J Grid Computing
https://doi.org/10.1007/s10723-017-9416-0

Computing Extremely Large Values of the Riemann Zeta Function

Norbert Tihanyi · Attila Kovács · József Kovács

Received: 16 December 2016 / Accepted: 19 October 2017

Abstract The paper summarizes the computation results pursuing peak values of the Riemann zeta function. The computing method is based on the RS-PEAK algorithm by which we are able to solve simultaneous Diophantine approximation problems efficiently. The computation environment was served by the SZTAKI Desktop Grid operated by the Laboratory of Parallel and Distributed Systems at the Hungarian Academy of Sciences and the ATLAS supercomputing cluster of the Eötvös Loránd University, Budapest. We present the largest Riemann zeta value known till the end of 2016.

Keywords Riemann zeta function · Distributed computing · Large $Z(t)$ values

N. Tihanyi (✉) · A. Kovács
Eötvös Loránd University, Budapest, Hungary
e-mail: norbert.tihanyi@inf.elte.hu

A. Kovács
e-mail: attila.kovacs@inf.elte.hu

J. Kovács
Institute for Computer Science and Control, Budapest, Hungary
e-mail: jozsef.kovacs@sztaki.mta.hu

1 Introduction

Let $s = \sigma + it$ where $\sigma, t \in \mathbb{R}$. The *Riemann zeta function* is defined by

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s} \qquad (\sigma > 1)$$

which is an extremely important function of mathematics and physics. In 1859 Bernhard Riemann conjectured that all non-trivial zeros of this function have real part $\sigma = 1/2$. This is the famous Riemann-hypothesis, one of the most important unsolved problem in the theory of prime numbers.

On the real line with $x > 1$ the function can be defined by the integral

$$\zeta(x) = \frac{1}{\Gamma(x)} \int_0^{\infty} \frac{u^{x-1}}{e^u - 1} du.$$

By analytic continuation the function can be extended to the whole complex plane, except for a simple pole at $s = 1$, satisfying the functional equation

$$\zeta(1-s) = 2(2\pi)^{-s} \cos\left(\frac{1}{2}s\pi\right) \Gamma(s)\zeta(s),$$

which can be written in a symmetric form

$$\Gamma\left(\frac{s}{2}\right)\pi^{-s/2}\zeta(s) = \Gamma\left(\frac{1-s}{2}\right)\pi^{-(1-s)/2}\zeta(1-s)$$

Published online: 06 November 2017

proved by Riemann. By introducing the meromorphic function $\xi(s)$ defined by

$$\xi(s) = \pi^{-s/2}\Gamma(s/2)\zeta(s)$$

we have that $\xi(s) = \xi(1-s)$. The function ξ has the same zeros as the zeta function in the critical strip, and is real on the critical line because of the functional equation, so one can prove the existence of zeros exactly on the real line between two points by checking numerically that the function has opposite signs at these points. Usually one writes

$$\zeta\left(\frac{1}{2}+it\right) = Z(t)e^{-i\theta(t)}$$

where $Z(t)$ is the *Hardy's function* or *Riemann-Siegel Z-function* and

$$\theta(t) = \arg\left(\Gamma\left(\frac{2it+1}{4}\right)\right) - \frac{\ln\pi}{2}t \approx \frac{t}{2}\ln\frac{t}{2\pi} - \frac{t}{2} - \frac{\pi}{8} + \frac{1}{48t} + \frac{7}{5760t^3} + \cdots$$

for real values of t. The previous approximation is not convergent, but the first few terms give a good approximation for $t \gg 1$. The function $Z(t)$ can be calculated in time complexity of $\mathcal{O}(t^{1/2})$ by

$$Z(t) = 2\sum_{n=1}^{\lfloor\sqrt{t/2\pi}\rfloor}\frac{1}{\sqrt{n}}\cos(\theta(t) - t\cdot\ln n) + \mathcal{O}(t^{-1/4}), \tag{1}$$

see e.g. [1, 2]. Ghaith A. Hiary in 2011 presented [3, 19] how to compute $Z(t)$ within $\mathcal{O}(t^{2/5})$, $\mathcal{O}(t^{1/3})$ and $\mathcal{O}(t^{4/13})$ time complexities, respectively. Since $\zeta(1/2+it)$ is unbounded, $Z(t)$ can take arbitrarily large values as t goes to infinity.

