RUHR-UNIVERSITÄT BOCHUM

Medizinische Fakultät
Medizinisches Proteom-Center
Medical Bioinformatics

Center for Protein Diagnostics

MaCPepDB: Increasing the performance of the mass centric peptide database with old hardware and a distributed database

Dirk Winkelhardt, Martin Eisenacher, Katrin Marcus, Julian Uszkoreit
Ruhr University Bochum, Medical Faculty, Medizinisches Proteom-Center
Ruhr University Bochum, Center for Protein Diagnostics (PRODI), Medical Proteome Analysis

Abstract

Often challenged with small amounts of samples, researchers who wants to run targeted proteomic experiments like single-, multiple- and parallel reaction monitoring (SRM, MRM and PRM), need to know their targets before even measuring their samples once. Furthermore, it is important to know, whether selected peptides are unique, at least for the species of interested. To these lengths we developed MaCPepDB (Mass Centric Peptide Database). MaCPepDB contains the tryptic in silico digest of all known proteins in UniProt, stored in an efficient manner to be quickly searched, currently featuring 185.5 Mio proteins (UniProt Release 2020_03) and around 6 billion peptides.

Since the publication of MaCPepDB we collected a lot more data about its performance, making evident, that high usage leads to completely IO blockage of the main storage of the underlying hardware. Resulting in poor response times and canceled requests. With rapidly increasing number of proteins stored in UniProt, UniProt Release 2021_03 contains 219.7 Mio proteins, the problem gets more inherent. Thus, MaCPepDB needed a way to scale with the increasing amount of data. Therefor the implementation was adjusted to work with Citus Data, an extension for the originally used database engine PostgreSQL, which enables it to scale horizontally on multiple server.

The actual adjustment to MaCPepDB was minor. Instead of automatically partitioning the peptides into multiple smaller tables using PostgreSQL partitioning feature, a table column was introduced, which contains the former partition index and is used by Citus Data to distribute the peptides on multiple servers. The new implementation was tested on six ten-year-old servers, each has 24 to 32 cores and 128 GB RAM and is fitted with 2 consumer SSDs configured in a OpenZFS RAID0. Additionally, a more recent server was setup which is used as database controller to receive and distribute the incoming queries.

Initial testing and first database builds show an increase in write performance by 27 % which suggests the same or even higher increase in read performance once the database indexes are fully build. The performance increase will reduce and stabilize the response times and decrease the canceled requests, while the distributed database engine Citus Data, gives MaCPepDB the flexibility to accommodate even higher amounts of data in the future by adding more or newer servers to the cluster.

Deployment and testing

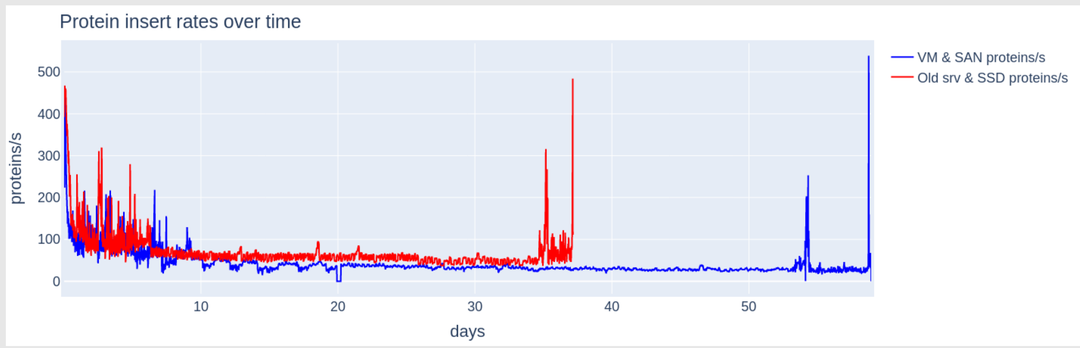
The original MaCPepDB was build on a single server with 112 Cores, 754 GB RAM and 3 consumer SSDs in a RAID 5. This setup quickly reaches the limits of its I/O at roughly 540 MB/s when published.
The second generation of MaCPepDB was built using Citus Data, extending PostgreSQL, to distribute and utilize multiple servers. The new implementation was tested first on our inhouse OpenStack with 6 virtual machines, each got 32 core and 128 GB RAM. The data was stored on SAN-storage with 32 SSDs connected using iSCSI over four parallel 10 GBit network, in theory allowing a theoretical throughput of 5 GB/s.
A second test was performed on a database cluster assembled of five 10 year old servers each had 24 to 32 cores and 128 GB RAM using two consumer SSDs each as main storage for the database and a sixth more recent server which was used as controller to store a few GB of metadata for the database and distribute the incoming queries.
To further decrease the I/O load in exchange for increased CPU time, both cluster initially using OpenZFS, a filesystem with build in compression, in RAID 0 mode.
Each cluster performed an in-silico digest of the complete UniProt to test the write performance while the read performance was tested by querying 26897 MS2 precursors from one of our standard measurements including two post translational modification

