

Grid porting of Bhabha scattering code through a master-worker scheme

Alejandro Lorca, José Luis Vázquez-Poletti, Eduardo Huedo,
Ignacio M. Llorente

Distributed System Architecture group,
Dpto. de Arquitectura de Computadores y Automática
Facultad de Informática, Universidad Complutense de Madrid,
C/ Prof. José García Santesmases s/n, E-28040 Madrid, Spain.
alejandro.lorca@fdi.ucm.es, jlvazquez@fdi.ucm.es, ehuedo@fdi.ucm.es,
llorente@dacya.ucm.es

Abstract. A program calculating Bhabha scattering at high energy colliders is considered for porting to the EGEE Grid infrastructure. The program code, which is a result of the AITALC project, is ported by using a master-worker operating scheme. The job submission, execution and monitoring are implemented using the GRIDWAY metascheduler. The unattended execution of jobs turned out to be complete and rather efficient, even when pre-knowledge of the grid is absent. While the batch of jobs remains organized at the user's side, the actual computation was carried out within the phenogrid virtual organization. The scientific results support the use of the small angle Bhabha scattering for the luminosity measurements of the International Linear Collider project.

1 Introduction

The International Linear Collider (ILC) is an electron-positron accelerator planned to supersede the Large Hadron Collider (LHC) and lead the high energy physics research in the nearby decades. Still, the proposal and supporting groups for the ILC are awaiting the first signals of LHC to complement their goals to the expected discoveries from the proton-proton collider. Thousands of scientists and engineers firmly believe [1,2] on the advantage coming from a cleaner environment resulting from the electron-positron collisions at the ILC. This translates into a clearer signal to background ratio than in the LHC due to the absence of plethora of hadronic subproducts. Therefore, this kind of accelerator could much precisely determine important parameters of the Standard Model of particle physics and thus lead to tighter model constraints and even discoveries “by precision”. Some high-precision measurements in history, like planetary motion, the speed of light or, more recently, deep inelastic scattering, led to remarkable advances in our understanding of astronomy, special relativity and quantum chromodynamics (QCD) respectively. They remind us the importance of such methodology.

The calculations involved in an accurate determination of any physical observable are, within perturbation theory in quantum field theory, rather cumbersome.

Fortunately, computing resources have grown fast enough during the last decades to accomplish these calculations despite of their increasing complexity. Automated software tools have arisen in the last thirty years to provide systematic and reliable answer for these predictions.

In this context, the EGEE project¹ provides a computing infrastructure where scientists and engineers perform numerous studies and tests in order to provide an efficient distributed architecture for Grid computing.

Along this paper we describe the scientific goals (Sec. 2) and the methodology we used for adapting a code for theoretical predictions to run properly on to the Grid (Sec. 3). Analyses on the execution of jobs (Sec. 4) and scientific results (Sec. 5) base the case study. Finally a look at further possibilities and conclusions finishes this contribution on Sec. 6.

2 Scientific scope

Nowadays, high energy physics applications intensively use Grid resources during data processing and analyses from LHC. Theoreticians also require heavy computational tasks to match the same level of accuracy achieved by the experimental measurements. In a close future, after the LHC experience, it could be very well established that Grid technologies are a standard procedure in order to achieve the permill level of uncertainty expected for the ILC.

In this article we will consider Bhabha scattering (i.e., the reaction of an electron-positron pair into themselves)

$$e^-e^+ \rightarrow e^-e^+, \quad (1)$$

as our target process.

One of the reasons to consider Bhabha scattering is the high cross section resulting at small angle due to the small deflection coming mainly from electromagnetic interactions. The resulting large amount of statistics within a well understood model allows for luminosity calibration. Such technique has been used at high energy colliders like LEP and SLD. Additionally, large angle Bhabha scattering has been also used at low energy *b*-quark factories like Belle or BaBar.

On the computational side, AITALC [3] is a useful tool for automating some calculations needed by theoreticians and phenomenologists in high energy physics. It produces numerical programs directly from the symbolic representation of the process, by using the so-called Feynman rules [4] and generating the complete analytic expressions for the respective Feynman diagrams within perturbation theory. Since the package is able to produce code for scattering of $2 \rightarrow 2$ fermions, it was suited to deliver the necessary routines to compute integrated cross sections for Bhabha scattering, including the first order corrections coming from electroweak quantum loop effects and soft-photon radiation.

