Quasars spectra classification with the help of GPU computing

P. Wasiewicz[a], K. Hryniewicz[b]

[a]Institute of Electronic Systems, Warsaw University of Technology
[b]N. Copernicus Astronomical Centre, Warsaw, Poland

ABSTRACT

Finding interesting celestial objects among tens of thousands or even millions of recorded raw data is not an easy task to implement. In this paper we speed up this process with high level nvidia cuda C++ template library called Thrust, which makes our database with R interface much more efficient.

Keywords: active galaxies, quasars, data analysis, line identification, database, gpgpu, cuda, nvidia thrust

1. INTRODUCTION

One of the most characteristic signatures of the active galactic nuclei (AGN) are their optical and ultra-violet spectral features. Typical spectra of the AGNs and their most luminous counterparts — quasars — contain prominent broad emission lines on top of a power-law continuum. Schematic view of an AGN is showed in Figure 1. Spectral analysis is one of the most important tool in astrophysics and is robust method of probing ionized matter behaviour, as in AGNs case, in the vicinity of the central black hole. Emission and absorption lines investigation allows us to put constraints on radial velocity along the line of sight and its range. Because of Doppler effect light emitted by a moving body is shifted towards shorter or longer wavelengths (appears to be bluer or redder) depending on the speed and direction of motion.

Spectra of the quasars have been extensivly studied for more than 50 years. Never before we have such amount of data to process and ongoing experiments will bring even more. Many mysteries are buried in the data and many questions have no answer so far. The astronomical analysis base on fitting few emission ingredients by the assumed models. Quite common strategy is to substract iterativly given models fit. It is important to accuretly identify continuum on top of which atomic emission is observed. The way it is done is to chose spectral windows with minimal or no emission and fit power-law continuum only based on the points from given windows.[1] Alternatively we can try fit accretion disc continuum using numerical computed shape for a given set of parameters describing accretion disc around black hole. Although in this approach estimated parameters are uncertain and to some degree there could be degeneracy between parameters.

Some of the elements like iron or hydrogen radiate in many energies effectivly producing emission bands. Those bands are fitted as a set of gaussians for single objects. Recently the most often used technique is to apply iron fits as a template convolved and scaled as needed.[2] In practice it works quite well but not perfectly. Emission lines of a single species are fitted either as a single Gauss or Voight profile or sum of few models. Exemplary components identification is plotted in Figure 3.[3]

There are many issues which become hard to obey in automatic fitting high number of spectra which may vary significantly in parameters values from object to object. In this work we took different approach to analysis having in mind that selection process is preliminary step towards parametrization. And on this stage we can test parameters which have high potential in grouping spectra to its respective classes.

Our method is simpler and is able to give accurate initial parameters in further fitting or filtering procedures. This together with the utilization of modern computational possibilities of GPUs is very promising approach.

Further author information: (Send correspondence to Piotr Wasiewicz)
P. Wasiewicz, e-mail: pwasiewi@elka.pw.edu.pl
K. Hryniewicz, e-mail: krhr@camk.edu.pl

Finding peaks position of the emission lines and comparing patern of peaks with the template spectra is an important process in the data preparation in massive extragalactic sky surveys (i.e. SDSS). This technique is used to estimate value of spectral redshift, however as we presenting in this paper, that task could be used also as a helper in spectra classification.

In this paper in order to compare active galactic nuclei spectra we utilized row-oriented PostgreSQL queries together with high-level parallel processing of nvidia cuda libraries joint by R environment.

