

ImageBox 2 – Efficient and Rapid Access of Image Tiles from Whole-Slide Images Using Serverless HTTP Range Requests

Erich Bremer¹, Joel Saltz¹, Jonas S Almeida²

¹Department of Biomedical Informatics, Stony Brook University, Stony Brook, NY, USA, ²Division of Cancer Epidemiology and Genetics, National Cancer Institute, National Institutes of Health, Maryland, USA

Submitted: 09-Apr-2020

Revised: 20-Apr-2020

Accepted: 03-Jul-2020

Published: 10-Sep-2020

Abstract

Background: Whole-slide images (WSI) are produced by a high-resolution scanning of pathology glass slides. There are a large number of whole-slide imaging scanners, and the resulting images are frequently larger than 100,000 × 100,000 pixels which typically image 100,000 to one million cells, ranging from several hundred megabytes to many gigabytes in size. **Aims and Objectives:** Provide HTTP access over the web to Whole Slide Image tiles that do not have localized tiling servers but only basic HTTP access. Move all image decode and tiling functions to calling agent (ImageBox). **Methods:** Current software systems require tiling image servers to be installed on systems providing local disk access to these images. ImageBox2 breaks this requirement by accessing tiles from remote HTTP source via byte-level HTTP range requests. This method does not require changing the client software as the operation is relegated to the ImageBox2 server which is local (or remote) to the client and can access tiles from remote images that have no server of their own such as Amazon S3 hosted images. That is, it provides a data service [on a server that does not need to be managed], the definition of serverless execution model increasingly favored by cloud computing infrastructure. **Conclusions:** The specific methodology described and assessed in this report preserves normal client connection semantics by enabling cloud-friendly tiling, promoting a web of http connected whole-slide images from a wide-ranging number of sources, and providing tiling where local tiling servers would have been otherwise unavailable.

Keywords: Digital imaging, serverless cloud computing, web services, whole-slide images, World Wide Web

INTRODUCTION

Whole-slide images are frequently scanned and stored by health systems, governed by a variety of regulatory and proprietary constraints that constrain the ability to share, aggregate, or disseminate WSI files. This is an important challenge, given the increasing number of research and surveillance applications requiring the analysis of whole-slide images generated by multiple sites. The acquisition and use of whole-slide images is rapidly growing in research and pathology practice.^[1] There are a variety of open source software projects that develop and support low level software to view these images.^[2-7] Accordingly, the ability to view whole-slide images and derived data products has become a crucial component of the analytic pipeline of comprehensive biomedical studies. At the same time, visual inspection is needed to make sure that proper QA/QC remains in place, sometimes including manual inspection of combined whole-slide images and

derived data products. Therefore, we target the problem of viewing distributed collections of whole-slide images in a manner that does not require having a tiling server local to the images. An illustrative example of this is the NCI SEER surveillance program. SEER data are produced locally at over 1000 facilities. We collaborate with the SEER program that employs neural networks to generate a variety of data products including tumor-infiltrating lymphocyte and nuclear segmentation studies. For legal, contractual, and logistical reasons, these analyses often must be generated locally, even if

Address for correspondence: Mr. Erich Bremer,
Department of Biomedical Informatics, Stony Brook University,
Stony Brook, 11794 NY, USA.
E-mail: erich.bremer@stonybrook.edu

This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms.

For reprints contact: WKHLRPMedknow_reprints@wolterskluwer.com

How to cite this article: Bremer E, Saltz J, Almeida JS. ImageBox 2 – Efficient and rapid access of image tiles from whole-slide images using serverless HTTP range requests. J Pathol Inform 2020;11:29.

Available FREE in open access from: <http://www.jpathinformatics.org/text.asp?2020/11/1/29/294806>

Access this article online

Quick Response Code:



Website:
www.jpathinformatics.org

DOI:
10.4103/jpi.jpi_31_20

the sites where slides are scanned are frequently insufficiently resourced computationally. In order to address that problem, we are involved in an effort to optimize that computational process,^[8-11] both for image classification in specialized high-performance computing settings and in developing lightweight methods for viewing locally produced whole-slide images and derived data.