The Riemann zeta function describes the behavior of the distribution of the primes numbers. Many authors computed the zeroes of $\zeta(t)$ proving that the Riemann-hypothesis is true in some bounded interval. In 1986 J. van de Lune, te Riele and Winter checked that the first $1.5\cdot 10^9$ non-trivial zeroes all lie on the critical line [18]. S. Wedeniwski used the distributed ZetaGrid project to verify that the first 100 billion zeros all lie on the critical line. In 2004 the Odlyzko-Schönhage algorithm [4] was used by Gourdon to verify the Riemann hypothesis for the first 10^{13} zeros of the zeta function [5].

Locating peak values of the Riemann zeta function on the critical line is another promising method for getting a better understanding of the distribution of prime numbers.

The Riemann hypothesis is equivalent to the statement that all local maxima of $Z(t)$ are positive and all local minima are negative, and it has been suggested that if a counterexample exists then it should be in the neighborhood of unusually large peaks of $\zeta(1/2+it)$.

For $m \geq 0$ the m^{th} Gram point g_m can be defined by the unique solution of

$$\theta(g_m) = m\pi.$$

Gram's law is based on the observation that $Z(t)$ usually changes sign in the Gram intervals $G_j = [g_j, g_{j+1})$ for $j \geq 0$. A Gram point g_j is said to be "good" if $(-1)^j Z(g_j) > 0$ and "bad" otherwise. A Gram block with length k is an interval $M_j = [g_j, g_{j+k})$ such that g_j and g_{j+k} are good Gram points and $g_{j+1}, \ldots g_{j+k-1}$ are bad Gram points for $k \geq 1$. The interval M_j satisfies the Rosser's rule if $Z(t)$ has at least k zeroes in M_j. Rosser's rule is violated infinitely often, but only for a small fraction of the Gram blocks. In 1979 Richard P. Brent computed the first 75 000 000 zeros of $\zeta(s)$ and observed an unusually large $Z(t)$ (> 79.6) near the 70354406^{th} Gram point [1]. In all the cases, where an exception to Rosser's rule was observed, there was a large local maximum of $Z(t)$ nearby. That is the reason that calculating peak values of $Z(t)$ can reveal new interesting behavior of the distribution of prime numbers. The aim of this paper is to present the computation result of thousands of large values of $|Z(t)| > 10\,000$. These values can be used to analyze the behavior of the Riemann zeta function deeply connected to the distribution of prime numbers. We also present the largest known $Z(t)(\approx 16\,854)$ found by the RS-PEAK algorithm [6].

2 Largest Known Values of $Z(t)$

In 1983 J. van de Lune found that for $t = 7251778 80629981.914$ we have $Z(t) \approx -453.9$. The first values where $|Z(t)| > 1000$ were found by Odlyzko [7] in 1989, the largest one that time was the value $Z(t) \approx 1581.7$ for $t = 503286876928828 9111.35$. Calculating $Z(t)$ is a very expensive task with the original Riemann-Siegel Formula, even with using the Odlyzko-Schönhage algorithm [4]. Applying the

$O(t^{1/3})$ algorithm of Hiary, J. Bober and G. Hiary were able to find many large values of $Z(t)$ [8]. They found that $Z(t)$ is approximately 16244.8652 for $t = 39246764589894309155251169284104.0506$. Table 1 shows the largest values of $|Z(t)| > 10.000$ found by Bober and Hiary.

