PrimeGrid: a Volunteer Computing Platform for Number Theory

Iain Bethune
Edinburgh Parallel Computing Centre,
The University of Edinburgh
Edinburgh, United Kingdom
ibethune@epcc.ed.ac.uk

***Abstract*—Since 2005, PrimeGrid has grown from a small project factorising RSA numbers by brute force to one of the largest volunteer computing projects in the world. The project has discovered over 60 new million-digit primes, as well as record sized twin and Sophie Germain primes. This paper will present a history of the project, the algorithms and software used by PrimeGrid and how the BOINC distributed computing architecture is used to harness tens of thousands of computers. We also highlight some recent results from several current prime search sub-projects.**

Keywords-volunteer computing; number theory; primality testing; computational mathematics; prime numbers

I. BACKGROUND

The properties and distribution of prime numbers have fascinated mathematicians since Renaissance times, in particular the search for very large primes. Indeed even today Riemann's famous Hypothesis [1] remains unproven and an active area of research. In the 17th century the search for large primes focused on Mersenne numbers having the form 2^p-1, where p is itself a prime. In 1558 Pietro Cataldi proved the primality of 2^{17}-1 and 2^{19}-1 [2], both 6 digit numbers, but further progress was slow, as the only known method to prove a number as prime or composite was trial division, which requires $O(\sqrt{N})$ divisions and rapidly becomes impractical. Indeed, Mersenne himself claimed (incorrectly) that 2^{257}-1 was prime, but was only proved wrong in 1947!

In 1867 the largest known prime contained only 13 digits, until Lucas' startling discovery of a 39 digit Mersenne prime 2^{127}-1, using a number theoretic method based on Lucas Sequences, much more efficient than trial division. This method was such a success that it is still in use today in an only slightly modified form.

However, the most major innovation in the search for large primes came with the advent of electronic computers in the 1950s. In 1951 and 1952 alone, the record for the largest known prime was increased from 79 digits to 687, and the length of the largest known prime has continued to grow exponentially in time up to the present day (see Fig. 1).

From the 1950s until the mid 1990s interest in searching for large primes was confined mainly to professional mathematicians, who had access to the latest computer hardware. Throughout the 80s and early 90s, several record-sized primes were found using Cray supercomputers, culminating with a 378362 digit Mersenne prime in 1996. However, later that year, another innovation was to change the way in which large primes could be found – volunteer computing.

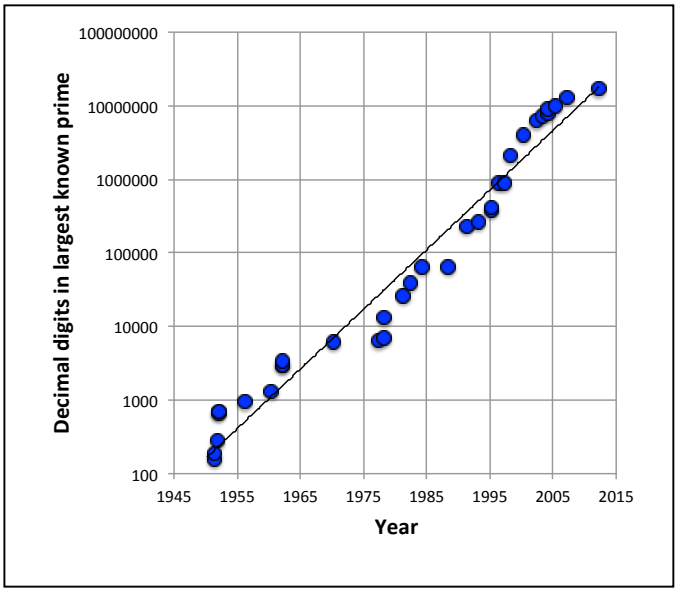

Figure 1. Size of largest known prime per year throughout the computer era, data from [3]

II. VOLUNTEER COMPUTING AND BOINC

Volunteer Computing is a paradigm within the field of Distributed Computing where a computational task, usually of a scientific nature, is subdivided into smaller work units that will be executed independently by volunteers' computers. Thus is it most suitable for 'embarrassingly parallel' problems, where each work unit requires no data from other work units, and so can be processed at the same time – in parallel. Volunteer Computing projects use a client/server architecture, where volunteers connect their computers (the clients) to a central server over the Internet, download work units, process them and return them to the server. Typically, the volunteers' contribution of CPU time is rewarded by a credit system, based on the number of work units, or some other measure of the amount of work done by each participant. Volunteer computing is thus a benefit to the project, which gets access to

This work was supported by the Centre for Numerical Algorithms and Intelligent Software, EPSRC grant EP/G036136/1.

potentially very large aggregate computational power at little or no cost, and also to the volunteers, who are able to engage with and contribute towards active scientific research. A fuller discussion of the motivations of participants and reward mechanisms is given in [4].

