

# Distributed Architecture for a Peer-to-Peer-Based Virtual Microscope

Andreas Jaegermann, Timm Filler, Michael Schoettner

► **To cite this version:**

Andreas Jaegermann, Timm Filler, Michael Schoettner. Distributed Architecture for a Peer-to-Peer-Based Virtual Microscope. Jim Dowling; François Taïani. 13th International Conference on Distributed Applications and Interoperable Systems (DAIS), Jun 2013, Florence, Italy. Springer, Lecture Notes in Computer Science, LNCS-7891, pp.199-204, 2013, Distributed Applications and Interoperable Systems. <10.1007/978-3-642-38541-4\_18>. <hal-01489459>

**HAL Id: hal-01489459**

**<https://hal.inria.fr/hal-01489459>**

Submitted on 14 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Distributed Architecture for a Peer-to-Peer-based Virtual Microscope

Andreas Jaegermann<sup>1</sup>, Timm J. Filler<sup>1</sup>, and Michael Schoettner<sup>2</sup>

<sup>1</sup> Department of Anatomy, University of Duesseldorf

<sup>2</sup> Department of Computer Science, University of Duesseldorf

**Abstract.** Virtual microscopes are commonly used in medical education. They provide a platform for distributing whole slide images (WSI) with several GB size to exploring students. Even in courses with a few hundred students and dozens of WSI the network traffic may be high, but it will vastly increase, when the system is opened to access from the Internet. The same applies to user-generated content like interactive annotations (each student generates approx. 200 labels per term). In a collection that consists of several thousand WSI, which need to be annotated for training or quiz-based purposes, there will be millions of user contributions. In an abstract view users navigate through a universe of WSI and annotations and may meet other users watching the same or related WSI. This paper presents a distributed architecture build on PathFinder for Internet-based virtual microscopy addressing the challenges of distributing tightly connected data chunks on an overlay network consisting of random graphs.

## 1 Introduction

A virtual microscope is a system, which provides digitalized slides for a large number of simultaneously accessing clients, similar to geographic applications like Google Maps. Slide scanners are used to digitize glass slides and create proprietary slide file formats depending on the scanners' manufacturer [6]. The file size varies with the dimensions of the scanned specimen and ranges from a few MB to several GB. Virtual microscopes have been developed more than a decade ago to fit the needs of pathologists and have been adopted for educational purposes in microscopic anatomy in the last few years [5]. As the typical setup consists of a server accessing the file resources and delivering them on demand to any connected client via either proprietary communication protocols or http/ftp, the system's transmission capacity is obviously limited by the server's network connection. Usually, all systems are designed to work with classroom-sized courses and some are capable of handling up to a few hundred clients connecting simultaneously. The number of virtual microscopes increased steadily, but nearly every solution suffers from one main disadvantage: if the number of histological slides in a collection exceeds a few hundred, the labeling process to provide students with detailed, qualified information about different structures is extremely time consuming for lecturers. However, the annotations created in a

slide are essential to virtual microscopes because they extend the virtual microscope far beyond the capabilities of a real light microscope. Single cells, strata or complex tissue formations are examples of structures to be labeled. Not only are these annotations some explanations in an otherwise silent image, but rather do they provide a feedback on the personal learning progress for students. On the other hand, an increasing number of already validated annotations will help to evaluate new and varying annotations. Additionally, annotations can be used in online exams for automated evaluation.

In a common client/server setup the number of connecting clients is limited to a few hundred at maximum due to the hardware and connection requirements. Meeting more substantial requirements often leads to disproportional costs. To address these challenges we develop a system that supports a distributed architecture for transmitting digitized slides as well as a context-based user platform for real-time interaction.

To improve the scalability of virtual microscopes accompanied by an elevation of interactivity between users among themselves and the network, we transformed the typical client/server setup into a (managed) peer-to-peer architecture. Our solution is based on PathFinder, a peer-to-peer overlay network that relies on random graphs and provides an efficient combination of long range queries and single key-value lookups [4].

This paper is organized as follows. In Sect. 2 we present an overview of our distributed virtual microscope (DVM). Section 3 catches up with related work in the field of peer-to-peer systems. In Sect. 4 the advantages of PathFinder for our approach and the necessary adaptations are explained. Conclusions are presented in Sect. 5.

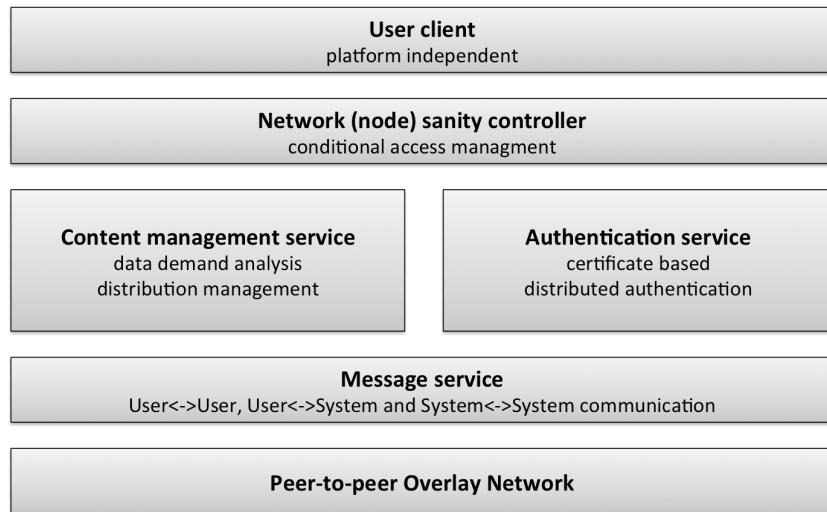
## 2 Architecture of the Distributed Virtual Microscope

