Grid Computing for Empirical verification of the even Goldbach conjecture.

[1]Tomás Oliveira e Silva , [2]Siegfried Herzog and [3]Silvio Pardi (presenter)
[1]Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro –Portugal - email tos@ua.pt
[2]The Pennsylvania State University, Mont Alto Campus, USA – email hgn@psu.edu
[1]INFN - Sezione di Napoli – ITALY – email spardi@na.infn.it
Project Web Page: http://www.ieeta.pt/~tos/goldbach.html

Introduction

The Goldbach Conjecture represents one of the most fascinating mathematic challenge. The problem was formulated in 1742, it assert that every even number larger than four is the sum of two odd prime numbers. Despite the simple formulation and several efforts made by mathematicians of all world, the proof or disproof is still unknown.
The empirical verification of the Goldbach Conjecture was approached by several scientists since 1800, looking for counterexamples, additional information about numbers behaviors and to enlarge the problem knowledge. Better Numerics enable researchers to verify probabilistic conjectures, reduce the reliance on some theorems and would also give more precise bounds on the number of representations of numbers as the sum of primes.
In 2001 Prof.Tomas Oliveira e Silva Started a Project to extend the previous record of computation ($4\cdot10^{14}$)[2] up to $4\cdot10^{18}$. The project has taken advantage from the collaboration of several institution, then in 2011 it was powered by the GRID technologies over the INFN-Grid Infrastructure and the SCoPE datacenter part of the EGI infrastructure.

The Code

The run code finds the minimal Goldbach partition of every even integer larger than four. In order to do this efficiently, the computation intensive parts of the program were written in assembly language (for the IA32 instruction set). A very efficient cache friendly implementation of the segmented sieve of Eratosthenes was used to generate the prime numbers. For each interval of 10^{12} integers, we record the number of times each (small) prime is used in a minimal Goldbach partition, as well as the even integer where it was first needed. Because it takes very little extra time, we also record information about the gaps between consecutive primes, viz., how many times each gap occurs, and its first occurrence. On a single core of a 3.3GHz core i3 processor, testing an interval of 10^{12} integers near 10^{18} takes close to 48 minutes. The execution time of the program grows very slowly, like log(N), where N is the last integer of the interval being tested, and it uses an amount of memory that is roughly given by 13 sqrt(N) / log(N).