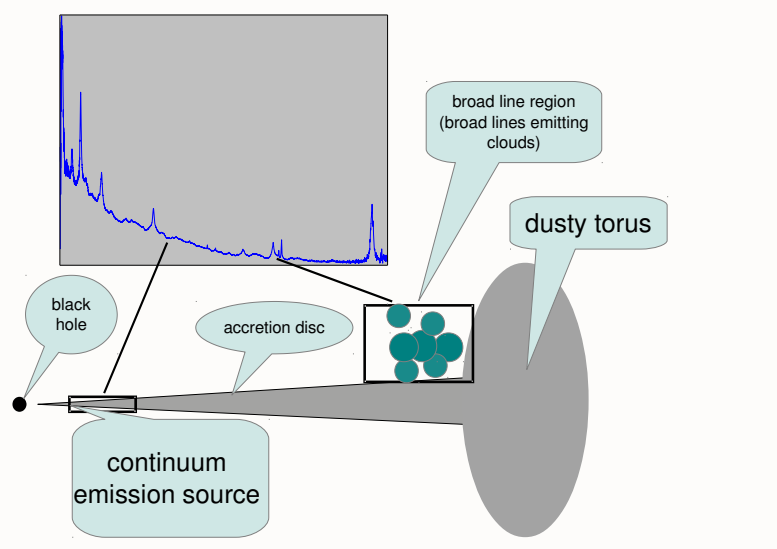

Figure 1. Schematic quasar structure showing cross-section of axialy symetric structures.

2. QUASARS AND THEIR SPECTRUM MODELS

Typical quasar spectrum[4] is shown in Figure 2. Characteristic pattern of the emission peaks is common between AGNs. Because of limited spectral range of instrument we can observe only part of quasar spectrum. Observed pattern is shifted because expansion of the Universe makes distant objects seen as escaping. Spectral shift is described by equation:

$$\lambda_{obs} = \lambda_{em}(z+1),$$

where z is redsift, λ_{obs} is the wavelength seen by telescope, λ_{em} is wavelength emitted by the cosmic source. If we account redshift we can directly compare peaks patterns of a given objects and template.

Interesting untypical objects could have different number of peaks in the given spectral range. Especially useful can be comparing emission and absorption peaks. When we detect smaller number of expected peaks in a given range that could be the case of Weak Line Quasar as in Figure 3 (detection pattern is shown in Figure 6; this object belongs also in to NAL category). If we detect more peaks or many close pairs of up and down peaks it would mean that most likely we found Broad Absorption Line (BAL) object (Fig.5), otherwise Narrow Absorption Line (NAL) object is a possibility (Fig.6).

3. PARALLEL COMPUTATION METHODOLOGY

3.1 Map Reduce Paradigm

Apache Hadoop is a initiated and led by Yahoo! distributed multihost Java framework implementing Google mapreduce paradigm, which enables applications written in different programming and script languages such as Java, Python, Perl, Bash to work with thousands of nodes and petabytes of data and can improve database performance by hot-plug adding the another server node to a cluster.

A Database Management System (DBMS) is a software package with computer programs, which control and manage the database, which an integrated collection of information. Hbase is a distributed database and