Finding peak values of $Z(t)$ is computationally expensive and challenging even with modern supercomputers. No explicit formula is known to find such t where $\zeta(1/2 + it)$ is large, however, methods are known from A.M. Odlyzko [7] and T. Kotnik [9] for locating large values of $Z(t)$ more efficiently than choosing t randomly.

The RS-PEAK algorithm [6] is based on simultaneous Diophantine approximations and can be used very effectively to find good candidates where lagre $Z(t)$ is likely. Applying the algorithm we have found thousands of $|Z(t)| > 10\,000$ values and the largest known value $Z(310678833629083965667540576593682.058) \approx 16858.119$ has been calculated recently.

3 The RS-PEAK Algorithm

Consider a set of irrationals $\Upsilon = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, let $\varepsilon > 0$ and let us define the set

$$\Lambda(\Upsilon, \varepsilon) = \{k \in \mathbb{N} : \|k\alpha_i\| < \varepsilon \text{ for all } \alpha_i \in \Upsilon\} \tag{2}$$

where $\|\cdot\|$ denotes the nearest integer distance function. Finding *many* appropriate $k \in \Lambda(\Upsilon, \varepsilon)$ is a *simultaneous Diophantine approximation problem.* One of the most efficient algorithms for solving such simultaneous Diophantine approximation problems is the Lenstra-Lenstra-Lovász (L^3) basis reduction algorithm [10]. One can use the L^3 algorithm in order to find *one* appropriate k satisfying (2). In 2013 the first two authors of this paper presented a method for solving n-dimensional simultaneous Diophantine approximation problems efficiently [11]. The method is based on the following theorem:

Theorem 1 *Let $\Upsilon = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ be a set of irrationals and $\varepsilon > 0$. Then there is a set Γ_n with 2^n elements with the following property: if $k \in \Lambda(\Upsilon, \varepsilon)$ then $(k + \gamma) \in \Lambda(\Upsilon, \varepsilon)$ for some $\gamma \in \Gamma_n$.*

The generation of Γ_n can be done efficiently and can be used to produce *many* $k \in \mathbb{N}$ much faster than with L^3 in small dimensions ($n \le 20$), see [11, 12].

Based on this result we have introduced the *Multithreaded Advanced Fast Rational Approximation* algorithm (MAFRA) [12] for solving simultaneous Diophantine approximation problems. Then, we introduced the algorithm RS-PEAK [6] which is based on the following idea: one has to find an integer k such that the real numbers $k\frac{\ln p_i}{\ln 2}$ are all close to some integers for as many primes p_i as possible. In this case at the points $t = \frac{2k\pi}{\ln 2}$ the approximation

$$\cos\big(\theta(t) - t\ln n\big) \approx \cos\big(\theta(t)\big)$$

holds. For this particular Diophantine approximation problem L^3 is substituted by MAFRA achieving

Table 1 Largest $Z(t)$ values found by Bober and Hiary [8]

t	$Z(t)$
39246764589894309155251169284104.0506	16244.8652
70391066310491324308791969554453.2490	−14055.8928
552166410009931288886808632346.5052	−13558.8331
35575860004214706249227248805977.2412	13338.6875
6632378187823588974002457910706.5963	12021.0940
698156288971519916135942940460.3337	11196.7919
289286076719325307718380549050.2563	10916.1145
50054757231073962115880454671617.4008	−10622.1763
803625728592344363123814218778.1993	10282.6496
690422639823936254540302269442.4854	10268.7134
1907915287180786223131860607197.5463	10251.5994
9832284408046499500622869540131.7445	−10138.5908

significant improvement in locating appropriate integers k. For further filtering we analyzed the function

$$F(t, A, B) = \sum_{n=A}^{B} \frac{1}{\sqrt{n}} \cos(\theta(t) - t \cdot \ln n)) \qquad (3)$$

and investigated its behavior for various A and B, where $B \leq \lfloor \ln t/2\pi \rfloor$. We concluded that in many cases (3) can be used effectively to indicate where large values of $|Z(t)|$ are likely. Combining these techniques led us to introduce the algorithm RS-PEAK for finding potential candidates. Applying the algorithm on a single Desktop PC one can find thousands of candidates where large $|Z(t)|$ is likely.