Advanced imaging technologies can capture extremely high-resolution images of tissue specimens, and quantitative analyses of cancer morphology using these images have shown value in a variety of correlative and prognostic studies. Our work on Oak Ridge National Laboratory (ORNL) Summit facility will generate a comprehensive multiscale mapping of cancer morphology with a dataset of more than 10,000 whole-slide tissue images from over 20 cancer types. The work will use a collection of deep-learning analysis pipelines we have developed to study, quantify, and characterize tissue structure in diseased and normal tissue specimens. These analysis pipelines generate distributions of nuclei and cells and patch-level maps of lymphocyte distributions and segmentations of tumor regions. The classification results are expected to provide first-ever representations of lymphocyte maps, nuclear characterizations, and characterizations of tumor regions on a dataset of this scale. Specifically, the unprecedented granularity of the automated classification will generate rich datasets with the potential to develop novel biomarkers to predict clinical outcome and a better epidemiological understanding of cancer subtypes and how constituent cells contribute to cancer invasion and expansion.

With the serverless execution methodology proposed, the ability to easily view whole-slide images residing in cloud and supercomputer facilities will be greatly facilitated. The critical practical advance is the reliance on local infrastructure that no longer requires deployment and management of specialized tiling image servers: Setting up tiling image servers wherever WSI images reside is neither practical nor scalable. Many applications are available and used for viewing locally based WSI images but, in order to serve image tiles on the web, an image tiling server must be employed. Modified versions of IIP which use the open-source library OpenSlide for image tile extraction require a certain level of domain expertise to build and maintain properly. Other solutions require transcoding of images into solution-specific image formats increasing storage costs and consuming valuable time for the needed conversions. Not all practices and operations are necessarily equipped to handle or afford these said requirements. WSI images also incur a large storage size requirement with smaller operations using cloud infrastructures to store, share, and access their images to reduce costs and the need for localized technical domain knowledge. The methodology proposed in this article for access to remote tiles via HTTP requests is, therefore, fitting for leveraging cloud infrastructures for WSI imaging. Furthermore, by removing the need to colocate specialized tiling services, a function replaced by the “range” parameter supported by standard web servers, the new solution paves the

way to the adoption of standard consumer-facing cloud services already used for indiscriminated storage, sharing, and backup.

METHODS

Processing WSI files in their entirety is difficult given their size because a fully decoded image would be larger than most workstations typical RAM sizes. Therefore, WSI image formats such as Aperio SVS, Olympus VSI, JPEG2000, and BigTIFF address this obstacle by tiling, allowing subportions of the image to be accessed without requiring the entire image to be decoded. The proposed methodology relies on the index of the tiled, scaled image pyramid stored within the file.

Moving WSI images to the cloud represents a problem in that an image tiling server would have to be installed locally to the cloud-based images. Using HTTP range requests,^[12] we are able to decouple the location of the WSI and the tiling server engine. This method has already been used to GeoTIFF images running under the flag of “Cloud-Optimized GeoTIFF (COG).”^[13] Although TIFFs (BigTIFF) are used extensively in cancer WSI, additional formats are required so repurposing a COG would be insufficient and limited. WSI images show up in many formats such as Aperio SVS and Olympus VSI. Due to the amount of data in scanned images, scanners will often encode these images using the JPEG format which has a lossy compression scheme. Transcoding of images from one format to another is time-consuming in our experiences, and this can typically take 30 min to more than an hour for each WSI. Even if transcoding is performed, packages such as OME Bioformats will decode all images and then re-encode when saving to the new format COG, even if the same compression and tile size is selected. If JPEG encoding is chosen, the re-encoding of the tile could cause further loss of image quality. If a nonlossy compression scheme is chosen such as ZIP or DEFLATE, the image file size can grow between 5x and 10x the original image file size. In keeping and working with the original file as it is, we save both time, image quality, and image storage.

ImageBox2 is built using the following two core software libraries: Eclipse Jetty HTTP library for all HTTP client and server functions. The second library is Open Microscopy Environment’s Bioformats^[15] for decoding of the needed WSI formats. ImageBox2 implements the IIIF^[14] interface HTTP specification.

A custom implementation of a low-level Bioformats class object was created to route Bioformats local file data calls over HTTP range requests. HTTP range requests are similar to normal HTTP requests with the exception of rather than download an entire entity, a range of bytes is specified by adding the HTTP header “range” with a value of “bytes = start-end” which would then only download bytes from < start > to < end >.

Test environment

A modified IIP server^[2,4] able to read WSI images using the OpenSlide WSI library^[16] was used for performance comparisons. IIP is an implementation of the IIIF protocol

which defines a URL pattern for accessing a specific image tile at a particular resolution using a template in the form:

```
{scheme}://{server}/{prefix}/{identifier}/{region}/{size}/{rotation}/{quality}.{format}.
```

Example:

```
http://imagebox.ebemer.com/iiif/?iiif=https://s3.amazonaws.com/ebemeribox/image.svs/131000,0,60000,10000/1024,0/default.jpg.
```

RESULTS

- Test Server #1 – Running Apache JMeter^[17] version 5.1.1. makes all simulated client calls to #2
- Test Server #2 – Running ImageBox and IIP/OpenSlide both having access to a local image copy
- Test Server #3 – Amazon S3 containing the same test WSI as in #2.