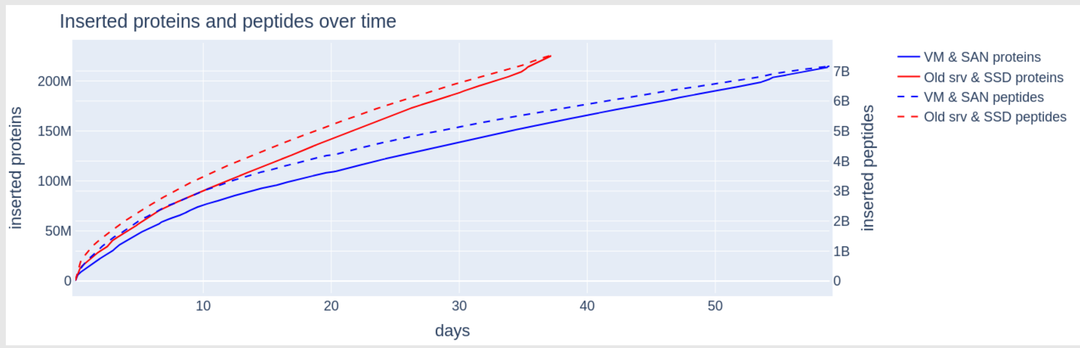
1. Static carboxyamidomethylation of Cysteine
2. Variable oxidation of Methionine

which results in 981566 masses to query.

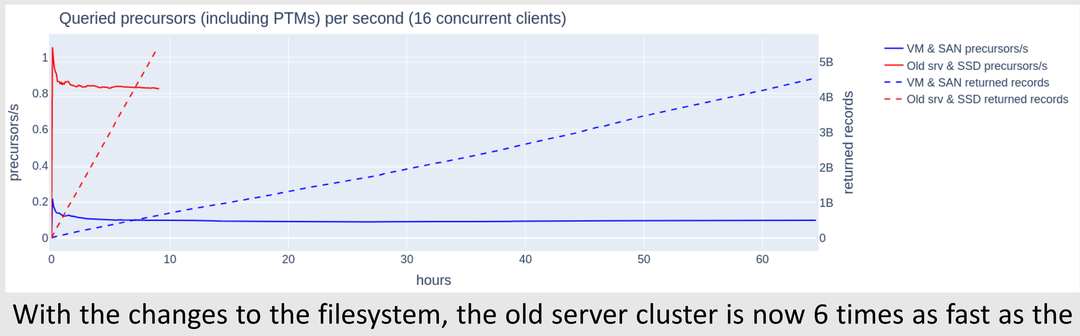
Results

The VM/SAN-solution needed 58.92 days to build the database with an average of 42.19 protein per second.
As already stated in the abstract, the old server cluster are much faster, requires only 37.15 days with an average of 70.27 protein inserts per second, to complete the database build, while using a newer UniProt release containing roughly 11 Mio. more proteins.





With 0.09 queries per second, the VM/SAN-culster was twice as fast as the initial MaCPepDB release.
The older servers did not obtain the expected increase in read performance, in fact they were as twice as slow as our VM/SAN-solution. Assuming the old CPUs had trouble decompressing the data of the ZFS filesystem, indicated by low CPU and disk utilization, we changed it back to the more traditional ext4 filesystem. With this change the read performance was increased by factor of 20.



With the changes to the filesystem, the old server cluster is now 6 times as fast as the VM/SAN-solution in the described read test. Despite much modern hardware the SAN-storage was not capable of supporting the required I/O operations, mainly limited by the network.
As shown, the use of Citus Data as MaCPepDBs database engine and the recycled 10-year-old servers equipped with consumer SSDs are able to overcome the limitations of the original implementation and our SAN-storage, increasing the performance and response times of MaCPepDB significantly.

Subsequent projects

The newly acquired performance enables us to identify roughly 27000 MS2 spectra in 9 hours using the complete UniProt, by identifying one spectra at a time, querying precursor specific peptides from MaCPepDB. A corresponding search engine is currently in development.

Bundesministerium für Bildung und Forschung
FKZ 031L0201