4 The Growth of $Z(t)$

The Lindelöf hypothesis is a conjecture about the rate of growth of the Riemann zeta function on the critical line that is implied by the Riemann hypothesis. It states that for any $\epsilon > 0$

$$\zeta\left(\frac{1}{2} + it\right) = \mathcal{O}(t^{\epsilon}).$$

Bourgain showed [13] that

$$\left|\zeta\left(\frac{1}{2} + it\right)\right| \ll t^{13/84+\epsilon}.$$

We also investigated the function $\phi(t) = \frac{\log|Z(t)|}{\log(t)}$ finding many large values.

5 Computational Environment for the $Z(t)$ Calculation

This section is going to introduce the computational environment serving the calculation of the $Z(t)$ values since the end of 2013, searching for candidates where large values of $|Z(t)|$ is expected.

5.1 Overview

The key components of the overall infrastructure (see Fig. 1) are the followings:

1. Zeta-search application binaries implementing the RS-Peak algorithm
2. BOINC server coordinating the distribution of millions of workunits to BOINC clients

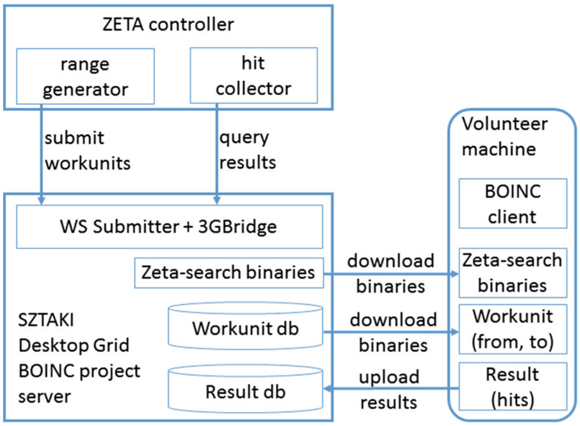

Fig. 1 Overview of the SZTAKI Desktop Grid infrastructure for Zeta calculation [14]

3. BOINC client for executing Zeta-search workunits on volunteers' machine
4. 3GBridge for generating workunits
5. Zeta controller for generating parameters

The Zeta-search application has been developed to implement the RS-PEAK algorithm. One run performs the search of candidates between the given range(s). Parameters for this application include the range (minimum and maximum number of t values among which the search to be performed) or list of ranges and a threshold value which represents the minimal value above which the t value is considered as a candidate. The application has been integrated to the BOINC [15] environment to make it executable under the control of the BOINC client, running on the volunteers' machine.

BOINC together with the extensions from the SZDG package [14] performs the distribution of workunits (jobs) to the volunteers' machine. In BOINC a volunteer person can join a particular project by registering an account on the project homepage. BOINC client is a piece of software that is able to communicate to the BOINC server to get workunits, to execute the workunit and to upload the result. After downloading, installing and running the BOINC client on the volunteer's machine, one can attach his/her BOINC client to a BOINC project with the given credentials (email/password). After attaching to a project it starts fetching workunits containing a pack of input files and parameters for a certain application.

The BOINC client downloads the application binary referred by the workunit. Then the BOINC client starts executing the application in the background with the input files and parameters described in the workunit. Once the workunit finishes, the results are uploaded and reported to the BOINC project and new workunits are fetched. BOINC ensures the distribution of millions of workunits by the server toward the attached volunteers.

The SZTAKI Desktop Grid [14] software package is an extension for the BOINC software on the field of job submission, execution and API. Its main contribution is the 3GBridge component – used in this solution – performing the creation of workunits inside the BOINC database. Once a job (application with arguments, input and output files, etc.) has been submitted through the WS Submitter into the 3GBridge, a new BOINC workunit is generated. The status can be queried and the results can be retrieved. WS Submitter is the web-service endpoint for the 3GBridge (see Fig. 1).