The goal was to scan the full range of centre-of-mass energies, from the 10 GeV at the B-factories, to the 1000 GeV expected to be achieved at the second

¹ Enabling Grids for E-science, INFOS-RI-222667, 7th Framework Programme. <http://www.eu-egee.org/>.

running phase of the ILC. This range will be covered by 2048 data-points from both, tree-level (zeroth order) and first order corrected integrated cross sections. This coverage permits to quantify the impact of the quantum loops effects and relays in an complementary simulation of hard-photon effects [5] by Monte Carlo programs [6]. Moreover two-loop photonic corrections [7,8,9] also play a role in order to complete the full theoretical prediction.

3 Porting the application

3.1 Preliminaries

Whilst AITALC is a tool integrating three other independent packages; FORM [10], DIANA [11] and LOOPTOOLS [12], the resulting code which brings the numerics to the end-user is built as a dynamically linked FORTRAN executable.

Our primary intention was to port to the Grid the complete process, the generating tool and the code execution.

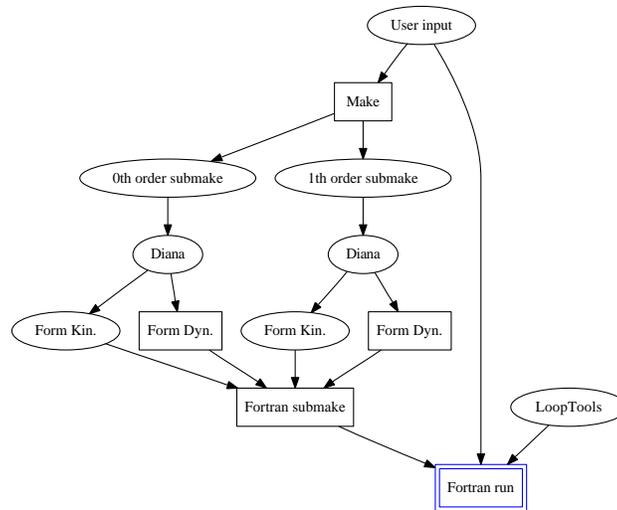


Fig. 1. Job work-flow for a typical process study with AITALC. Squares enclose those tasks with an internal structure suitable for some kind of parallelization. Double squared box shows the most suitable part to be ported to the Grid.

Fig. 1 describes the work-flow of the tool AITALC. There is a main make process which organizes the tree-level (0th order submake), the loop corrections (1st order submake) and the last numerical (FORTRAN submake) part through automated Makefiles. These three processes depend on each other as depicted, so the 0th and 1st order could be run in parallel. Because the complexity of the internal

calculations, it turns out that the 1st order demands much more resources than the 0th order. Still there are some internal parts of the branches, the dynamical ones under FORM (FORM Dyn.), which could in principle also be parallelized. The last block which evaluates the numerics (FORTRAN run) strongly depends on the user’s needs.

There are two main reasons why we considered only the final task to be ported to the Grid:

- **Independence.** AITALC runs natively on Unix machines, but even if it is not uncommon to find dedicated machines hosting FORTRAN compilers, FORM, DIANA and LOOPTOOLS, are very specialized software packages whose licences and distribution channels avoid out-of-the-box availability in standard Linux distributions. For this reason, AITALC supplies an installation script sorting out these inconveniences, but is unfeasible to provide a single executable and the whole tool with these three packages should be installed altogether in each running node. The later scenario immediately conflicts with standardized user permissions and node specifications.
- **Timings.** The production of the FORTRAN code does not take much time in a single machine for an example process. Our full-massive electroweak Bhabha scattering, being the largest process AITALC is able to generate, took not much more in our testing machine, just a few minutes to finish as shown in Tab. 1. Because the total time to evaluate 2048 data-points is approximately two orders of magnitude larger (e.g. Tabs. 4-7), we might neglect the time to create the FORTRAN code.

Task	Time [s]
0th order submake	12
1st order submake	381
FORTRAN submake	57

Table 1. Detailed running time for each building block of AITALC when producing the FORTRAN code for Bhabha scattering.

3.2 Master-worker scheme

The running profile of the application let the *master-worker* scheme being a very suitable one in order to achieve a balance between user intervention and execution performance. A scheme of the final application is depicted in the Fig. 2. It shows the work-flow consisting of a major master piece which decides how heavy the jobs are going to be (according to the user’s instructions) and a worker part being transferred and remotely run. The master communicates with the Grid via the `gwsuubmit` command of the GRIDWAY metascheduler [13]. GRIDWAY metascheduler offers a clean and user-friendly interface to submit jobs into a Grid middleware like GLOBUS or GLITE.