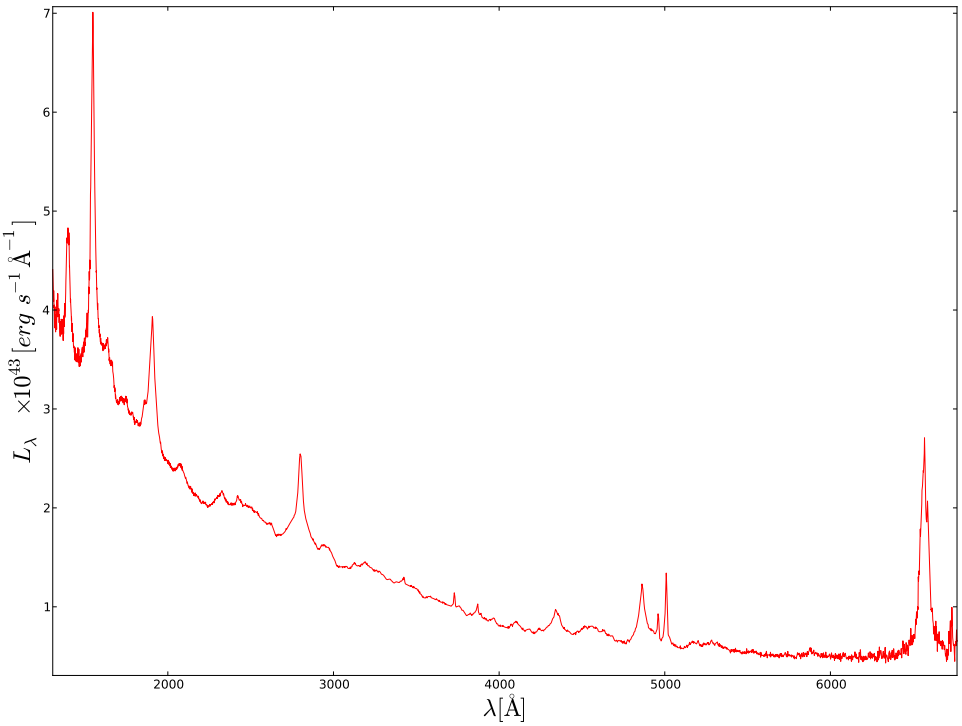

Figure 2. Typical quasar spectrum (composite by Richards[4]).

runs on top of HDFS (Hadoop Distributed Filesystem) providing fault tolerant storing huge data BigTable-like capabilities for Hadoop. Structured Query Language (SQL) is a database computer language designed for managing data in relational database management systems (RDBMS) and based upon relational algebra and calculus is very popular, easy to use and row-oriented. Hive is a data warehouse infrastructure based directly on Hadoop with a SQL-like language HiveQL.

3.2 Column-oriented Paradigm

R is a column-oriented environment[6,7] for statistical computing and graphics, where most calculations are made on arrays, in particular matrices and it has a well-developed, simple, effective programming language which includes e.g. conditionals, loops, user-defined functions, packages from the Comprehensive R Archive Network (CRAN) and enables an effective data handling, storage facility and parallel computing possibilities.

3.3 Stream CUDA Paradigm

Stream processing is a computer programming paradigm, related to SIMD (single instruction, multiple data), that allows using multiple computational units, such as float pointing units (FPUs) on a Graphics Processing Unit (GPU) without explicitly managing allocation, synchronization, or communication among those units. The stream processing paradigm simplifies parallel software and hardware by restricting the parallel computation that can be performed. Given a set of data (a stream), a series of operations (kernel functions) are applied to each element in the stream. Uniform streaming, where one kernel function is applied to all elements in the stream, is typical.

Compute Unified Device Architecture (CUDA) is a parallel computing architecture developed for Nvidia graphics processing units (GPUs). Programmers use C for CUDA (C with Nvidia extensions and certain restric-

