Motivation

Problems with existing .tar.bz2 format:

1. Uncompressing .tar.bz2s is very slow. It's a legacy format at this stage.
2. Anaconda Installer unpacks something around 260 of these. More modern compression formats tend to be between 4 and 10x faster.
3. .tar.bz2 is not at the cutting edge of compression in terms of ratio. By utilizing modern compression algorithms, we can reduce our packages to sometimes as low as 60% smaller than the equivalent .tar.bz2 file.
4. We rely on unbzip2 to be installed on the system. Sometimes it is not (docker).
5. Metadata reading from packages currently requires complete extraction of a .tar.bz2 file. We could index packages and perform other metadata tasks much more quickly if we could read just the metadata without decompressing/extracting the whole package.
6. Package signing currently would require us to sign either all files in a package, or have a sidecar signature file shipped alongside .tar.bz2 files. It is desirable to ship a signature within a package, but to have that signature apply to an archive within the package.

Proposed implementation: the .conda format

Overall structure

Using the Debian .deb package as an example, our design uses an outer container with inner compressed archives. The outer container facilitates keeping everything together and abstracts the implementation details of the file format. A

Outer container

The outer format must be:

- extractable using standard tools that are present on every platform
- indexable, so that subsections of the file can be accessed and extracted quickly
- uncompressed, because the inner containers handle the compression of any real data

Zip file support is the most ubiquitous format that matches all of these criteria.

The outer container has minimal information in it:

- An API version, to be used by Conda, such that Conda can determine whether it is equipped to handle the contents of this package (generally, does conda have adequate libarchive support built in for the compression algorithm used in this package?) It is assumed that the ability to decompress and extract .conda files will be built into conda using a compiled language, such as Rust. Updating conda may thus confer new capabilities for what kinds of compression algorithms may be installed.

- Signatures (either as files, or as text in a JSON file) for the other archives contained in this outer container. By including these inside the outer container, they may be distributed internally rather than as sidecar files.

Inner metadata container

Metadata is separate from the other contents of the package so that it can be decompressed and extracted independently. Because the outer format is indexed, we can extract this inner container, decompress it, and read the metadata without decompressing the potentially larger contents of the actual package. The inner metadata package has a few expectations:

- It is a .tar file that has some compression algorithm applied to it
- Its extension reflects the compression algorithm that has been applied to it. For example, a bzip2-compressed archive would have the .tar.bz2 extension.
- It is not designed for random access. Instead, it is assumed that it will always be decompressed and extracted as a whole.

For ideological purity, the extracted metadata files here do not share any subfolder with the contents of the inner package container. For example, you would not bundle info/index.json here, and info/paths.json in the inner package container.

Inner package container

The inner package container has a few expectations:

- It is a .tar file that has some compression algorithm applied to it
- Its extension reflects the compression algorithm that has been applied to it. For example, a bzip2-compressed archive would have the .tar.bz2 extension.
- It is not designed for random access. Instead, it is assumed that it will always be decompressed and extracted as a whole.

Migration strategy:

Conda 4.7.0 will be the target version for release of the .conda file format. To aid in adoption and reduce necessary work, a tool will be created that allows conversion from the .conda format to the .tar.bz2 format. By convention, a package named "something-1.0-0.conda" will install the exact same contents as a package named "something-1.0-0.tar.bz2."

Anaconda will produce .tar.bz2 packages until the conda 4.6.x release series is no longer supported for enterprise customers. Conda-build will drop official support for building .tar.bz2 files when Anaconda no longer produces .tar.bz2 files. Conda will maintain the ability to consume and install .tar.bz2 files in perpetuity.

Anaconda.org will need to support the new format. It is not decided here whether that will be done via modification to the AEv4 codebase, or whether it will need to wait for some future AE platform to be used for anaconda.org.

Technical references

- Debian package file format: https://en.wikipedia.org/wiki/Deb_(file_format)
- Considered archive formats for inner/outer containers: https://blog.sourced.tech/post/siva/