Ten clients were simulated with Apache JMeter each client making 250 random 256 × 256 tile samples. Our test image is an Aperio SVS image of size 191,352 × 91,462 pixels in size with a file size of 3.8 GB [Figure 1].

For a full interactive example see: <http://imagebox.ebemer.com/osd.html>.

DISCUSSION

In Figure 1, we show the time in milliseconds for a tile to be returned versus time for our three test cases. The older, but highly optimized, IIP performs the best for image files local to the server, followed by ImageBox2 for images local to the server, and finally ImageBox2 accessing image files remotely via http. Whereas the overhead of http does add an additional layer, it still provides very usable access times but adds the flexibility that images do not have to be local to the tiling engine. Response times improve quickly in the HTTP case as the tiling engine completes reading the remote image file headers.

In Table 1,^[18] we do a cost comparison to show that being able to view parts of remote images can significantly reduce network transfer costs as opposed to downloading entire WSI images. Cloud providers such as Amazon and Azure do not charge for transfer data into their cloud infrastructures, however, they do impose a cost on data transfer from their cloud infrastructures out of the cloud to the clients making the requests (egress costs). Since image decoding is moved to the client side, a reduction in CPU utilization on the image-hosting server would occur, however, cloud providers such as Amazon and Azure do not charge for the utilization levels of the CPUs but only for the number of CPUs that were provisioned with the original server. Whether they run at 100% or 50% the cost would be the same unless CPU loads pushed the need to add additional CPUs. We did not measure this reduction, but it is a potential saving cost-point.

Use case-ORNL Summit-our group runs a variety of deep-learning semantic segmentation and feature extraction pipelines at the Oak Ridge National Laboratory Summit supercomputing facility. The ability to remote-view the data product results of our various pipelines will allow us to carry out data product quality control during the course of a computational run.

Table 1: Our costs^[18] for ten full file downloads would have been 38.1 GB at \$0.05/GB=\$1.90, however, with ImageBox2, the 130 partial file downloads were 0.21 GB=\$0.01

Service	Response	Comparison description
ImageBox Amazon S3	120 ms/tile	ImageBox accessing tiles on an Amazon S3 (Server #3) hosted image
ImageBox Local	50 ms/tile	ImageBox accessing tiles from test WSI local to itself on Server #2
IIP/OpenSlide	25 ms/tile	IIP/OpenSlide accessing tiles from the same local file on Server #2

WSI: Whole-slide images

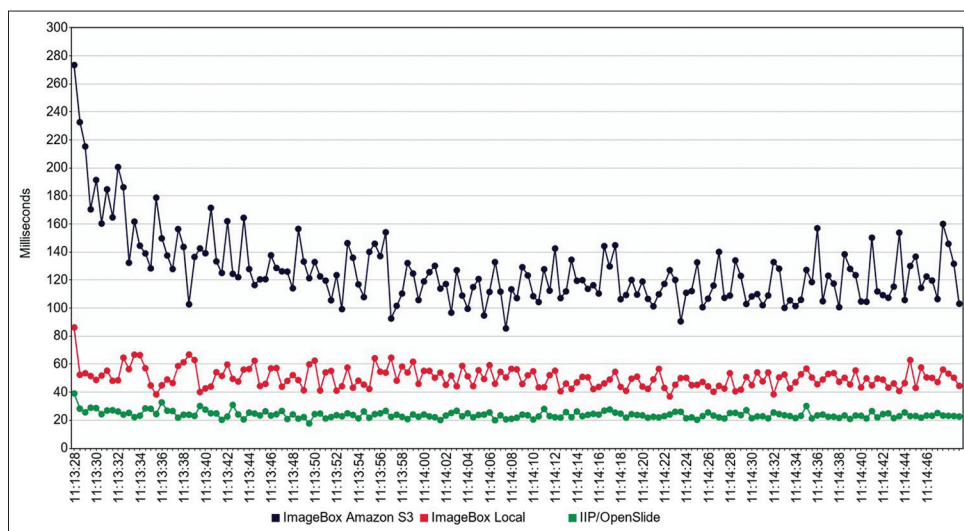


Figure 1: Response graphs for time for image tile return

The supercomputer generates results extremely quickly. For instance, we are able to generate results for a cohort of 1000 patients in roughly 2 h rather than the many weeks otherwise required on standard GPU cluster facilities. Moving large amounts of data between ORNL facilities and remote research groups such as Stony Brook University is impractically time-consuming. Because these are computational processes with efficiencies that vary between datasets, there is a constant need to interactively monitor progress of the computations and to assess the quality of the results being produced. This infrastructure allows our pathologists and biomedical scientists to carry out this near real-time performance assessment.