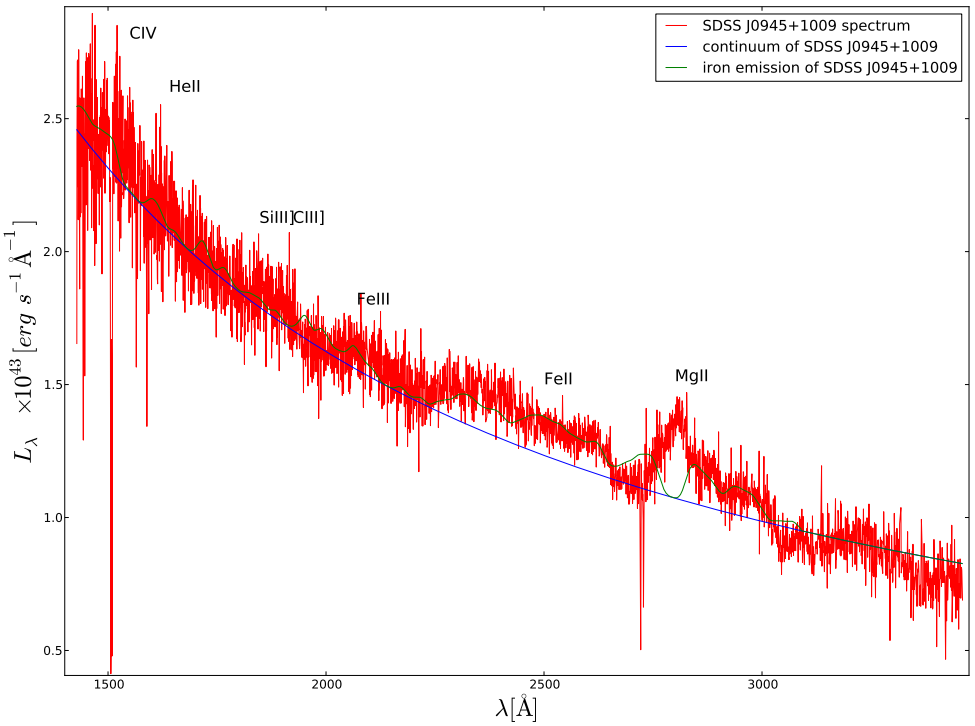

Figure 3. Emission components identification in the exemplary spectrum of the weak line quasar SDSS J0945+1009.

tions), compiled through a PathScale Open64 C compiler, to code algorithms for execution on the GPU. CUDA provides both a low level API and a higher level API.

In this paper we used high-level nvidia thrust template C++ library operating with the help of kernel functions on data vectors -column-major order matrices. Thrust is a CUDA library which is similar to the parallel algorithms in the C++ Standard Template Library (STL). Thrust provides two vector containers, host_vector and device_vector. The first one, as the names suggest is stored in host memory, while the second one exists in GPU device memory. They are able to store any data type and as generic containers can be resized dynamically. Finally, the = operator can be used to copy from one host or device vector to another host or device vector.

```
thrust::device_vector<float> X(10,1);
thrust::device_vector<float> Y(10);
// initialize all ten elements of a device_vector Z to 1
thrust::device_vector<float> Z(10,1);
// moving sum e.g. x[2]=x[0]+x[1]+x[2]
thrust::inclusive_scan(X.begin(), X.end(),Z.begin());
// make index in X from 0 to X.size()-1
thrust::sequence(X.begin(), X.end(),0);
// set all elements of a vector Y to 0
thrust::fill(Y.begin(), Y.end(), 0);
// compute Y = X + Z, add to each X element an appropriate vector Z element
thrust::transform(X.begin(), X.end(), Z.begin(), Y.begin(), thrust::plus<float>());
// make zip of index X and Z vectors
first = thrust::make_zip_iterator(thrust::make_tuple(X.begin(), Z.begin()));
last = thrust::make_zip_iterator(thrust::make_tuple(X.end(), Z.end()));
// a user functor - kernel function from z (get<1>) substract x (get<0>)
struct zipsub : public thrust::unary_function<Numeric2,Numeric> {
```