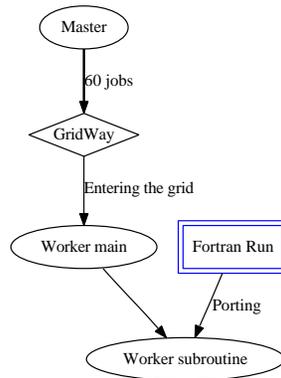


Fig. 2. Grid job work-flow. The GRIDWAY metascheduler acts as middleware establishing communication between master and worker.

Being GRIDWAY capable to perform unattended job migration, recovery and rescheduling [14], it was extremely useful for getting, with the generic and simplest configuration, a complete set of jobs being delivered for execution at different computing elements. Thus a deeper study of the queue and hardware characteristics of each cluster is not required.

Converting the FORTRAN program into a worker was a minor issue which is nevertheless worth to be mentioned since it is a common problem met by the scientific community. We chose to subdivide the worker into two parts:

- A simple main program, written in C++, accepting running parameters as arguments passed through command line interface. This main program calls, for each parameter configuration, a unique FORTRAN subroutine.
- The FORTRAN subroutine, which is an adaptation of the unported main program commenting out the settings for initial parameters. These are, according to AITALC description: `nsqrtsman`, `minsqrtsman`, `maxsqrtsman`, `setlimcost`, `setfracomega`.

Moreover, it was also required to ensure 32 bits compatibility and static linkage (usually flags `-m32 --static` in compilers). In such a way, the final executable was suited to be run in any working node without lacking any external libraries. The size of this executable (5.8MB) did not create transfer time bottlenecks.

4 Execution analysis

The scanning of data-points were planned in batches of jobs. Each of them contained specific instructions to generate automatically job templates with specific configuration for every job. This configuration had to do only with the definition of parameter range. The batches had equal amount of data to process, so we can consider the blocks equivalents to one another:

- 1st block: 4 jobs containing 512 data-points each,
- 2nd block: 8 jobs containing 256 data-points each,
- 3rd block: 16 jobs containing 128 data-points each,
- 4th block: 32 jobs containing 64 data-points each.

The evolution of a successful job is as follows: First, after a computing element has been chosen and accepted, the waiting time for a working node is denoted as *pending*. There is some *stage-in* time required to transfer data to the computing element. Then, the job enters in its *execution* stage. Finally a *stage-out* time is needed to transfer back the output and status. Moreover we have also to add some *overhead* due to job request processing and state change notifications. Occasionally the job gets migrated after a `suspension_time` configured by default in GRIDWAY to another computing element, in that case it appears in the graphics as *waiting*. Regarding totals, we noted by *computing* time the whole time summed up for the different machines and with the keyword *human*, the waiting time from batch start to the end.

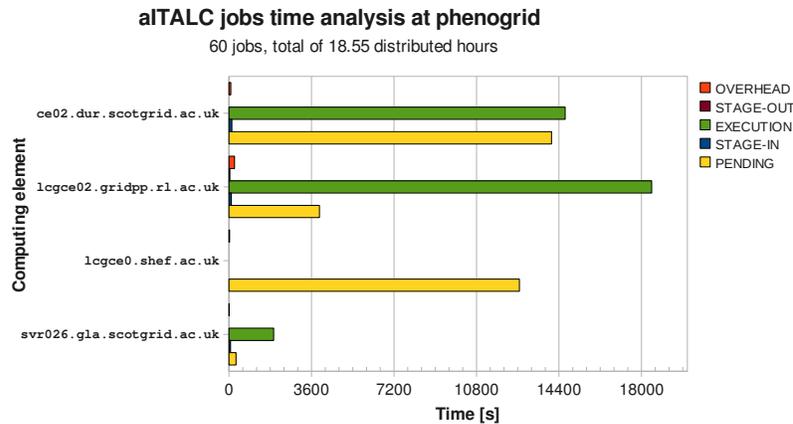


Fig. 3. Overall job-stage time for each computing element. The vertical order of bars is the same as shown in the legend.