The WSI are stored in tiles and a fully digitalized slide with a decent Z-stack in the highest magnification can easily produce up to 0.5 million tiles. A whole collection of a few thousand slides will lead to billions of images that need to be transmitted across the DVM along with corresponding annotations, messages and status information.

New WSI can only be stored in the network by selected nodes to ensure proper quality and compliant preparation.

Each node participating in the DVM has a standardized architecture and provides different services to the user and/or the network itself. It has to contribute a variable amount of storage capacity to be used by the DVM for the distribution of WSI. Additionally the user can store needed (or predictably needed) data. To assure data integrity the storage container acts like a black box and no standard user is allowed to modify any stored objects or store new objects.

The bottom most layer of a node (see Fig. 1) contains the overlay network responsible for handling joins and deletions of nodes. Additionally it detects and handles crashes as well as it adapts the network size. It manages the queries to physically locate data, nodes and users.



**Fig. 1.** Modular layered architecture of node in the DVM

The message service layer uses the overlay network to transmit messages between different users, users and nodes and among nodes themselves. Messages are generated in upper layers and may contain different types of information or requests. This service uses fingerprints and checkpointing to recover from a partial or complete loss of messages.

As the heart of a node the content management controller decides upon locally observed and from the neighborhood acquired data, what objects should be moved to or from the local storage container. By calculating performance values from network and storage parameters it can decide to ask neighbors for help while processing complex requests. In addition it is responsible for generating requests to increase the size of the immediate neighborhood if too much load is generated on the node and its existing neighbors.

The authentication service is tightly controlling user access and responsible for distributed authentication in case the authentication server is not available. Therefore signed key pairs from a trusted resource are created for each node. For distributed authentication each participating peer can check the certificate chain to decide if single requests should be executed or declined. As this is a only fallback mechanism to keep the network operational in case of a server failure no new accounts can be created at that time.

The next higher authority for each node is the sanity controller that has to make sure all needed local services are running and that the node has a working connection to a sane neighborhood.

On top there is just a graphical user interface to allow users to interact with the DVM. Slides can be selected, annotations can be created and messages can be send to other users.

### 3 Searchable Peer-to-Peer Overlay Network

Multicast protocols to transmit a tiled WSI were discarded as the data in DVM is no longer located at a single source and the complete transmission of a single WSI is, due to regions of interest, commonly not needed.

As we designed DVM to be a peer-to-peer (P2P) system build on hundreds of thousands of heterogeneously distributed nodes of varying performance, we were in need of a search that can be used to locate specific tiles of a WSI (a specific hash can be assigned to each tile) as well as processing exhaustive searches e.g. to find WSIs matching given criteria.

The decision to create a structured or an unstructured network while designing DVM was strengthened by the benefits of distributed hash tables (DHT). DHTs offer some advantages over unstructured P2P system (like Gnutella or Napster) as they include load balancing, efficient routing and resilience against node failures.

Although the scientific community has seen many implementations and enhancements to DHTs [8, 7] all designs rely on randomizing hash functions for inserting and searching keys in the hash table. This hashing is the main advantage for the efficient value retrievals of DHTs for a given key.

Other systems on unstructured networks like SkipGraphs [1], Mercury [3] and VoroNet [2] provide multiple-attribute range queries. Compared to DHTs they are less fault tolerant (or require more effort to equalize) and are not as scalable (concerning the search performance for known hashes).

BubbleStorm [9] is an overlay network capable of exhaustive searches in large-scale heterogeneous environments with adjustable probabilistic guarantees. The underlying structure of a well connected multigraph where each node has an even degree is extremely resilient to node crashes, as they do not destroy the search paths, and highly efficient concerning multiple-attribute queries. The only thing missing is the efficiency of a value lookup (like in a DHT) if the hash is known. PathFinder [4] fills this spot by augmenting BubbleStorms random graphs with a deterministic lookup mechanism.

### 4 Distributed slides

After thoroughly comparing different approaches of overlay networks and DHT augmentations regarding query performance, routing and traffic overhead, we considered PathFinder to be a good starting point. The heterogeneous context of DVM regarding hardware, operating systems and network connections is incorporated into the topology of PathFinder. Designed as a connected multigraph PathFinder models this structure as a circular permutation only modified by *JOIN*, *LEAVE* and *CRASH* events of nodes and uses subgraphs of controlled size for data retrieval by solving the rendezvous problem. The proportionality of each nodes degree and its capacity leads to fixed size routing tables and locally controlled workload, which is highly appreciated. If a nodes workload increases, the DVM's content management service can simply request to add more nodes to the neighborhood.