SZTAKI Desktop Grid is not only a software package, but also a BOINC based volunteer desktop grid project operated by the Laboratory of Parallel and Distributed Systems in the Institute for Computer Science and Control of the Hungarian Academy of Sciences, Budapest, Hungary. The SZTAKI Desktop Grid BOINC project [16] is hosted on a virtual machine on the cloud computing infrastructure of the laboratory. The virtual machine has Debian GNU/Linux 6.0.7 (Squeeze) OS with 4 core AMD64 CPU (KVM). Due to the low resource requirements of BOINC, the server runs with 2GB RAM. For storing the wokunits the server has 500GB. The bandwidth of its network connection is 1Gbit/s.

The SZTAKI Desktop Grid Project was established in 2005 and has currently about 40 thousands volunteers and 112 thousands hosts registered. However, the number of actively working hosts is much lower since hosts and users join and leave frequently in a BOINC environment. The number of active hosts is approximately 2200 owned by about 1700 active users. A host is considered active if at least one finished workunit is reported by the host within 48 h.

In the infrastructure the Zeta controller (see Fig. 1) is responsible for generating the load (i.e. millions of jobs) for the BOINC project. Range generator performs the calculation of the ranges to be processed and submits the corresponding jobs to SZDG through the WS Submitter. The Hit collector downloads the outcome of Zeta-search executions and extracts the valuable results, i.e. the value of t and $|Z(t)|$.

Overall, the flow of jobs starts with the Zeta-controller component by a range generator, continues with 3GBridge and BOINC and ends up on the volunteer machines which downloads the Zeta-search executables and parameters. Once the workunit has been processed, the results travel back through the same route to the Zeta controller server (see Fig. 1) machine.

5.2 Distribution of Ranges Among Multiple BOINC Projects

The Zeta search controller server is able to feed multiple BOINC projects with jobs (ranges) as it is depicted in Fig. 2. When the capacity of a BOINC project is not enough for processing the workunits fast enough it is needed to attach further resources. The Parallel and Distributed Systems Laboratory operated multiple BOINC projects for some years. In order to speed up the computation for the Zeta-search application, the Zeta controller machine was configured to submit jobs to multiple BOINC projects through the WS Submitter and 3GBridge components. ZETA controller was responsible to keep track of the different ranges submitted to the different projects and to detect if the computation for a certain range has not arrived in time from one of the BOINC projects.

Until the end of 2015, the laboratory has assigned the EDGeS@home BOINC project resources (volunteers) for executing Zeta-search calculations. During that period both BOINC projects issued workunits with the same applications but with different parameters (ranges). The accumulated performance increased the processing speed by 15-20 percentage since the

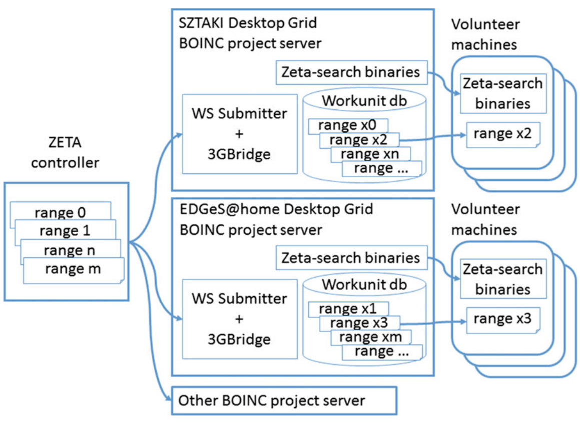

Fig. 2 Distribution of ranges among BOINC infrastructures

number of volunteers on the SZTAKI Desktop Grid was 5-6 times more than on the EDGeS@home BOINC project.