When volunteer computing first became popular in the mid 1990s, typical computer processors used approximately the same amount of electrical power whether they were running programs or idle. Thus volunteer computing was touted as a way to make use of these 'spare' processor cycles at little marginal cost to either the volunteer or the project. Today, modern CPUs such as the Intel i3/5/7 series have a much greater focus on power-saving features for use in battery powered devices, and so draw much less power when idle (for example only 35% [5]) then when under full load. The number of participating users in volunteer computing has stagnated and declined slightly since 2010 (see Fig. 2), and reason for this is that the cost of electricity usage to the participant is now non-negligible.

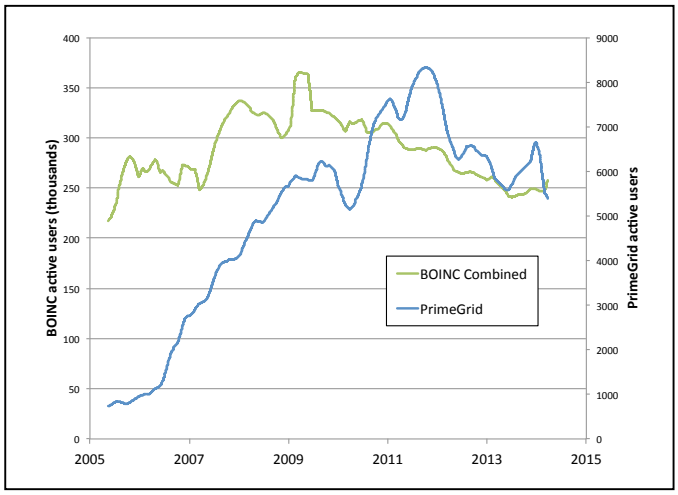

Figure 2. Number of active users - those with completed tasks in last 30 days - on all BOINC projects (data from [6]), compared with PrimeGrid.

The first example of a Volunteer Computing project was the Great Internet Mersenne Prime Search (GIMPS) [7], which was set up in 1996 by George Woltman, specifically focused on searching for large Mersenne primes. Primality testing is well suited for volunteer computing as the status of each candidate number is independent of any other, and so individual tests (or ranges of tests) can be packaged as work units and distributed to the client computers. GIMPS proven immediately to be very successful, and since finding a 420921 digit prime in 1996 has not only held the record for the largest known prime, but also extended it by finding a further 13 new primes. Today, the largest known prime is $2^{57885161}$-1, which has some 17 million digits! GIMPS uses PrimeNet, a client/server system written by Scott Kurowski specifically to handle the generation, distribution and management of GIMPS work units.

Another high profile early use of Volunteer Computing was SETI@Home [8], first made public in 1999. Developed by the Space Sciences Laboratory at Berkeley, SETI@Home analyses data from radio telescopes to detect potential signals from extraterrestrial civilisations. Recorded signals from the Arecibo raido telescopt are stored, split into short blocks, and distributed to volunteers' computers for analysis. While no such signal has yet been discovered, SETI@Home has had two major successes. Firstly, it attracted a great deal of publicity which has drawn in a large user base (over 1.5 million users since the project began), some of whom also contribute to other projects. Secondly, the client and server software developed by the SETI@Home team was generalized and released in 2002 as the Berkeley Open Infrastructure for Network Computing (BOINC) [9].

BOINC is now the largest platform for Volunteer Computing, with over 3.2 million users, around 70 projects in total (including SETI@Home), and an aggregate performance of around 7 PFLOP/s, equivalent to the 6th largest supercomputer in the world [10]. The BOINC architecture consists of three parts: a server package, which is deployed by each individual project, providing a website, an administration interface for creating work units, and a web service which sends tasks to and receives results from connected computers; the client, which is installed by volunteers on their computers and set to connect to one or more project servers to receive work, manages the execution of a queue of work units subject to user preferences, and returns the results to the servers; and a library, which is used by software developers to enable their programs to communicate with the client, handles common functionality such as suspending and resuming computation, checkpointing and progress reporting.