All the jobs were submitted to the infrastructure phenogrid², which takes part in the EGEE project under the phenomenology-focussed virtual organization *pheno*. The timing analysis by cluster is given in Tab. 2 and graphically shown in Fig. 3. Three of the six computing elements worked properly and output data files were produced at regular rate. Other (`lcgce0.shef.ac.uk`) seemed to have a misconfiguration [15] and therefore left our jobs pending for too long and therefore being migrated automatically by GRIDWAY to most capable hosts. With the other

² Phenogrid, particle physics phenomenology on the grid.
<http://www.phenogrid.dur.ac.uk/>.

Computing Element	CPU		Nodes	Time performance [s]				
	Type	MHz		Total	Pend.	St.-in	Exec.	St.-out
FQDN	Xeon	2667	672	14074	127	14670	0	78
ce02.dur.scotgrid.ac.uk	Xeon	2667	672	14074	127	14670	0	78
lcgce0.shef.ac.uk	Opteron	2400	190	12673	0	0	0	27
lcgce02.gridpp.rl.ac.uk	P.III	1001	2525	3948	104	18447	41	243
svr026.gla.scotgrid.ac.uk	Opteron	2200	1896	306	66	1949	0	11
ce1.pp.rhul.ac.uk*	P.IV	1000	136	-	-	-	-	149
svr021.gla.scotgrid.ac.uk*	Opteron	1896	2200	-	-	-	-	18

* These computing elements reiteratively returned job-callback errors.

Table 2. Computing elements used for scheduling belonging to the pheno Virtual Organization at the EGEE infrastructure. Sums of short times rounded up to a second may be shown as zero, even if there was some accumulated latency.

Block	Time performance [s]					
	Pending	Stage-in	Execution	Stage-out	Overhead	Total
4 × 512	3980	11	8648	5	30	12674
8 × 256	9484	42	8939	3	39	18507
16 × 128	5989	86	8728	10	82	14895
32 × 64	11548	158	8751	23	208	20688
	Pend./job	St.-in/job	Exec./job	St.-out/job	Overh./job	Human
4 × 512	995.00	2.75	2162.00	1.25	7.50	3348
8 × 256	1185.5	5.25	1117.38	0.38	4.88	3131
16 × 128	374.31	5.38	545.50	0.63	5.13	1616
32 × 64	360.88	4.94	273.47	0.72	6.50	1328

Table 3. Time performance by block distribution of jobs. Precision is limited by rounded up of invidual timings to the second.

two computing elements (`ce1.pp.rhul.ac.uk` and `svr021.gla.scotgrid.ac.uk`) we found some errors after submitting our jobs that could be related to the local resource management system (LRMS), so GRIDWAY applied a temporal banning policy to them to avoid unsuccessful retries. This is just one of the already implemented mechanisms to enhance performance without user intervention.

Having a look to Tab. 3 the Figs. 4-7, we can appreciate the fair behaviour of the Grid computing. The more jobs contains the block, less *human* time waits the end-user, so parallelization works as expected. Nonetheless for a short amount of jobs as we have in this study, we cannot expect a linear statistical reduction due to the timeout of the pending time which some computing elements introduce. This timeout is configured in GRIDWAY via the `SUSPENSION_TIMEOUT` parameter, and the end-user might include it in his job template. This setting, as well as many other performance enhance strategies, lie outside of this paper’s scope.

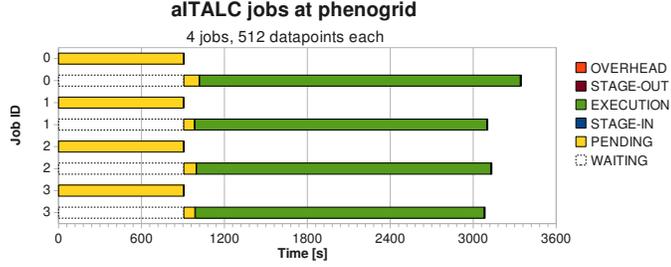


Fig. 4. Batch of 4 jobs computing 512 data-points each at phenogrid. The evolution of each job is split into separated bars every time it was migrated from computing element.