In a test, the results of a deep-learning segmentation pipeline were stored in a web-accessible folder provided by ORNL systems.^[19] A remote ImageBox server was deployed at Stony Brook and an OpenSeadragon-based viewer to display The Cancer Genome Atlas WSI image that was local to that ImageBox instance and simultaneously overlay the ORNL-based segmentation results for that image using HTTP range requests [Figure 2].

Currently, in all of our work, access has been restricted to unprotected, publicly available HTTP access points. However, it would be useful and is a planned feature for ImageBox2 to add authentication capabilities such as OAuth2, OpenID, and token-based authentications in order to add access capabilities to protected image assets.

CONCLUSIONS

The new approach to distributed image tile serving is able to pull tiles of varying sizes and resolutions directly over the web using HTTP range requests. This solution is both scalable and safe because it does not require customized software installation and does not circumvent the access provisions set by the web server for the image file being served. The ubiquity of HTTP range operations is a byproduct of the development of modern cloud infrastructure configured for

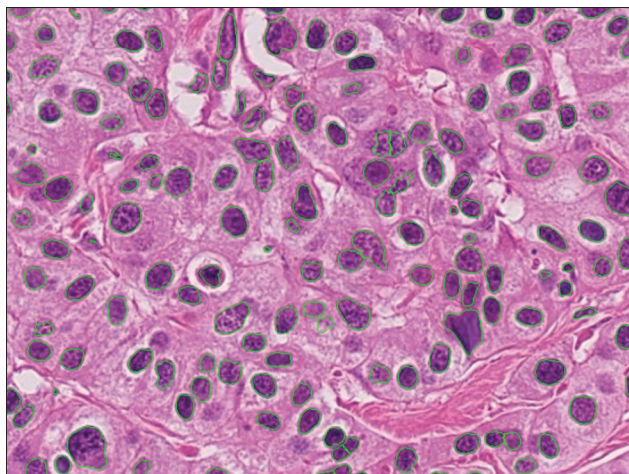


Figure 2: Remote segmentation results overlaid dynamically on a whole-slide image using HTTP range requests

equally ubiquitous data intensive API ecosystems such as those now being advanced by Data Commons, as illustrated by the National Cancer Institute (NIH/NCI) Genomic Data Commons, as well as by reference repositories such as The Cancer Imaging Archive. The solution proposed here is in line with that cloud-hosted granular API delivery model, devising a mechanism where one tiling imaging server can provide subtiles from WSI images all over the web. This solution is validated by an accompanying application, ImageBox2, which advances earlier work on “safe image cloudification” by ImageBox.^[20] It is also informed by our recent work on granular orchestration of stateless Application Programming Interfaces at Data Commons scale.^[21] The test results below illustrate a use case where, in addition to the advantageous security and scalability, the costs associated with the proposed solution are up to one hundredth of the conventional approach [Table 1].

From the response time graph, one can see that the HTTP range requests incur a time penalty as an HTTP request must be generated in order to retrieve data as opposed to a local file call for the same data. The ability to then access tiles from cloud-hosted images makes up for that lag with twofold increases in efficiency measured by the ability to stream only the tiles needed, reducing cloud network transfer costs and avoiding file duplication steps altogether. Although response time is slightly more than twice that of a local copy, it is more than ample for human-interactive displays such as OpenSeadragon.^[22] HTTP range requests are supported by most cloud providers, including Amazon, Box, Dropbox, and most HTTP server implementations such as Apache HTTP server and NGINX.

With the advent of WebAssembly (Wasm), it is possible to compile languages such as C, C++, and Java to a binary instruction set that runs in browsers. In Wasm 1.0, garbage collection is not currently supported, so Java compilation to Wasm is problematic. Garbage collection is slated to be added to the next Wasm specification. ImageBox2 is written in Java, and we are working to move it into a client-only configuration using Wasm technology.

Financial support and sponsorship

NCI U24CA215109 and NCI U24CA180924.

Conflicts of interest

There are no conflicts of interest.