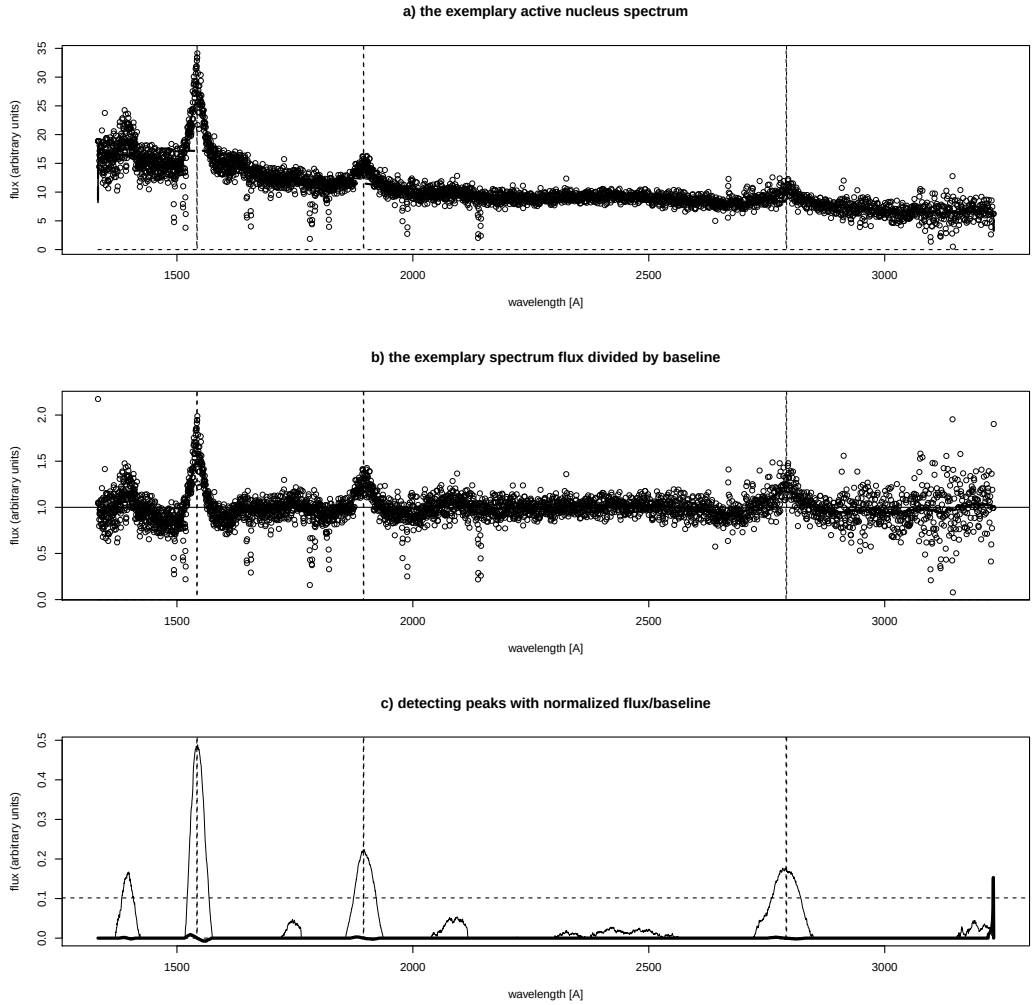

Figure 4. The active nucleus spectrum and its smoothed copies

```
__host__ __device__ Numeric operator()(const Numeric2& a) {
        return thrust::get<1>(a)-thrust::get<0>(a); }};
// X substraction from Z; result: 1 1 1 1 1...
thrust::transform(first, last, Y.begin(), zipsub());
```

Operators begin() and end() return pointers to the first and last vector elements. Functions e.g. fill, copy, and sequence take these pointers as input and output arguments with filling, copying and creating number sequence. More sophisticated functions like inclusive_scan, sort, transform hides parallel algorithms written in CUDA and e.g. with a predefined functor plus every element of vectors X, Y, Z will be processed by transform theoretically in parallel in one moment: $x_i + z_i = y_i$. In practice the parallelization degree depends on number of CUDA processors in GPU. User-defined functors are called kernel functions and should group as many mathematical operations as possible to obtain smaller computation times. Additionally, the zip_iterator allows us to group many independent sequences into a single sequence of tuples, which can be processed by a broad set of algorithms more quickly.

4. CUDA THRUST APPROACH

Our stream computing approach was implemented in column-oriented R environment[7] using PostgreSQL as a dataframe storage before moving to Hbase - database with the unlimited column number.