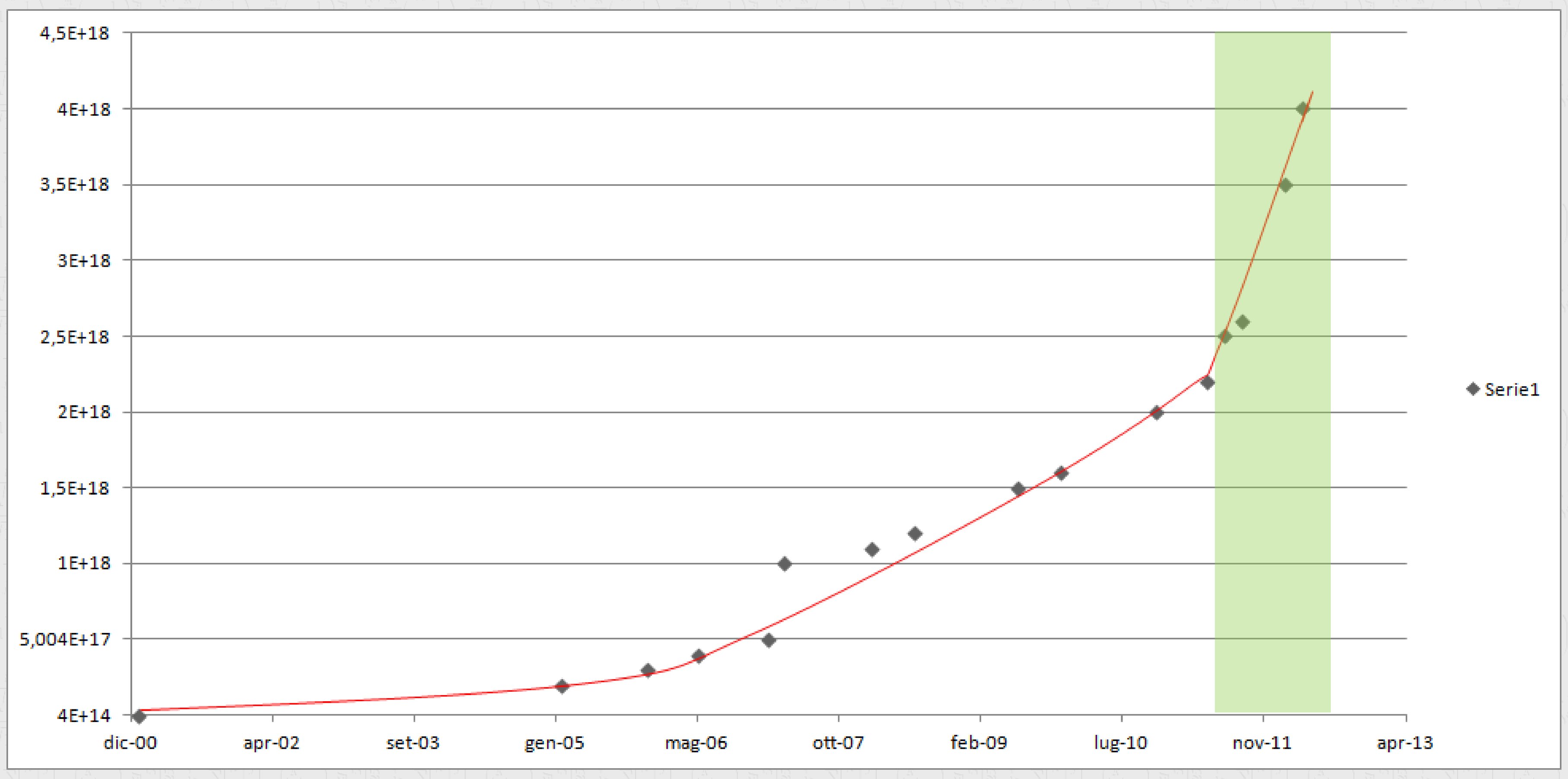
The Grid Porting

The developed code can work both following a master-worker paradigm to automatically manage the computations and in standalone version. We used the second option in order to decouple the Grid implementation from the legacy computation already on going.
The porting works as follow:
Given a pool of numbers to evaluate, we have created a set of standalone Grid jobs, each one requiring about 6h of intensive computation on a single core. Then we have used the Parametric functionality provided by the WMS service, in order to distribute this large amount of jobs and retrieve the results efficiently.
A set of scripts analyzes the incoming results looking for submission or execution errors, when a fault occurs, the specific job of the pool is automatically rescheduled on the Grid with a separate JDL created at run-time. Furthermore the analysis scripts maintain the blacklist of all the Computing Elements in which the computation fails, in order to improve the reliability through the use of a real-time fault avoidance strategy.

```
JobType = "Parametric";
Executable = "xclient.sh";
StdOutput = "goldbach_out_PARAM_.txt";
StdError = "goldbach_err_PARAM_.txt";
Parameters = 300;
ParameterStart = 100;
ParameterStep = 1;
Arguments = " _PARAM_ ";
InputSandbox =
{"client","xclient.sh","_PARAM_"};
OutputSandbox = {"goldbach_out_PARAM_.txt",
"goldbach_err_PARAM_.txt","c_state_out_00" };
```

Example of Parametric JDL used to distribute the computation on Grid

The Computation on Grid started in September 2011 and finished in April 2012. In 7 months we analyze about 19% of the total tasks amount. The graph below shows that the introduction of Grid facilities affect dramatically the computation by increasing the mean speed over the previous 6 years of a factor 6.



10 largest *g* (prime gaps) found

Rank	*g*	*P(g)*	Discoverer
1	1476	1425 17282 44376 99411	Tomás Oliveira e Silva
2	1454	3219 10718 24928 71783	GRID - Silvio Pardi
3	1442	804 21283 06866 77669	Siegfried "Zig" Herzog
4	1418	3725 23553 35041 01511	Tomás Oliveira e Silva
5	1416	3750 99252 93399 78877	Tomás Oliveira e Silva
6	1410	2635 28193 24815 39903	Siegfried "Zig" Herzog
7	1400	3431 65779 58583 78003	GRID - Silvio Pardi
8	1398	2424 70872 97267 67749	Tomás Oliveira e Silva
9	1392	1480 03203 79396 34731	Tomás Oliveira e Silva
10	1390	3492 65766 10051 61107	GRID - Silvio Pardi

10 largest *p* (least primes of a Goldbach partition) found

Rank	*p*	*S(p)*	Discoverer
1	9781	3325 58170 73339 60528	GRID – Silvio pardi
2	9629	2795 93511 65744 69638	GRID – Silvio Pardi
3	9341	906 03057 95622 79642	John Fettig & Nahil Sobh
4	9203	1348 11357 94295 47486	Siegfried "Zig" Herzog
5	9161	887 12380 30778 37868	Siegfried "Zig" Herzog
6	9091	3164 06916 06618 44912	GRID – Silvio Pardi
7	9001	3893 00922 74334 20582	Tomás Oliveira e Silva
8	8971	2588 35699 18831 39892	Tomás Oliveira e Silva
9	8951	914 47723 42519 16254	John Fettig & Nahil Sobh
10	8941	555 27435 15567 50822	Siegfried "Zig" Herzog

April 4, 2012 - $4*10^{18}$ our desired verification limit, reached.
New record of computation

Conclusions

Through this poster we demonstrate the strong impact of the pervasive use of Grid Technologies in field of pure mathematics.
Thanks to the large resources availability of the EGI Infrastructure, we achieved our prefixed computation goal at least 2/3 years before respect the previsions suggested by the trend.
Full Mathematical considerations are contained in a more extensive work.

[1] Project Web page http://www.ieeta.pt/~tos/goldbach.html
[2] Jörg Richstein, *Verifying the Goldbach conjecture up to 4·10^14*, Mathematics of Computation, vol. 70, no. 236, pp. 1745-1749, July 2000