Running a project using BOINC is advantageous to the project administrators, as they are able to concentrate on the scientific aims of the project, while BOINC helps to deal with issues which affect all volunteer computing projects:

- Unreliable client computers: due to hardware or software errors, it is possible for clients to return incorrect or incomplete results for a work unit. BOINC can automatically replicate work units to multiple clients, and compare the results. If they do not agree, subsequent replicates of the work unit are generated, until a consensus correct result is obtained and the erroneous results are marked as invalid.
- Slow or disconnected computers: work units are created with a particular deadline, typically a few times the expected processing time. Once a work unit has been sent to the client, the client manages the execution of all the work units in its queue in order to meet the deadlines. If a work unit passes its deadline and the client has not yet started work, it will report to the server that it has abandonded the test. If the server does not receive any response from the client before a work unit deadline has expired, it will automatically send out the work unit to another client. Thus overall progress can be made on the project, without being unduly delayed by individual clients
- Hardware and software heterogeneity: BOINC supports a wide range of operating systems (Windows, Mac OS X, Linux, and more) and

hardware (x86 CPUs, GPUs, mobile devices). Projects need only to provide application versions to run on some particular combination of OS and hardware, and BOINC automatically detects the capabilities of the client computer and distributes only suitable work units. The capability for users to compile their own optimized or ported versions of project applications is also available.

III. PRIMEGRID

In June 2005 a BOINC project called Message@Home was set up by Rytis Slatkevičius, a Lithuanian secondary school student, as a test-bed for a Perl implementation of BOINC. As example applications, the brute-force decryption of a message encoded with the MD5 algorithm [11] and the factorization of the RSA-640 number [12] were chosen as they had suitably small-sized work units. In September 2005 the MD5 application was discontinued and project name was changed to PrimeGrid. By November, RSA-640 was successfully factorized by another group [13], and the project moved on to attempt factorization of RSA-768. The following year, the RSA-768 effort too was abandoned and an attempt was made to generate a list of primes starting from 2. After computing all primes up to 6.4×10^{11} (approximately 23 billion prime numbers), this project was again abandoned. The project finally found lasting success through collaboration with the (now defunct) Riesel Sieve project, another independent volunteer computing project already established in the prime searching community, to find large Twin primes (primes which have a difference of 2). In November 2006, the PrimeGrid Twin Prime Search was launched, and after only two months of work, found a then-record Twin prime pair - $2003663613 \times 2^{195000} \pm 1$. The 'twin prime conjecture' states that there are infinitely many pairs of twin primes, and though recent progress [14] showed that infinitely many primes of primes separated by less than 70,000,000 exist, the conjecture remains unproven. PrimeGrid subsequently extended the record twice, most recently in 2011 with the discovery of $3756801695685 \times 2^{666669} \pm 1$, which has 200700 digits.

Since 2006 the project has grown dramatically, with the addition of several new prime searches available through BOINC, which are summarized in Table 1. Each subproject is either searching for larger and larger primes of a given form, for example Generalised Fermat Primes $b^{2^n}+1$, or attempting to prove a specific Number Theoretic conjecture. For example the Seventeen or Bust subproject arises from a theorem by Sierpiński in 1960 [15] that there exist infinitely many odd integers k such that all Proth numbers $k \times 2^n + 1$ are composite irrespective of n. Within 2 years Selfridge showed by the use of covering sets that k = 78557 has this property, and it is an example of what is now called a Sierpiński number. The Sierpiński problem is then to show that 78557 is the smallest such number, which can be done by exhibiting a prime for each odd $k < 78557$. By 2002 only 17 values of k remained for which no prime was known, and an organized search was set up known as 'Seventeen or Bust!', using a volunteer computing system developed by Louie Helm and David Norris [16]. By 2007, the list of remaining k was reduced to 6, and after several years without further finds, they joined with PrimeGrid to continue the search. Similar conjectures exist for Riesel numbers and Proth numbers with prime k. The latest status and full history of these conjectures is discussed in Section III.B and can be found online [17][18].

As PrimeGrid has grown in popularity it has become one of the largest volunteer computing projects in the world (see Table 2), and is now run by a team of volunteers, including the author. As a result, almost all of the subprojects have yielded world-record sized primes, the most recent of which are listed below:

- Largest Cullen prime: $6679881 \times 2^{6679881} + 1$ (2.01 million digits), discovered in July 2009 by Magnus Bergman.
- Largest Woodall prime: $3752948 \times 2^{3752948} - 1$ (1.13 million digits), discovered in December 2007 by Matthew Thompson.
- Largest Twin prime: $3756801695685 \times 2^{666669} \pm 1$ (200700 digits), discovered in December 2011 by Timothy Winslow.
- Largest Sophie Germain prime: $18543637900515 \times 2^{666667} - 1$ (200701 digits), discovered in April 2012 by Philipp Bliedung.
- Largest Generalised Fermat prime: $475856^{524288} + 1$ (2.98 million digits), discovered in August 2012 by Masashi Kumagai.
- Largest Riesel prime: $502573 \times 2^{7181987} - 1$ (2.16 million digits), discovered in October 2014 by Denis Iakovlev.
- Longest arithmetic progression of primes: $43142746595714191 + 23681770 \times 23\# \times n$ for $n = 0 .. 25$, discovered in April 2010 by Benoãt Perichon (note 23# denotes the 'primorial' product of all primes ≤ 23).
- Most mega-primes: As of October 2014, PrimeGrid has discovered 63 primes with over 1 million digits (mega-primes), out of a total of 114 known [3]