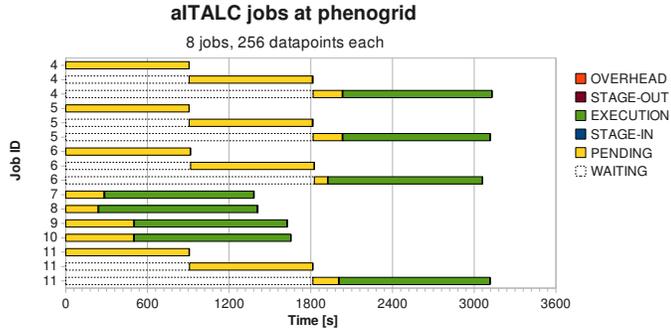


Fig. 5. Batch of 8 jobs computing 256 data-points each at phenogrid. The evolution of each job is split into separated bars every time it was migrated from computing element.

5 Scientific results

The correct execution of all the jobs let us compose the complete scan of the integrated cross section for Bhabha scattering. Two scans were performed with crossed exponential stepping to ensure that a late/failed job could still be interpolated from the rest without much loss of precision. The following two configurations for different scattering angle θ and maximum soft photon energy $E_{\gamma_{\text{soft}}}^{\text{max}}$ were considered:

- Large angle: $-0.9 < \cos \theta < 0.9$, $E_{\gamma_{\text{soft}}}^{\text{max}} = 0.1\sqrt{s}$
- Small angle: $25 \text{ mrad} < \theta < 90 \text{ mrad}$, $E_{\gamma_{\text{soft}}}^{\text{max}} = 0.2\sqrt{s}$

being \sqrt{s} the centre-of-mass energy.

We can observe the resonance induced by the gauge neutral Z -boson in Fig. 8. Here different symbols indicate different job identifiers, giving us therefore the complete result only when all the jobs are finished. The electroweak corrections

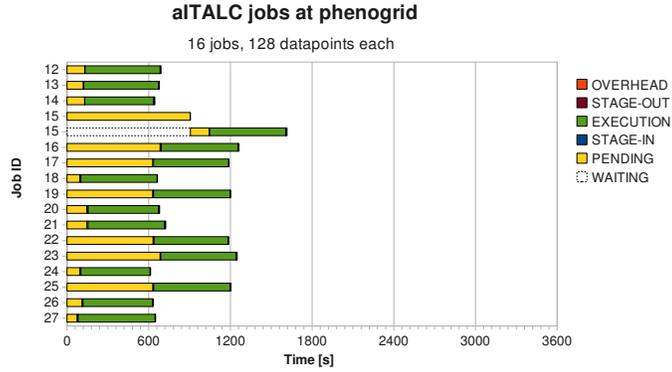


Fig. 6. Batch of 16 jobs computing 128 data-points each at phenogrid. The evolution of each job is split into separated bars every time it was migrated from computing element.

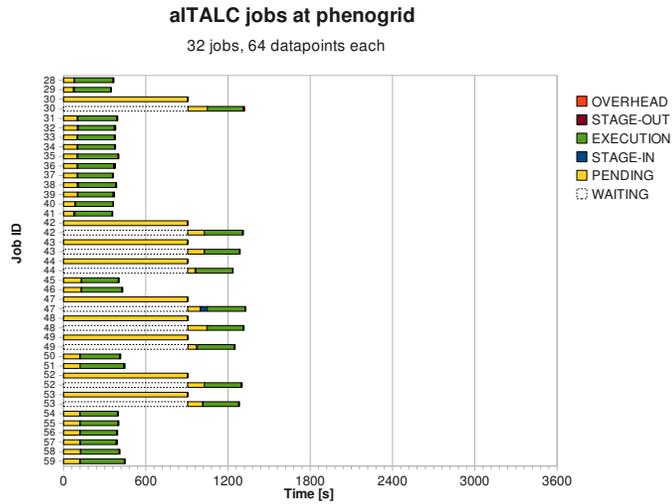


Fig. 7. Batch of 32 jobs computing 64 data-points each at phenogrid. The evolution of each job is split into separated bars every time it was migrated from computing element.

are in this range quite important at percent level, since the angular cross section is stopped at the angles close to collinearity.

Fig. 9 depicts the second configuration for small angle scattering in the forward region and the relative importance of the first order ($\mathcal{O}(\alpha)$) perturbative corrections. Here the resonance disappears due to the large contributions coming from the divergent Feynman diagram exchanging a photon at the collinear case.

