each thread decreased proportionally to the number of threads while the synchronization time...
increased in the same proportion. We expect that the synchronization overhead will exceed the
computation time somewhere between 20 and 28 threads limiting the effectiveness of this
solution on systems with larger thread counts. For these future systems perhaps a combination
of data and process parallel approaches will provide additional performance.

Summary

Methods for improving the quality of medical images are well known but have been waiting for
the computer industry to deliver cost-effective compute power. This work demonstrates
significant progress toward being able to handle these computationally intensive methods by
optimizing them for the latest Intel hardware using the Intel® Compiler, Intel® Vtune™
Performance Analyzer and performance tuning methodologies.

We feel this is paving the way to better healthcare by providing improved images for accurate
and timely diagnosis at reasonable cost, improving patient care and more efficiently utilizing
staff and facility resources.

Additional Resources


• A very useful overview of tomographic reconstruction is available at:
  http://www.owlnet.rice.edu/~elec539/Projects97/cult/node1.html

Media Credits

Figures

• 1a: http://commons.wikimedia.org/wiki/File:Computer_Tomography_1.jpg , Public Domain
• 1b: http://commons.wikimedia.org/wiki/File:Neuroblastoma_liver.jpg, Public Domain
• 2a: http://commons.wikimedia.org/wiki/File:US_Navy_030819-N-9593R-228_Civilian_technician,_Jose_Araujo_watches_as_a_patient_goes_through_a_Magnetic_Resonance_Imaging,_%20MRI%29_machine.jpg, Public Domain
• 3a: http://commons.wikimedia.org/wiki/File:PET_at_NIH_Clinical_Center.jpg, Public Domain
• 3b: http://commons.wikimedia.org/wiki/File:PET_scan_Parkinson%27s_Disease.jpg Public Domain
• 4a: http://commons.wikimedia.org/wiki/File:HybridSpectCTScanner.png, Public Domain
About the Authors

**Kerry Evans** is a Software Engineer in the Digital Health Enabling Group at Intel Corporation where he helps healthcare companies optimize performance of their systems through the use of Intel technologies, performance tools and libraries. Kerry joined Intel in 2005. He received his B.S. degree in Electrical Engineering in 1975 and his M.Eng. and Ph.D. in Bioengineering in 1977 and 1979, respectively, from the University of Utah. He holds 4 US patents.

**Terry Sych** is a Staff Software Engineer in the Digital Health Enabling Group at Intel Corporation. He joined Intel in 1992, and has worked on performance analysis and software optimization of enterprise applications for the last 10 years. Terry works with enterprise software vendors analyzing, tuning, and optimizing applications. He received a B.S. degree in Computer Engineering from the University of Michigan in 1981 and an MSEE from the University of Minnesota in 1988. He holds 3 US patents.

**Kirk Dunsavage** is a Staff Software Engineer for the Digital Health Enabling Group in the Software & Services Group at Intel Corporation. Kirk's current areas of interest are platform architecture enhancements for highly parallel and multi-threaded applications, multi-core platform tuning and optimizations, high-performance computing, and digital healthcare infrastructure enhancements. Kirk joined Intel in 1995. He received B.S. degrees in Computer Science and Business Management from the University of Maryland, College Park in 1993, a M.S. in Computer Science from the University of Wisconsin – Madison in 1995, and an MBA from Arizona State University in 1998. Kirk holds 1 US patent and has several other patent applications currently in process.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR
PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL
PROPERTY RIGHT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual
property rights that relate to the presented subject matter. The furnishing of documents and other materials and
information does not provide any license, express or implied, by estoppel or otherwise, to any such patents,
trademarks, copyrights, or other intellectual property rights.

Intel products are not intended for use in medical, life-saving, life-sustaining, critical control or safety systems, or
in nuclear-facility applications.

Intel may make changes to dates, specifications, product descriptions, and plans referenced in this document at
any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or
"undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or
incompatibilities arising from future changes to them.

The Intel® PXA27x Processor and Intel® PXA9xx Cellular Processor Family may contain design defects or errors
known as errata which may cause the product to deviate from published specifications. Current characterized
errata are available on request.

This Binary Library ("Software") is furnished under license and may only be used or copied in accordance with the
terms of that license. No license, express or implied, by estoppel or otherwise, to any intellectual property rights
is granted by this document. The Software is subject to change without notice, and should not be construed as a
commitment by Intel Corporation to market, license, sell or support any product or technology. Unless otherwise
provided for in the license under which this Software is provided, the Software is provided AS IS, with no
warranties of any kind, express or implied. Except as expressly permitted by the Software license, neither Intel
Corporation nor its suppliers assumes any responsibility or liability for any errors or inaccuracies that may appear
herein. Except as expressly permitted by the Software license, no part of the Software may be reproduced,
stored in a retrieval system, transmitted in any form, or distributed by any means without the express written
consent of Intel Corporation.

The source code contained or described herein and all documents related to the source code ("Material") are
owned by Intel Corporation or its suppliers or licensors. Title to the Material remains with Intel Corporation or its
suppliers and licensors. The Material may contain trade secrets and proprietary and confidential information of
Intel Corporation and its suppliers and licensors, and is protected by worldwide copyright and trade secret laws
and treaty provisions. No part of the Material may be used, copied, reproduced, modified, published, uploaded,
posted, transmitted, distributed, or disclosed in any way without Intel's prior express written permission.

No license under any patent, copyright, trade secret or other intellectual property right is granted to or conferred
upon you by disclosure or delivery of the Materials, either expressly, by implication, inducement, estoppel or
otherwise. Any license under such intellectual property rights must be express and approved by Intel in writing.

Unless otherwise agreed by Intel in writing, you may not remove or alter this notice or any other notice
embedded in Materials by Intel or Intel's suppliers or licensors in any way.

JPEG, H.263, H.264, MPEG-4, G.729, G.723, AMR-WB, AMR-NB, AAC, MP3, MIDI, SBC are international standards.
Implementations of JPEG, H.263, H.264, MPEG-4, G.729, G.723, AMR-WB, AMR-NB, AAC, MP3, MIDI, SBC codecs, or
JPEG, H.263, H.264, MPEG-4, G.729, G.723, AMR-WB, AMR-NB, AAC, MP3, MIDI, SBC enabled platforms may require
licenses from various entities, including Intel Corporation.