PrimeGrid currently has three aims. Firstly, we engage members of the public in active mathematical research, and give them the chance to be the discoverer of very large primes. The first finder is always given primary credit for the discovery, and the associated kudos is an important motivation for many of our users. Secondly, we provide education about computational number theory and mathematics in general. This takes place mainly in our active user forum, where there are a mixture of expert mathematicians, amateur enthusiasts, and complete novices – ultimately everyone benefits from this interaction. There is also a great deal of user involvement in the development and testing of new applications, resulting in the award of the 'Volunteer Tester' title. Thirdly, but not least, we aim to make a significant contribution to the field of Number Theory. Besides the record-sized primes reported above, PrimeGrid also contributes to several important mathematical projects:

TABLE I. DETAILS OF CURRENT AND PREVIOUS PRIMEGRID SUBPROJECTS RUNNING ON BOINC

Subproject	Date Started	Notes
Twin Prime Search	Nov 2006	Merged with Sophie Germain Search in Aug 2009
Cullen Prime Search	Aug 2007	$n \times 2^n+1$
Woodall Prime Search	Aug 2007	$n \times 2^n -1$
Proth Prime Search	Feb 2008	$k \times 2^n +1$, $k < 2^n$
321 Prime Search	Jun 2008	Proth primes, k=3
Prime Sierpiński Problem	Jul 2008	Proving 271129 is the smallest prime Sierpiński number
AP26 Search	Dec 2008	Searching for sequences of 26 primes in arithmetic progression, search completed in April 2010
Sophie Germain Search	Aug 2009	Searching for Sophie Germain pairs of primes p, 2p + 1, and Twin primes
Seventeen or Bust!	Jan 2010	Proving 78557 is the smallest Sierpiński number
The Riesel Problem	Mar 2010	Proving 509203 is the smallest Riesel number
Generalised Fermat Prime Search	Jan 2012	$b^{2^n} + 1$ for n=20,22
Sierpiński /Riesel Base 5 Problem	Jun 2013	Proving 159986 and 346802 are the smallest base 5 Sierpiński and Riesel numbers respectively
Extended Sierpiński Problem	Jun 2014	Proving 271129 is the second Sierpiński number

TABLE II. COMPARISON OF PRIMEGRID WITH OTHER MAJOR VOLUNTEER COMPUTING PROJECTS (OCT 2014)

Project	Total Users	Current Users	Current Computers	Current Performance
GIMPS	128,000	3,800	22,000	0.2 PFLOP/s
PrimeGrid	84,400	11,100	16,500	1.1 PFLOP/s
Collatz Conjecture	37,000	3,200	6,000	1.3 PFLOP/s
GPUGrid	24,500	2,700	4,000	1.4 PFLOP/s
SETI@Home	1,512,300	121,800	2,600,000	0.7 PFLOP/s

A. *Fermat Number Factoring*

The Fermat numbers $F_n = 2^{2^n} + 1$, named for Pierre de Fermat who first studied them in the 17th century, have been shown to be prime for n = 0 .. 5, but no others Fermat primes are known. Indeed, the size of these numbers grows so rapidly that determining their status as prime or composite (and if so fully factorizing them) is a significant computational task. Lucas proved that all factors of Fermat numbers have the form $k \times 2^{n+1} + 1$, i.e. they must be Proth Primes. As a result, even although most primes found in PrimeGrid's Proth Prime search are relatively small (having around 400,000 digits), they are each tested to see if they divide any Fermat numbers. At present, F_5 through F_{32} are proven composite, although not all are yet complete factorised. The largest Fermat number $F_{3329780}$ whose status is known (composite) was shown to have a factor $193 \times 2^{3329782} + 1$, found by Raymond Ottusch in July 2014, which is the first and only known mega-prime Fermat factor. The latest status of the Fermat Factoring project is recorded by Wilfrid Keller [19] but it remains unproven if any further Fermat primes exist, and the search continues.

B. *Conjecture Subprojects*

In addition to the Sierpiński conjecture and the Seventeen or Bust (SoB) sub-project mentioned earliere, PrimeGrid leads the search for the primes needed to prove a number of related conjectures. 78557 is the smallest known Sierpiński number, and 271129 is the second. 271129 happens to be prime, and so the Prime Sierpiński Problem (PSP) is to show that it is indeed the smallest such prime. Assuming the resolution of the Sierpiński conjecture, an additional 16029 prime values of $78557 < k < 271129$ must be tested to find Proth Primes. Primegrid tests increasing values of n for each k, until a prime is found at which point that k is removed from the search. To date only 8 k's remain without a known prime.