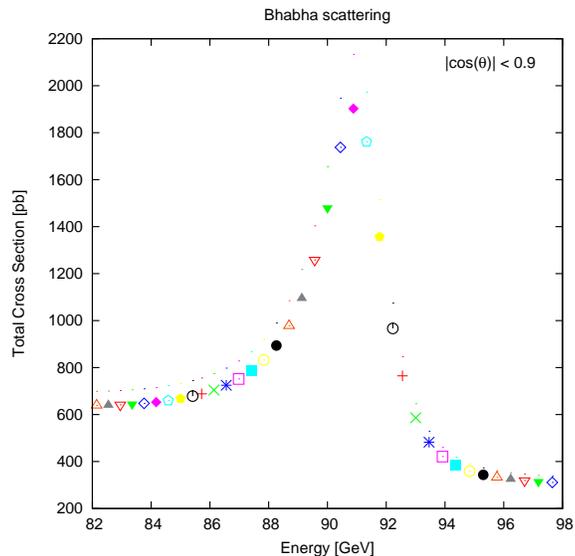


Fig. 8. Bhabha scattering around the Z -pole. Tiny dots stand for the tree-level while shaped symbols do it for the electroweak 1-loop plus soft photon corrections with $E_{\gamma_{\text{soft}}}^{\text{max}} = 0.1\sqrt{s}$. Each of the sixteen different symbols matches a different job identifier, being therefore the scanning of data-points computed in a distributed environment.

In this case the relative corrections are small, about the permill order: $\mathcal{O}(10^{-3})$. As mentioned before, this study should be supplemented with the hard-photon bremsstrahlung and second order perturbative corrections in order to specify the amount of uncertainty coming from the theoretical prediction.

6 Conclusions and outlook

Porting the application was successful with the help of the GRIDWAY metascheduler. A scheme of master/worker was developed where the master remain at user's side, giving instructions through GRIDWAY about how the submission should be partitioned and managed. Because the code created by AITALC is a dynamically linked executable, a few modifications to the original code were required in order to safely run as a worker under different configurations at every Grid node. Different workload balances were studied for the sake of performance, finding out a reasonable default behaviour without any configuration tweaking. Still the time of waiting for a working node limites improvements when making the jobs smaller. Therefore submitting jobs which require less than a few minutes to complete do not increase performance because Grid latencies start playing a significant role.

Future work includes fine-graining of data results through feedback into the master and standardization of porting of similar codes developed without parallel

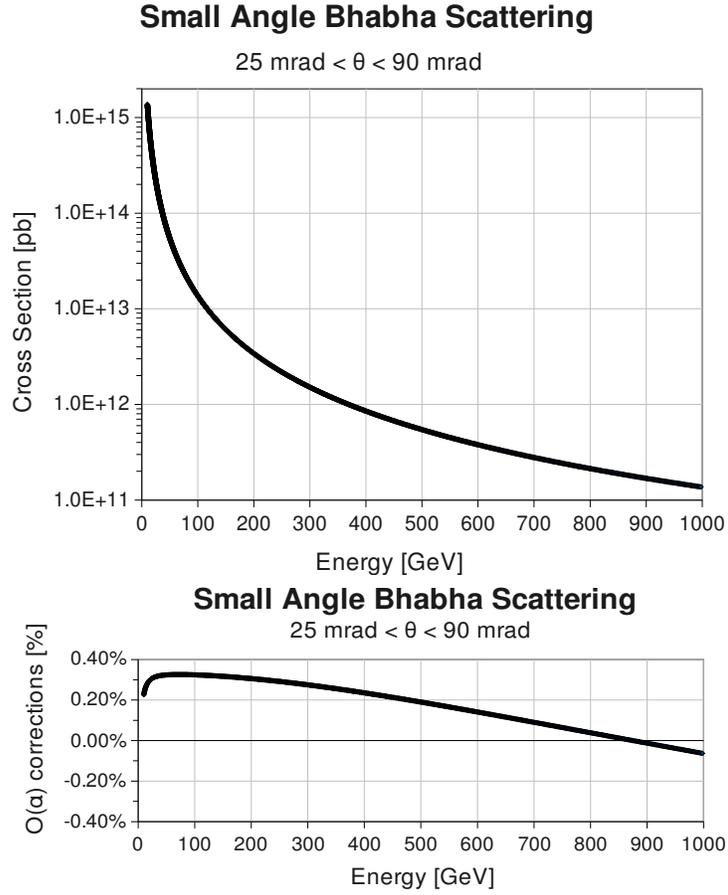


Fig. 9. Integrated cross section between $25 \text{ mrad} < \theta < 90 \text{ mrad}$ for Bhabha scattering (above). The percentage of the one-loop corrected cross section with respect to the tree-level is also shown (below), with a maximum of 0.33% at $E = m_Z$. The maximum of soft photon energy was taken to be $E_{\gamma_{\text{soft}}}^{\text{max}} = 0.2\sqrt{s}$.

execution in mind. An interesting possibility would be the implementation of more advanced job managers which dynamically obtain feedback from finished jobs and try strategies according to well defined policies. This could minimize waiting time, avoiding strictly failing nodes or avoiding resubmission of jobs being processed slowly.