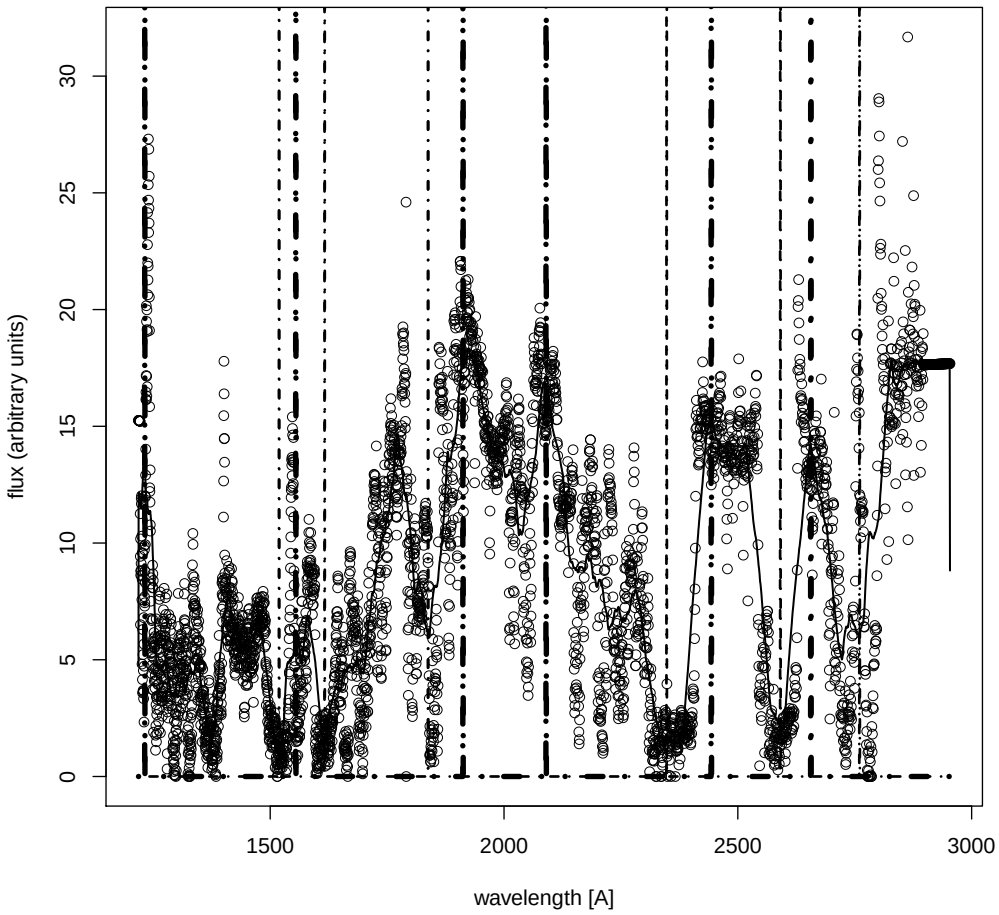

Figure 5. The BAL spectrum of object SDSS J172341.10+555340.5[5] with detected significant peaks (up - a thick dashed line, down - a thin dashed line). However redshift in that kind of object is very uncertain, in normal quasar three or four peaks are expected in plotted range.

For this research the data set of 60 thousand active galactic nuclei spectra was obtained from Sloan Digital Sky Survey (SDSS) online database.[8] Every object has its typical parameters e.g. its name, its kind, z - shift of wavelengths, its place in an observatory plate and additionally light spectrum of about 3500 wavelengths.

In the typical relational database such as PostgreSQL[9] there is a limit for a column number (200-2000 columns - their number depends on their type). Connection between R and PostgreSQL was made with a help of RPostgreSQL package, which enables storing data frames of maximum 1000 columns of a numerical type double. The table object was created with ten basic parameters. The table wave of 60 thousand spectra was partitioned into 60 column-oriented tables. Each of them stores in one column one object spectrum. Thus, one wave part table contains 1000 columns with 1000 objects. Such part tables can be transfered into R data frames and treated massively parallely with the help of nvidia cuda techniques. Our own open source R package gpRepel[10] contains moving average, finding peaks functions.

