**When to use Conda and when to use Pip?**

Both pip and Conda are tools for managing virtual environments, but they have some differences in terms of features and use cases. Here is a comparative analysis of pip and Conda:

* **Package management**: Pip is a package manager that installs packages from PyPI (Python Package Index), while Conda is a package manager that installs packages from multiple channels, including the official Anaconda repository, the conda-forge community repository, and the PyPI repository. Conda's ability to manage packages from multiple channels gives it an advantage over pip in terms of package availability.  
  *Due to larger set of packages conda is the winner.****Conda:1, pip:0***
* **Environment management**: Both pip and Conda allow you to create and manage isolated environments for your projects. However, Conda's environment management is more flexible than pip's, as it can handle dependencies for multiple languages and packages that are not available on PyPI.  
  *Due to flexibility conda is the winner.* ***Conda:2, pip:0***
* **Dependency management**: Conda has a powerful dependency solver that can automatically resolve complex dependency conflicts, while pip relies on the user to manage dependencies manually. This makes Conda more suitable for managing complex projects with many dependencies.  
  *Due to better dependancy management, conda is the winner.* ***Conda:3, pip:0***
* **Cross-platform compatibility**: Conda is designed to work on different operating systems, including Windows, macOS, and Linux, while pip is a Python-specific tool that may not work as well on other platforms.  
  *Due to better cross-platform performance, conda is the winner.* ***Conda:4, pip:0***
* **Ease of use**: Both pip and Conda are relatively easy to use, but Conda has a more user-friendly command-line interface and documentation, which can make it easier for beginners to get started.  
  *Due to similar commands and user picking pip as initial point despite good conda documentation, both are the winner*. ***Conda:5, pip:1***
* **Community support**: Pip has a larger community of users and developers, as it is the standard package manager for Python. This means that there are more resources available for troubleshooting and support. However, Conda also has a large and active community, particularly in the scientific computing and data science communities.
* *Bigger community is always a plus point, thus pip is the winner*. ***Conda:5, pip:2***

In summary, both pip and Conda are useful tools for managing virtual environments, but Conda's ability to manage packages from multiple channels, handle dependencies for multiple languages, and automatically resolve complex dependency conflicts make it a more versatile and powerful tool for managing dependencies and environments. However, pip may be more suitable for simpler projects that only require Python packages from PyPI.

**Final Conclusion**

* If you are building a simple project handled by very few to single person for a short span of time. It is good to go with simple pip.
* For Larger projects with many people working on it or one that uses different versions of dependencies, It is advisable to use conda over pip to make the process smoother.