Now, this architecture gives a justification of creating the Zeta controller job (range) generator as a separate component instead of developing it as an integrated component inside the BOINC architecture. With this solution it was possible to assign multiple BOINC projects for the same application. Moreover, 3GBridge is able to create jobs not only for BOINC, but for other systems as well (due to its pluggable architecture) therefore other type of resources could have been attached. Due to the huge computational capacity requirement of the Zeta calculation the cheapest, i.e. BOINC resources were used.

5.3 Architecture of Zeta Controller

One of the key components of the execution environment is the Zeta controller. The controller has several different functionalities:

1. Generates the ranges for which jobs can be created and processed as workunits on the volunteer machines
2. Makes sure the numbers to be scanned through are covered by the ranges
3. Detects if result has not arrived back for a certain range
4. Resubmits failed or lost workunits
5. Distributes the generated ranges among the BOINC projects
6. Keeps the number of jobs in the BOINC projects on a predefined level
7. Collects results and stores the value of t and $|Z(t)|$ in the database
8. For a predefined interval (currently one week) it summarizes hits, creates and stores a report on an external storage and notifies scientists
9. Removes unnecessary entries in database

Range generator (see in Fig. 3) is invoked periodically, i.e. once per hour. First it queries the number of unsent workunits on the target BOINC project to detect the available freely downloadable workunits for the clients. If this number is close to zero the frequency or the number of maximum unsent workunit must be increased. Once it has this number then it generates the number of ranges needed to fill up the number of unsent workunits to a predefined level.

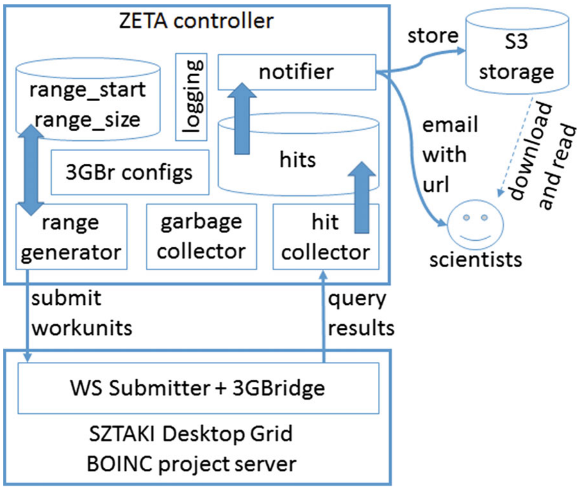

Fig. 3 Architecture of the Zeta controller server

This level can be specified per BOINC project. With this mechanism the workunit generation can follow the changing capacity of the volunteer computational systems.

The same range generation mechanism is invoked for each configured BOINC projects (see 3GBr configs in Fig. 3) with their own configuration parameters. The range generator reads the actual starting number from the database and updates it when new ranges have been generated. The size of the range is also a configuration parameter which is proportional with the amount of work a workunit contains (see range_start, range_size in Fig. 3).

The Hit collector (see in Fig. 3) is also invoked periodically, i.e. once per hour. First, it queries all the finished jobs stored in 3GBridge of the BOINC project and downloads their outputs. Then, extracts the results and stores them in the database (see hits in Fig. 3). Moreover, this component also queries and stores the amount of computational capacity spent for processing the given range. It is needed for summarizing the overall computational capacity used for finding candidates.

The Notifier (see in Fig. 3) is also executed periodically, i.e. once per week in order to summarize the large Zeta values found during the last week. This report is generated and uploaded to a storage accessible for the scientists. Then, the scientists are notified via email containing the url of the report.

The operation of all the components are controlled by using a logging subsystem (see in Fig. 3) to make root cause analysis easier. Garbage collector is originally designed to cleanup hits and already reported values, but it is not used currently, since hits does not consume large amount of storage space.

The careful and modular design of the zeta controller attached to the SZTAKI BOINC projects resulted in continuous operation of the Zeta controller for three years now without interruption caused by malfunction during the generation of almost 34 millions of jobs (ranges).