In Fig. 4a) the original spectrum is depicted with small circles, its smoothed with moving average in a range (-45,45) version – with a continuous line, its smoothed with moving average in a range (-250,250) baseline – with a dashed line. In Fig. 4b) the original spectrum (circles) and its smoothed version divided by baseline are shown and for finding upper peaks outlines above one are used. For original, divided and cut above one spectrum flux

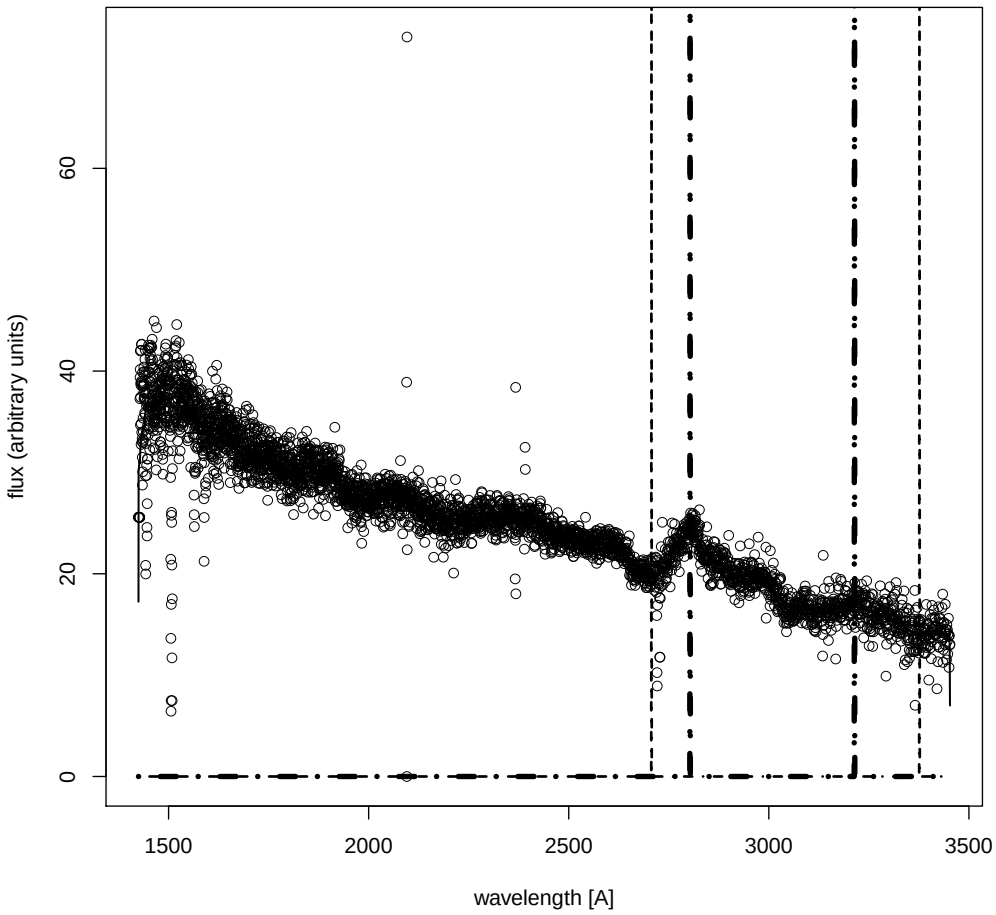

Figure 6. The weak line quasar spectrum of object SDSS J094533.99+100950.1[3] with detected significant peaks (up - a thick dashed line, down - a thin dashed line). In typical quasars three or four peaks would be detected in plotted range. Because of smoothing detected narrow absorption line is not visible but marked by thin dashed line.

a mean is calculated and from smoothed, divided and cut above one spectrum flux this mean is substracted. This final outline is substracted from shifted by one itself and forms the lowest curve in Fig. 4c). Based on this substraction peaks are find where it changes a sign from plus to minus.

All 60000 object spectra were stored in PostgreSQL database and searched for significant peaks with a use of nvidia thrust functions in about 20 minutes with the help of nvidia gtx 1GB card, quad core processor 2.8 GHz, parallelized loops.