Assuming proof of the Prime Sierpiński Problem, a third conjecture may be made, that 271129 is the second Sierpiński number, which can be proven by exhibiting a Proth prime for each of the 80256 composite numbers $78557 < k < 271129$. This task is known as the Extended Sierpiński Problem (ESP). Currently 12 k's remain in this search.

Similarly to Sierpiński numbers, k's for which no Proth Primes exist, it is possible to define a Riesel number as a k for which no Riesel Primes exist. 509203 is the smallest known Riesel number, and proving that this is indeed the smallest is significantly more challenging than the Sierpiński Problem since over 6 times as many k need to be tested. As a result 50 k's remain in the search. However, due to the greater number of tests, the current value of n which has been reached (7.2 million) is lower than the other both SoB at 27 million and PSP at 17 million. Since 509203 is prime, there is no equivalent of the Prime or Extended Sierpiński Problems for the Riesel case.

Likewise, these conjectures exist for bases other than 2, and PrimeGrid is also leading the search for primes to prove the Riesel and Sierpiński base 5 conjectures. We will not discuss these in detail here, but the latest status is available online [20]. Overall progress on these projects is steady, as shown in Fig. 3.

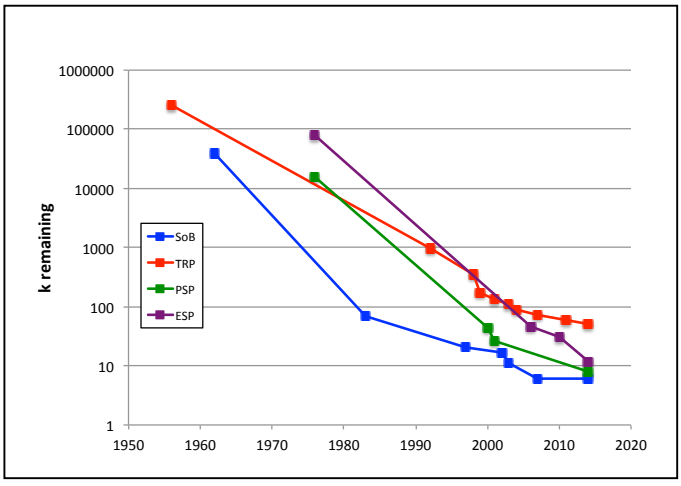

Figure 3. k remaining in each conjecture sub-project against time

C. Wieferich and Wall-Sun-Sun Primes

While the majority of PrimeGrid subprojects find many large primes, we also are working to find examples of two smaller, but extremely rare classes of primes.

A Wieferich prime is defined as a prime p which satisfies the condition that:

$$p^2 \mid 2^{p-1} - 1 \quad (1)$$

They were first described by in 1909 by Arthur Wieferich [21], who showed that if Fermat's Last Theorem were false for a particular prime exponent p, then that p is a Wieferich prime. The converse does not hold, of course. Only two Wieferich primes are known, p = 1093 and 3511, and Crandall *et al* [22] showed via a computational search that no others exist up to 4×10^{12}. This bound was subsequently extended to 6.7×10^{15} by others including Dorais and Klyve [23] without finding further primes. Nevertheless, there are expected to be infinitely many Wieferich primes, with each p having approximately 1/p 'probability' of satisfying the condition. We have extended the search up to 3.2×10^{17}, and no further Wieferich primes have yet been discovered.

Another class of primes are those which satisfy:

$$p^2 \mid F_{p-\left(\frac{p}{5}\right)}, \; p > 5 \quad (2)$$

They are known as Wall-Sun-Sun [24][25] primes. Unlike Wieferich primes, no Wall-Sun-Sun primes have yet been found, despite several prior attempts which verified none exist for $p < 9.7 \times 10^{14}$. We have continued up to 6.8×10^{16} without success, and the search is ongoing.

D. A World Record GFN Prime

Since 1989 the largest known prime has always been a Mersenne prime. While the Generalised Fermat Number ($b^{2^n} + 1$) prime search has been running for n < 20 since 2009 using PRPNet (see Section IV.C), the development of more efficient software using high performance Graphics Processing Units (GPUs) [26] enabled the search to be moved to BOINC and start testing higher values of n. Currently the n = 22 search has reached b = 29230, so all candidates being tested have at least 18.7 million digits. Thus any new prime would become the largest prime ever found, and the first non-Mersenne to hold this position for over 25 years.

IV. Algorithms and Software

None of the our achievements would have been possible without the support of a vibrant community of software developers who write the programs used by the project. We use a wide range of software, most of which is developed by members of the PrimeGrid user community and is open source. The most important of these are described below.