6 Computation Results

The Zeta-search application on SZDG (containing the RS-PEAK algorithm) is trying to locate as many large values of the zeta function as possible on the critical line for statistical analysis. It has been deployed on SZDG on November, 2013. The largest known $Z(t)$ value has been found in the end of November 2015. During this 2 year period, altogether 33 905 823 pieces of workunit have been processed by the attached BOINC clients worldwide. Altogether, the capacity used for these workunits is 95 302 445 457 996 834 FLOPS, i.e. 95 302 TFLOPS. During the overall 3 years period, more than 10 million Z(t) candidates were found by the SZDG.

6.1 ATLAS Supercomputing Cluster

The candidates then were verified by ATLAS Supercomputing cluster operating in the Eötvös Loránd University calculating the exact $Z(t)$ values by applying Hiary's $\mathcal{O}(t^{1/3})$ algorithm. The architecture consists of one dedicated Headnode and 44 Computing nodes. The most important characteristics of ATLAS are the following:

Headnode:

1. 2x Intel Xeon E5520 Nehalem Quad Core 2.26 GHz Processor with 8 MB cache (HyperThreading OFF)
2. 72 Gbyte RAM
3. 10 Gbit eth interface to the 44 computing nodes

44 Computing Nodes:

1. 2x Intel Xeon E5520 Nehalem Quad Core 2.26 GHz Processor with 8 MB cache (HyperThreading ON)
2. 12 Gbyte RAM

Each Nehalem Quad core CPU has 4 physical cores with SSE extension. Each node has 2×36.256 GFLOP/sec peak performance [17]. There are 44 computing nodes which contain 88 physical CPU. The total number of physical cores are 352 (4×88). The peak performance of the ATLAS Computing Cluster is $72.512 \times 44 = 3190.528$ GFLOP/sec.

Approximately 30 000–50 000 candidates were found weekly by the SZTAKI Desktop Grid applying the RS-PEAK algorithm. As mentioned earlier, until September 2016 more than 10 million candidates were calculated. The first candidates were sent by the SZDG on 16th of December, 2013. The largest $|Z(t)|$ value among the first 20 thousand candidates was $Z(t) \approx 5732.123$ for $t = 451632495000754348192139717644.136$.

After ten weeks the SZTAKI Desktop Grid found thousands of large values where $|Z(t)| > 10\,000$. After one year only two large values found by the SZTAKI Desktop Grid where $|Z(t)| > 15\,000$ (see Table 2).

For $t = 73027109216315547125974615.940$ we have $Z(t) \approx 5297.23$ with $\phi(t) \approx 0.1439$. The largest $\phi(t)$ occurred at $t = 6436526919750171929565.992$, where we have $Z(t) \approx 2942.71$, $\phi(t) \approx 0.15905$.

On 30th of November, 2015 SZDG indicated an unusual large value in the region $t \approx 3.106 \cdot 10^{32}$. For $t = 310678833629083965667540576593682.058$ we have $Z(t) \approx 16858.119$. To the best of our knowledge at the time of writing of this paper, this is the largest known $Z(t)$ value.

Summary of computation results:

1. More than 4 million values where $Z(t) > 1\,000$.
2. More than 5 000 values where $Z(t) > 10\,000$.
3. Largest known value of $Z(t) \approx 16\,858$.
4. Largest known value of $\phi(t) \approx 0.15905$ is at 6436526919750171929565.992.

7 Further Researches

Applying the RS-PEAK algorithm one can find many candidates where large $Z(t)$ is likely. In the last few

Table 2 Large $Z(t)$ values

t	$Z(t)$
28456701449396688374847224655196.737	-15481.206
167495487143636918323945908249296.675	15323.691

years we have collected more than 10 millions of candidates where large $Z(t)$ is expected. Peak values of the zeta function can be used to analyze the behavior of the Riemann zeta function and have a better understanding of the distribution of prime numbers. Our next goal is to identify new patterns in the distribution of large values of the Riemann zeta function on the critical line. The whole project data is available at www.riemann-siegel.com.