5. SUMMARY

With the help of CUDA high level libraries we successfully accelarated AGN spectra comparison software environment, consisted of column-oriented storage and processing in parallelized by gpu card R environment. We decreased by 30 times computation time (due to massively parallel nvidia cuda parallelization) in comparison with the detection peak system based only on R environment and a store in PostgreSQL database.[11]

In future we will accelarate computing within hadoop cluster environment with cuda capabilities.

ACKNOWLEDGMENTS

Work of KH was partly supported by grant NN203 380136 of the Polish State Committee for Scientific Research.

Funding for the Sloan Digital Sky Survey (SDSS) and SDSS-II[8] has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, and the Max Planck Society, and the Higher Education Funding Council for England.

The SDSS is managed by the Astrophysical Research Consortium (ARC) for the Participating Institutions. The Participating Institutions are the American Museum of Natural History, Astrophysical Institute Potsdam, University of Basel, University of Cambridge, Case Western Reserve University, The University of Chicago, Drexel University, Fermilab, the Institute for Advanced Study, the Japan Participation Group, The Johns Hopkins University, the Joint Institute for Nuclear Astrophysics, the Kavli Institute for Particle Astrophysics and Cosmology, the Korean Scientist Group, the Chinese Academy of Sciences (LAMOST), Los Alamos National Laboratory, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, Ohio State University, University of Pittsburgh, University of Portsmouth, Princeton University, the United States Naval Observatory, and the University of Washington. The SDSS Web site is http://www.sdss.org/.

REFERENCES

[1] Forster, K., Green, P. J., Aldcroft, T. L., Vestergaard, M., Foltz, C. B., and Hewett, P. C., "Emission Line Properties of the Large Bright Quasar Survey," *Astrophysical Journal Suplement* **134**, 35–51 (May 2001).

[2] Vestergaard, M. and Wilkes, B. J., "An Empirical Ultraviolet Template for Iron Emission in Quasars as Derived from I Zwicky 1," *Astrophysical Journal Suplement* **134**, 1–33 (May 2001).

[3] Hryniewicz, K., Czerny, B., Nikołajuk, M., and Kuraszkiewicz, J., "SDSS J094533.99+100950.1 - the remarkable weak emission line quasar," *Montly Notices of the Royal Astronomical Society* **404**, 2028–2036 (June 2010).

[4] Richards, G. T., Hall, P. B., Vanden Berk, D. E., Strauss, M. A., Schneider, D. P., Weinstein, M. A., Reichard, T. A., York, D. G., Knapp, G. R., Fan, X., Ivezić, Ž., Brinkmann, J., Budavári, T., Csabai, I., and Nichol, R. C., "Red and Reddened Quasars in the Sloan Digital Sky Survey," *Astronomical Journal* **126**, 1131–1147 (Sept. 2003).

[5] Aoki, K., "Broad Balmer-Line Absorption in SDSS J172341.10+555340.5," *PASJ* **62**, 1333–1339 (Oct. 2010).

[6] Venables, W. N. and Ripley, B. D., [*Modern applied statistics with S*], Springer-Verlag, New York, 4th ed. (2002).

[7] R Development Core Team, *R: A Language and Environment for StatisticalComputing*. R Foundation for Statistical Computing, Vienna, Austria (2010).

[8] Abazajian, K. and et al, "The seventh data release of the sloan digital sky survey," *Astrophysical Journal Supplement Series* **182**, 543–558 (June 2009).

[9] PostgreSQL Global Development Group, "Postgresql documentation." `http://www.postgresql.org/docs/` (2011).

[10] Piotr Wasiewicz, "R package gpRepel." `http://r-forge.r-project.org/projects/gprepel/` (2011).

[11] Wasiewicz, P. and Hryniewicz, K., "Astronomical spectral database of active galactic nuclei," in [*XXVIII-th IEEE-SPIE Joint Symposium on Photonics, Web Engineering, Electronics for Astronomy and High Energy Physics Experiments,Wilga*], (2011).