A. Sieving

The first stage in a particular prime search subproject is to reduce the number of candidates by sieving – that is testing divisibility by a large number of fairly small primes (e.g. up to $\sim 10^{17}$, for TRP Sieve) to find and remove composite candidates. Sieving is carried out to a particular depth (the size of the trial factors) depending on the relative speed of removal of candidates by sieving and by direct primality testing. Once the CPU time taken to find a factor becomes more than twice (to allow for the fact that tests are replicated to two clients) the amount of time taken to primality test a candidate, the optimal sieve depth has been reached and the sieve is stopped. The optimal sieve depth depends on the efficiency of both the sieving and primality testing software, as well as the quantity and type of hardware being used for each, and so sieving activity is kept under regular review by the PrimeGrid administrators. Typically a sieve is run concurrently with the associated primality testing sub-project(s), with the sieve works on a much higher range of candidates. Those candidates which pass through the sieve are then added to the list for primality testing.

For Proth and Riesel primes two approaches are used, fixed-n and fixed-k sieving. Fixed-n sieving is employed when a large range of k are to being and there are only a small number (or even 1) of values of n, such as in the case of the Sophie Germain search sub-project. Since the aim is to find factors p such that (for Riesel numbers):

$$k \times b^n - 1 = 0 \pmod{p} \quad (3)$$

Since n is fixed, then

$$k = b^{-n} \pmod{p} \quad (4)$$

Thus the sieving program must compute $b^{-n} \pmod{p}$ for each p with the range being used as a sieve, and if any of the computed values of k falls within the range of candidates, then a factor is found and that k can be eliminated. An analagous expression also exists for Proth numbers, where k is negated. This process is implemented in the ppsieve and tpsieve programs developed by Ken Brazier [27]. As well as implementation on CPUs, very efficient versions for GPUs using CUDA [28] and OpenCL [29] have been developed, to the point where most fixed-n sieving is now done exclusively using GPUs.

By contrast the conjecture projects, which have very few k remaining, used the fixed-k approach. Similarly to fixed-n, we start with the condition for p to be a factor of a Riesel Number:

$$k \times b^n - 1 = 0 \pmod{p} \quad (5)$$

Then for fixed k,

$$b^n = 1/k \pmod p \quad (6)$$

$$n = \log_b(1/k) \pmod p \quad (7)$$

Thus for each p we must compute 1/k (mod p) – the modular multiplicative inverse, and then the discrete logarithm in base b to determine whether a given p factors any candidate value in the range, with the calculated n. These algorithms have been have been implemented in the srsieve program by Geoff Reynolds and Mark Rodenkirch [30]. No GPU implementation has yet been developed.

Lastly, for Generalised Fermat Numbers an algorithm was devised by Carmody [31] which has been implemented in various forms for different hardware. Most recently a CUDA GPU version was developed by Anand Nair in 2012. Within six months of work sieving was completed to a depth of 1.9×10^{19} (for n = 19), and sieving is currently suspended.

B. *Primality Testing*

The remaining candidates after sieving are then individually tested for primality, using a variety of different algorithms depending on the form of number.

For Riesel numbers, the Lucas-Lehmer-Riesel test [32] is used, which computes the sequence:

$$u_i = u_{i-1}^2 - 2 \quad (8)$$

Starting from a particular u_0 (depending on k), it is a necessary and sufficient condition for p to be prime that $p = k \times 2^n - 1 \mid u_{n-2}$. In practice, the sequence is computed modulo p, using large integer multiplication based on Discrete Weighted Transforms [33]. Efficient DWT computational kernels for x86-based CPUs have been developed by George Woltman [34], originally for use by GIMPS, and these have been incorporated into Jean Penné's 'LLR' program [35], which is used for the majority of subprojects on PrimeGrid.

For Proth numbers, we make use of Proth's theorem that if:

$$a^{(p-1)/2} = -1 \pmod p, \quad \text{where } \left(\frac{a}{p}\right) = -1 \quad (9)$$

then p is prime. Similarly to the LLR test, the repeated squaring of a is performed modulo p, and this algorithm is also implemented in the LLR program.

In the case of Generalised Fermat Numbers, we have a very efficient implementation of Fermat's Little Theorem in the Genefer program [36][37] that if p is prime, then

$$a^{p-1} = 1 \pmod p, \quad 1 \le a < p \quad (10)$$

Note that this is a necessary condition for p to be prime, but not sufficient – if the equality fails for a single value of a then p is composite. We perform a test using a single value of a, and if the equality holds, we say p is probably prime (PRP) and then perform a subsequent deterministic test (for example an N−1 test, as described in [38]). Because the fraction of primes is very small, this process is still very efficient, as very few deterministic primality tests are ever performed. Of the over 1000 GFN PRPs we have found to date, none have in fact turned out to be composite.