REFERENCES

1. Aeffner F, Zarella MD, Buchbinder N, Bui MM, Goodman MR, Hartman DJ, *et al.* Yuil-Valdes, and Douglas Bowman Introduction to digital image analysis in whole-slide imaging: A white paper from the digital pathology. *Assoc J Pathol Inform* 2019;10:9.
2. caMicroscope Iipimage Containerized IIP. Available from: <https://github.com/camicroscope/iipImage>. [Last accessed on 2019 Dec 04].
3. Goode A, Gilbert B, Harkes J, Jukic DM. Satyanarayanan-OpenSlide: A vendor-neutral software foundation for digital pathology. *J Pathol Inform* 2013;4:27.
4. Ruven Pillay – IIPImage. Available from: <https://iipimage.sourceforge.io/>. [Last accessed on 2019 Dec 04].

5. Bankhead P, Loughrey MB, Fernández JA, Dombrowski Y, McArt DG, *et al.*, QuPath: Open source software for digital pathology image analysis. *Sci Rep* 2017;7:16878.
6. Tony Collins-ImageJ for Microscopy – BioTechniques; 16 May 2018.
7. Allan C, Burel JM, Moore J, Blackburn C, Linkert M, Loynton S, *et al.* OMERO: Flexible, model-driven data management for experimental biology. *Nature Methods* 2012;9:245-53.
8. Saltz J, Gupta R, Hou L, Kurc T, Singh P, Nguyen V, *et al.* Rebecca and others Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. *Cell Rep* 2018;23:181-93.
9. Gupta R, Kurc T, Sharma A, Almeida JS, Saltz J. The Emergence of pathomics. *Curr Pathobiol Rep* 2019;7:73-84.
10. Cooper LA, Demicco EG, Saltz JH, Powell RT, Rao A, Lazar AJ. Pan cancer insights from the cancer genome atlas: The pathologist's perspective. *J Pathol* 2018;244:512-24.
11. Le H, Gupta R, Hou L, Abousamra S, Fassler D, Kurc T, *et al.* Utilizing automated breast cancer detection to identify spatial distributions of tumor infiltrating lymphocytes in invasive breast cancer. *arXiv Preprint arXiv* 2019; 2019 May 26; Available from: <https://arxiv.org/abs/1905.10841>.
12. Fielding RT, Lafon Y, Reschke JF. Hypertext Transfer Protocol (HTTP/1.1): Range Requests-Internet Engineering Task Force (IETF); June, 2014. Available from: <https://tools.ietf.org/html/rfc7233>. [Last accessed on 2020 Mar 27].
13. Holmes C. Even Rouault-Cloud Optimized GeoTIFF. Available from: <https://www.cogeo.org/>. [Last accessed on 2019 Dec 03].
14. Snyderman S, Sanderson R, Cramer T. The International Image Interoperability Framework (IIIF): A community & Technology Approach for Web-Based Images-The IS&T (digital) Archiving Conference Archiving 2015 Final Program and Proceedings;2015 May, p.1621(6). Available from: <https://www.ingentaconnect.com/content/ist/ac/2015/00002015/00000001/art00005> and <https://iiif.io/>. [Last accessed on 2020 Mar 27].
15. Linkert M, Rueden CT, Allan C, Burel JM, Moore W, Patterson A, *et al.* Metadata matters: Access to image data in the real world. *J Cell Biol* 2010;189:777-82.
16. Goode A, Gilbert B, Harkes J, Jukic D, Satyanarayanan M. OpenSlide: A vendor-neutral software foundation for digital pathology. *J Pathol Inform* 2013;4:27. Available from: <https://github.com/openslide/openslide>.
17. Apache JM. The Apache Software Foundation. Available from: <https://jmeter.apache.org/>. [Last accessed on 2019 Dec 12].
18. Amazon S3 Simple Storage Service Pricing Amazon Web Services. Available from: <https://aws.amazon.com/s3/pricing/>. [Last accessed on 2019 Dec 03].
19. Oak Ridge National Laboratory Data Storage and Transfers. Available from: <https://www.olcf.ornl.gov/for-users/system-user-guides/rhea/file-systems/>. [Last accessed on 2020 Mar 27].
20. Bremer E, Kurc T, Saltz J, Almeida JS. Safe “cloudification” of large images through picker APIs. *AMIA Annu Symp Proc*. 2016:342-51. [PMID:28269829].
21. Almeida JS, Hajagos J, Saltz J, Saltz M. 2019. Serverless OpenHealth at data commons scale—traversing the 20 million patient records of New York’s SPARCS dataset in real-time. *PeerJ* 7:e6230 [PMID:30671301].
22. OpenSeaDragon An Open-Source, Web-Based Viewer for High-Resolution Zoomable Images, Implemented in Pure JavaScript, for Desktop and Mobile. Available from: <https://openseadragon.github.io/>. [Last accessed on 2019 Dec 01].