Acknowledgments

The authors would like to thank the phenogrid VO for letting us access their resources for this study. We also welcome the openness of this UK/Ireland regional

declared VO to other European partners and users interested in the phenomenology of particle physics.

This work makes use of results produced with the EGEE grid infrastructure, co-funded by the European Commission (INFISO-RI-222667).

This research was supported by Consejería de Educación de la Comunidad de Madrid, Fondo Europeo de Desarrollo Regional (FEDER) and Fondo Social Europeo (FSE), through BIOGRIDNET Research Program S-0505/TIC/000101, by Ministerio de Educación y Ciencia, and through the research grant TIN2006-02806, and by the European Union through the research grant EGEE-III grant agreement 22667.

References

1. J. Brau *et al.* [ILC Collaboration], “ILC Reference Design Report Vol. 1: Executive Summary,” [arXiv:0712.1950](#).
2. A. Djouadi *et al.* [ILC Collaboration], “International Linear Collider Reference Design Report Vol. 2: Physics at the ILC,” [arXiv:0709.1893](#).
3. A. Lorca and T. Riemann, “An integrated tool for loop calculations: AITALC,” *Comput. Phys. Commun.* **174** (2006) 71. [arXiv:hep-ph/0412047](#).
4. R. P. Feynman, “Space-time approach to nonrelativistic quantum mechanics,” *Rev. Mod. Phys.* **20** (1948) 367.
5. F. A. Berends, K. J. F. Gaemers and R. Gastmans, “Hard photon corrections for Bhabha scattering,” *Nucl. Phys. B* **68** (1974) 541.
6. S. Jadach, E. Richter-Was, B. F. L. Ward and Z. Was, “Monte Carlo program BHLUMI-2.01 for Bhabha scattering at low angles with Yennie-Frautschi-Suura exponentiation,” *Comput. Phys. Commun.* **70** (1992) 305.
7. R. Bonciani, A. Ferroglia, P. Mastrolia, E. Remiddi and J. J. van der Bij, “Two-loop $N(F) = 1$ QED Bhabha scattering: Soft emission and numerical evaluation of the differential cross-section,” *Nucl. Phys. B* **716** (2005) 280. [arXiv:hep-ph/0411321](#).
8. M. Czakon, J. Gluza and T. Riemann, “Master integrals for massive two-loop Bhabha scattering in QED,” *Phys. Rev. D* **71** (2005) 073009. [arXiv:hep-ph/0412164](#).
9. A. A. Penin, “Two-loop corrections to Bhabha scattering,” *Phys. Rev. Lett.* **95** (2005) 010408. [arXiv:hep-ph/0501120](#).
10. J. A. M. Vermaseren, “New features of FORM,” [arXiv:math-ph/0010025](#).
11. M. Tentyukov and J. Fleischer, “A Feynman diagram analyser DIANA,” *Comput. Phys. Commun.* **132** (2000) 124. [arXiv:hep-ph/9904258](#).
12. T. Hahn and M. Perez-Victoria, “Automatized one-loop calculations in four and D dimensions,” *Comput. Phys. Commun.* **118**, 153 (1999). [arXiv:hep-ph/9807565](#).
13. E. Huedo, R.S. Montero and I.M. Llorente, “The GridWay Framework for Adaptive Scheduling and Execution on Grids,” *Scalable Computing - Practice and Experience* **6**, **3** (2005) 1. <http://gridway.org>
14. J.L. Vázquez-Poletti, E. Huedo, R.S. Montero and I.M. Llorente, “A Comparison Between two Grid Scheduling Philosophies: EGEE WMS and GridWay” *Multiagent and Grid Systems* **3**, **4** (2007) 429-439.
15. David Grellscheid. Private communication, 19/02/2009.