Acknowledgments The authors gratefully acknowledge the constructive comments of Ghaith A. Hiary. We would like to thank the valuable suggestions for the anonymous reviewers. The authors would like to thank the opportunity for accessing to the ATLAS Super Cluster operating at Eötvös Loránd University and for using the capacity of SZTAKI Desktop Grid project operated by the Laboratory of Parallel and Distributed Systems in the Institute for Computer Science and Control of the Hungarian Academy of Sciences.

References

1. Brent, R.P.: On the zeros of the Riemann Zeta function in the critical strip. Math. Comp. **33**(148), 1361–1372 (1979)
2. Sherman Lehman, R.: Separation of zeros of the Riemann Zeta-Function. Math. Comp. **20**, 523–541 (1966)
3. Hiary, G.A.: Fast methods to compute the Riemann Zeta function. Ann. Math. **174-2**, 891–946 (2011)
4. Odlyzko, A.M., Schönhage, A.: Fast algorithms for multiple evaluations of the Riemann Zeta function. Trans. Am. Math. Soc. **309**, 797–809 (1988)
5. Gourdon, X.: The 10^{13}-rst zeros of the Riemann Zeta function, and zeros computation at very large height. http://numbers.computation.free.fr/Constants/Miscellaneous/zetazeros1e13-1e24.pdf. Accessed 5 December 2016 (2004)
6. Tihanyi, N.: Fast method for locating peak values of the Riemann Zeta function on the critical line. In: Sixteenth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) (2014). IEEE Explorer. https://doi.org/10.1109/SYNASC.2014.20
7. Odlyzko, A.M.: The 10^{20}-th zero of the riemann zeta function and 175 million of its neighbors. http://www.dtc.umn.edu/∼odlyzko/unpublished/. Accessed 5 December 2016 (1992)
8. Bober, J.W., Hiary, G.A.: New computations of the Riemann Zeta function on the critical line. Exp. Math., **27**, 1–13 (2016)
9. Kotnik, T.adej.: Computational estimation of the order of $\zeta(1/2+it)$. Math. Comp. **73**(246), 949–956 (2004)
10. Lenstra, A.K., Lenstra, H. Jr, Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. 261(4), 515–534 (1982)
11. Kovács, A., Tihanyi, N.: Efficient computing of n-dimensional simultaneous Diophantie approximation problems. Acta Univ. Sapientiae, Informatica **5-1**, 16–34 (2013)
12. Tihanyi, N., Kovács, A., Szücs, Á.: Distributed computing of simultaneous Diophantine approximation problems. Stud. Univ. Babes-Bolyai Math. **59**(4), 557–566 (2014)
13. Bourgain, J.: Decoupling, exponential sums and the Riemann Zeta function. J. Am. Math. Soc. **30**(1), 205–224 (2017)
14. Kacsuk, P., Kovács, J., Farkas, Z., et al.: SZTAKI desktop grid (SZDG): a flexible and scalable desktop grid system. Journal of Grid Computing, Special Issue: Volunteer Computing and Desktop Grids **7**(4), 439–461 (2009)
15. Anderson, D.P.: BOINC: a system for public-resource computing and storage. In: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing (GRID '04). IEEE Computer Society, Washington, DC, USA
16. The SZTAKI desktop grid BOINC project. http://szdg.lpds.sztaki.hu/szdg. Accessed 5 December 2016
17. Peak performance of intel CPU's. http://download.intel.com/support/processors/xeon/sb/xeon_5500.pdf. Accessed 5 December 2016
18. van de Lune, J., te Riele, H.J.J., Winter, D.T.: On the zeros of the Riemann Zeta function in the critical strip. IV. Math. Comp. **46**, 667–681 (1986)
19. Hiary, G.A.: A nearly-optimal method to compute the truncated theta function, its derivatives, and integrals. Annals of Math. **174**, 859–889 (2011)