Alongside LLR and Genefer, there are also many special-purpose programs that we use, for example in the AP26, Wieferich and Wall-Sun-Sun subprojects, which we will not describe here. The most significant of these is OpenPFGW [39], which implements both deterministic and PRP tests for a very wide range of candidate forms, especially the $b \neq 2$, Primorial and Factorial primes tested on PRPNet.

C. *PRPNet*

The majority of PrimeGrid's work is done using BOINC as discussed in section III. However, for some sub-projects we make use of PRPNet [40], a client/server system written by Mark Rodenkirch, specifically designed for prime searching. While not as fully-featured as BOINC, for example lacking a Graphical User Interface, PRPNet has the advantage that the code can easily be modified or extended to support testing with applications which cannot be easily integrated with BOINC. Thus it is used mainly for either the initial stage of searches that are then subsequently moved to BOINC, or for specialised projects which do not attract enough users to make it worth porting the application to BOINC. Table 3 lists the currently active projects.

TABLE III. DETAILS OF CURRENT PRIMEGRID SUBPROJECTS RUNNING ON PRPNET

Subproject	**Notes**
27/121 Search	Proth and Riesel Primes for k=27 and 121
Factorial Prime Search	Primes of the form n! ± 1
Generalised Cullen/Woodall Base 13	Primes of the form $n \times b^n \pm 1$, for b where no such prime is yet known
Generalised Fermat Prime Search	$b^{2^n} + 1$, for n=15,16,18,19
Primorial Prime Search	Primes of the form p# ± 1, where # denotes the 'primorial' product of all primes ≤ p
Wieferich Prime Search	Searching for a third Wieferich prime
Wall-Sun-Sun Prime Search	Searching for the first Wall-Sun-Sun prime

V. CONCLUSION AND OUTLOOK

Since it was founded in 2005, PrimeGrid has established itself as a major contributor to the field of computational number theory, driving software and methodological developments and breaking many records along the way. However, perhaps the most significant achievement of the project has been to generate public engagement with Mathematics and build a large, active user community, whose contribution of computer time has enabled all of the results we have highlighted.

As a result, we are well-placed to continue work on existing sub-projects and branch out into new areas, for example searching for new prime zeros of Ramanujan's modular tau function [41]. We also welcome new collaborations with researchers in computational number theory with problems that could be addressed by Volunteer Computing.

Against a background of declining interest in Volunteer Computing, PrimeGrid's ongoing growth is a testament to the powerful appeal of finding large primes, and we are sure to continue this for the foreseeable future.

ACKNOWLEDGMENT

PrimeGrid was made possible and is kept running on a daily basis by a team of volunteer administrators and developers. Particular thanks are owed to Rytis Slatkevičius, Lennart Vogel, Mike Goetz, Jim Breslin and John Blazek. We also thank Rackspace for providing the virtual servers and storage used by the PrimeGrid project free-of-charge, and finally all the volunteer 'crunchers' who contribute the computing time that has made PrimeGrid a success.

REFERENCES

[1] B. Riemann, "Ueber die Anzahl der Primzahlen unter einer gegebene Grösse," Monatsberichte der Berliner Akademie, 1859.

[2] P. A. Cataldi, "Trattato de nvmeri perfetti di Pietro Antonio Cataldo," Presso gli Heredi di Giovanni Rossi, Bologna, 1603.

[3] C. Caldwell, "The List of Largest Known Primes," http://primes.utm.edu/primes/.

[4] P. Darch and Annamaria Carusi, "Retaining volunteers in volunteer computing projects", Phil. Trans. R. Soc. A, vol. 268, no. 1926, 2010, pp. 4177-4192, doi:10.1098/rsta.2010.0163.

[5] M. Hähnel, B. Döbel, M. Völp, and H. Härtig, "Measuring energy consumption for short code paths using RAPL," SIGMETRICS Perform. Eval. Rev. 40, 3, 2012, pp. 13-17, doi:10.1145/2425248.2425252 .

[6] N. Jones, "Computer sharing loses momentum," Nature, 506, 2014, pp.16-17, doi:10.1038/506016a.

[7] G. Woltman and S. Kurowski, "The Great Internet Mersenne Prime Search", http://www.mersenne.org

[8] E. Korpela, D. Werthimer, D. Anderson, J. Cobb and M. Lebofsky , "SETA@home - Massively Distributed Computing for SETI," Computing in Science & Engineering, 3, 2001, pp. 78-83, doi:10.1109/5992.895191.

[9] D. P. Anderson, "BOINC: A System for Public-Resource Computing and Storage", 5th IEEE/ACM International Workshop on Grid Computing, 2004.

[10] "Top 500 Supercomputers list: June 2014", http://www.top500.org/lists/2014/06/.

