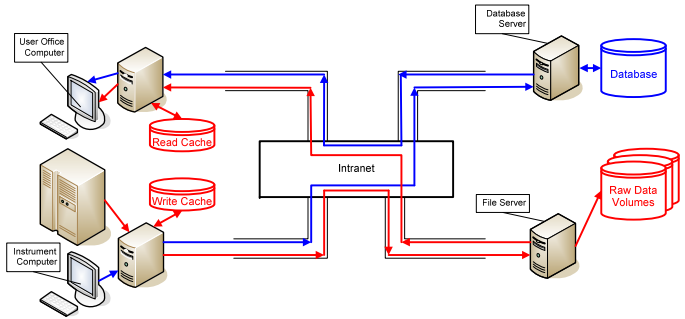
Scalable Data Management on High Throughput MALDI TOF Mass Spectrometer

Poster Number WP 687

George Mills, Matthew Gabeler-Lee Virgin Instruments Corporation, Sudbury, MA

Introduction

Data management on high throughput mass spectrometers rapidly becomes difficult due to the volume of data. With simple file based storage, organizing and locating data is difficult, especially for audit and migration, and the large numbers of files often cause performance problems. With storage wholly in a database, organization is improved, but scalability over time is poor, as the monolithic system requires ever increasing knowledge and storage capacity to keep online, data snapshots for offline work, demos, and project transfers are difficult. We describe a hybrid solution employing a central metadata database with distributed file storage for raw data that seeks to overcome these scalability issues, while also allowing for offline snapshots to ease data archival and migration.



Conclusions and Future Work

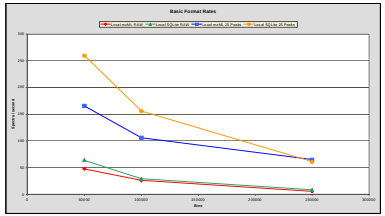
This work demonstrates the feasibility of operating a hybrid storage solution to a high throughput put MALDI TOF Mass Spectrometer even in a modestly performing intranet. It essentially keeps the overall projects organized in a central database without bogging down the database with massive amounts of spectra data while keeping the administration fairly simple. Further work is underway to build a secondary data analysis cache to facilitate high performance 3D Chromatograms for applications such as imaging.

References/Technologies

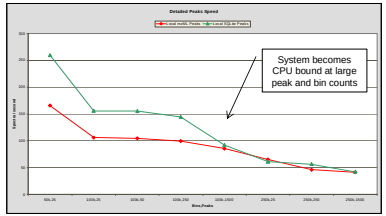
1. SQLite is an Public Domain open source project http://www.sqlite.org/
2. PostgreSQL is under BSD license see http://www.postgresql.org/

Acknowledgements

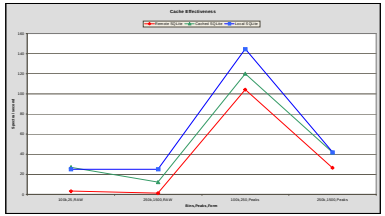
This work was supported in part by the National Institutes of Health under grants RR025705 and GM079832.



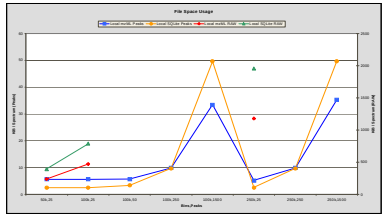
Data rate tradeoffs of using HUPO mzML vs. our SQLite format for both Peaks and Raw data. At large bin counts, the system becomes CPU bound and the file format no longer impacts performance.



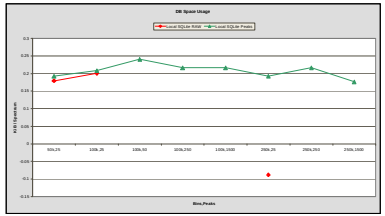
Our SQLite implementation is considerably faster than the HUPO mzML format when the system is I/O bound. At larger bin counts, the system becomes CPU bound doing peak detection, and the choice of format no longer affects performance.



Performance of saving acquisitions to local vs. network shares. A slow fileserver was used for this test to demonstrate effectiveness of the outgoing spectrum cache, which allows usage of a slow or congested file server at near-local speed (not fully optimized).



Data storage requirement tradeoffs of HUPO mzML vs. our SQLite format for both Peaks and Raw data. The SQLite format is somewhat less efficient in some scenarios, but for the more common cases is equivalent.



Database usage is essentially flat with respect to peak and bin count, as it stores a fixed amount of metadata per spectrum. The one negative point shows where the database server automatically ran internal cleanup and freed space.