The DHT-like hash lookups of PathFinder are realized by using two pseudorandom number generators (PRNG) that produce a deterministic sequence of numbers when initialized with a specific number. The first PRNG is used to calculate the number of neighbors each virtual node has by producing Poisson distributed numbers. The second one generates numbers that are treated as the node IDs of the neighbors for the current virtual node. This procedure enables each node to determine the neighbors for any node in the network without network communication. To find a path from the seeker  $u$  to node  $v$  containing the required information, the physical node responsible for  $u$  computes the neighbors of  $u$  (what is already accomplished, since they are contained in the routing table) and  $v$ , looking for matching neighbors while increasing the distance to  $u$  and  $v$  in each step by 1 if no match is found.

One key point in distributing slides is the connection between its tiles and the consequences this implies for the computed hash of the tiles. It is necessary that the hash for a tile can be calculated from the WSI it belongs to and its position therein (see Tab. 1).

**Table 1.** Definition for Naming Tiles in DVM

WSI Identifier	Magfication Layer	Z-Stack Layer	Tile Position X	Tile Position Y
32 Bit	8 Bit	8 Bit	16 Bit	16 Bit

Additionally, we necessarily want to store multiple copies of a tile in the DVM to reduce the load on single nodes containing some or many popular tiles. To achieve this, we change the definition of a virtual node. In PathFinder a virtual nodes is located on exactly one physical peer. We augment the definition in a way that a virtual node can be spared over one or more physical nodes.

What needs to be changed in the original concept is that a virtual node has to keep track of the physical nodes it relies on. Therefore each physical node maintains a list of physical nodes sharing a virtual node with it. Incoming requests for an object can be passed from one physical node to his equally responsible neighbor if allowed by performance information from the content management service.

Further, a virtual node must be able to acquire more capacities as new physical nodes join the system or at least move some of its content to adjacent virtual nodes. The former is possible only if a new node joins as a direct neighbor and the latter is already implemented in PathFinder as this is just the creating of a new virtual node resulting in a modification of the hash space. The new virtual node can be moved to other physical nodes if required by means of performance.

## 5 Conclusion

In this paper we have presented DVM, a distributed virtual microscope based on the PathFinder overlay network. We showed that the combined strengths of

efficient hash value lookups as well as efficient long range queries in a single overlay network made PathFinder the superior starting point for DVM. The integrated possibility to locally manage any nodes workload with respect to asynchronous connection bandwidths was a major argument, too. We have also presented DVM-specific extensions to PathFinders structure definition by augmenting the distribution of virtual nodes across physical nodes and choosing a proper basis for the hash function to reflect the correlation between different tiles and their position in a WSI. The random graph based overlay network constructed by PathFinder will provide optimized search results in the application scope of a distributed virtual microscope. Another advantage of our structure enhancement is that many operations can be implemented using parallel algorithms what will most probably lead to at least slightly increased performance. The development process of the distributed architecture will be finished shortly and practical testing will allow further evaluation, adaption and optimization from the gained experiences.

## References

1. James Aspnes and Gauri Shah. Skip graphs. In *SODA*, pages 384–393, 2003.
2. Olivier Beaumont, Anne-Marie Kermarrec, Loris Marchal, and Étienne Rivière. Voronet: A scalable object network based on voronoi tessellations. In *Proceedings of the 21st International Parallel and Distributed Processing Symposium (IPDPS 2007)*. Society Press, 2007.
3. Ashwin R. Bharambe, Mukesh Agrawal, and Srinivasan Seshan. Mercury: Supporting scalable multi-attribute range queries. In *SIGCOMM'04*, pages 353–366, 2004.
4. Dirk Bradler, Lachezar Krumov, Max Mühlhäuser, and Jussi Kangasharju. Pathfinder: Efficient lookups and efficient search in peer-to-peer networks. In M. K. Aguilera et al., editor, *ICDCN*, pages 77–82. Springer-Verlag Berlin Heidelberg, 2011.
5. Richard L. Drake, Jennifer M McBride, Nirusha Lachman, and Wojciech Pawlina. Medical education in the anatomical sciences: The winds of change continue to blow. *Anat Sci Educ*, 2:253–259, 2009.
6. Marcial García Rojo, Gloria Bueno García, Carlos Peces Mateos, Jesús González García, and Manuel Carbajo Vincente. Critical comparison of 31 commercially available digital slide systems in pathology. *International Journal of Surgical Pathology*, 14(4):285–305, October 2006.
7. Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. *Middleware*, pages 329–350, 2001.
8. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM'01*, pages 149–160, 2001.
9. Wesley W. Terpstra, Christof Leng, and Alejandro P. Buchmann. Bubblestorm: Analysis of probabilistic exhaustive search in a heterogeneous peer-to-peer system. Technical Report TUD-CS-2007-2, Technische Universität Darmstadt, 2007.