[11] "RFC 1321 – The MD5 Message-Digest Algorithm," Internet Engineering Task Force, 1992. https://tools.ietf.org/html/rfc1321.

[12] "The RSA Factoring Challenge", RSA Laboratories, http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-factoring-challenge-faq.htm.

[13] J. Franke "We have factored RSA640 by GNFS", 2005, http://www.crypto-world.com/announcements/rsa640.txt.

[14] Y. Zhang, "Bounded gaps between primes," Annals of Mathematics, 2013.

[15] W. Sierpiński, "Sur un problème concernant les nombres $k \cdot 2^n + 1$," Elem. Math, 15, 1960, pp. 73-74

[16] L. Helm and D. Norris, "Seventeen or Bust! A distributed attack on the Sierpiński Problem," http://www.seventeenorbust.com.

[17] W. Keller, "The Sierpiński Problem: Definition and Status," http://www.prothsearch.net/sierp.html.

[18] W. Keller, "The Riesel Problem: Definition and Status," http://www.prothsearch.net/rieselprob.html.

[19] W. Keller, "Prime factors $k \cdot 2^n + 1$ of Fermat numbers F_m and complete factoring status," http://www.prothsearch.net/fermat.html.

[20] "Sierpinski/Riesel Base 5 Problem," http://primegrid.wikia.com/wiki/Sierpinski/Riesel_Base_5_Problem.

[21] A. Wieferich, "Zum letzten Fermat'schen Theorem," J. Reine Angew. Math, 136, 1909, pp. 293– 302.

[22] R. Crandall, K. Dilcher and C. Pomerance, "A search for Wieferich and Wilson primes," Math. Comp, 66, 1997, no. 217, pp. 433–449, doi: 10.1090/S0025-5718-97-00791-6.

[23] F. G. Dorais and D. Klyve, "A Wieferich Prime Search up to 6.7×10^{15}," Journal of Integer Sequences, 2011.

[24] D. D. Wall, "Fibonacci Series Modulo m," American Mathematical Monthly, 67, 6, 1960, pp. 525-532, doi:10.2307/2309169.

[25] Z-H. Sun and Z-W Sun, "Fibonacci number and Fermat's last theorem," Acta Arithmetica, 60, 4, 1992, pp. 371-388.

[26] I. Bethune and M. Goetz, "Extending the generalized Fermat prime search beyond one million digits using GPUs," Proceedings of the 10th International Conference on Parallel Processing and Applied Mathematics, PPAM 2013, Lecture Notes in Computer Science 8384, 2014, pp. 106-113, doi:10.1007/978-3-642-55224-3_11.

[27] K. Brazier, "Prime Programs," https://sites.google.com/site/kenscode/prime-programs.

[28] J. Nickolls, I. Buck, M. Garland and K. Skadron, "Scalable Parallel Programming with CUDA," Queue, 6, 2, 2008, pp. 40-53, doi:10.1145/1365490.1365500.

[29] J. E. Stone, D. Gohara, and G. Shi, "OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems," IEEE Des. Test, 12, 3, 2010, pp. 66-73, doi:10.1109/MCSE.2010.69.

[30] G. Reynolds, "srsieve", https://sites.google.com/site/geoffreywalterreynolds/programs/srsieve.

[31] P. Carmody, "GFN filters," http://fatphil.org/maths/GFN/maths.html.

[32] H. Riesel, "Lucasian Criteria for the Primality of $N = h \cdot 2n - 1$", Math. Comp, 23, 108, 1969, pp. 869-875.

[33] R. Crandall and B. Fagin, "Discrete weighted transforms and large-integer arithmetic," Math. Comp, 62, 1994, pp. 305–324.

[34] G. Woltman, "Free Mersenne Prime Search Software," http://www.mersenne.org/download/.

[35] J. Penné, "LLR home page," http://jpenne.free.fr/index2.html.

[36] H. Dubner and Y. Gallot, "Distribution of generalized Fermat prime numbers," Math. Comp, 71, 2002, pp. 825–832.

[37] I. Bethune and Y. Gallot, "Adaptive DFT algorithms for large-integer multiplication on modern desktop hardware," unpublished.

[38] H. C. Williams, "Édouard Lucas and primality testing," Canadian Math. Soc. Series of Monographs and Adv. Texts, 22, John Wiley & Sons, New York, NY, 1998.

[39] C. Nash, J. Fougeron and M. Rodenkirch, "OpenPFGW", http://sourceforge.net/projects/openpfgw/.

[40] M. Rodenkirch, "PRPNet, A client/server application used to help in the search for primes," http://sourceforge.net/projects/prpnet/.

[41] N. Lygeros and O. Rozier, "A new solution to the equation $\tau(p) = 0 \pmod p$," Journal of Integer Sequences, 13, 10.7.4, 2010.