Electronic structure calculations are a widely used tool in materials science and large consumer of supercomputing resources. Traditionally, the software packages for these kind of simulations have been implemented in compiled languages, where Fortran in its different versions has been the most popular choice. While dynamic, interpreted languages, such as Python, can increase the efficiency of programmer, they cannot compete directly with the raw performance of compiled languages. However, by using an interpreted language together with a compiled language, it is possible to have most of the productivity enhancing features together with a good numerical performance. We have used this approach in implementing an electronic structure simulation software GPAW using the combination of Python and C programming languages. While the chosen approach works well in standard workstations and Unix environments, massively parallel supercomputing systems can present some challenges in porting, debugging and profiling the software. In this paper we describe some details of the implementation and discuss the advantages and challenges of the combined Python/C approach. We show that despite the challenges it is possible to obtain good numerical performance and good parallel scalability with Python based software.