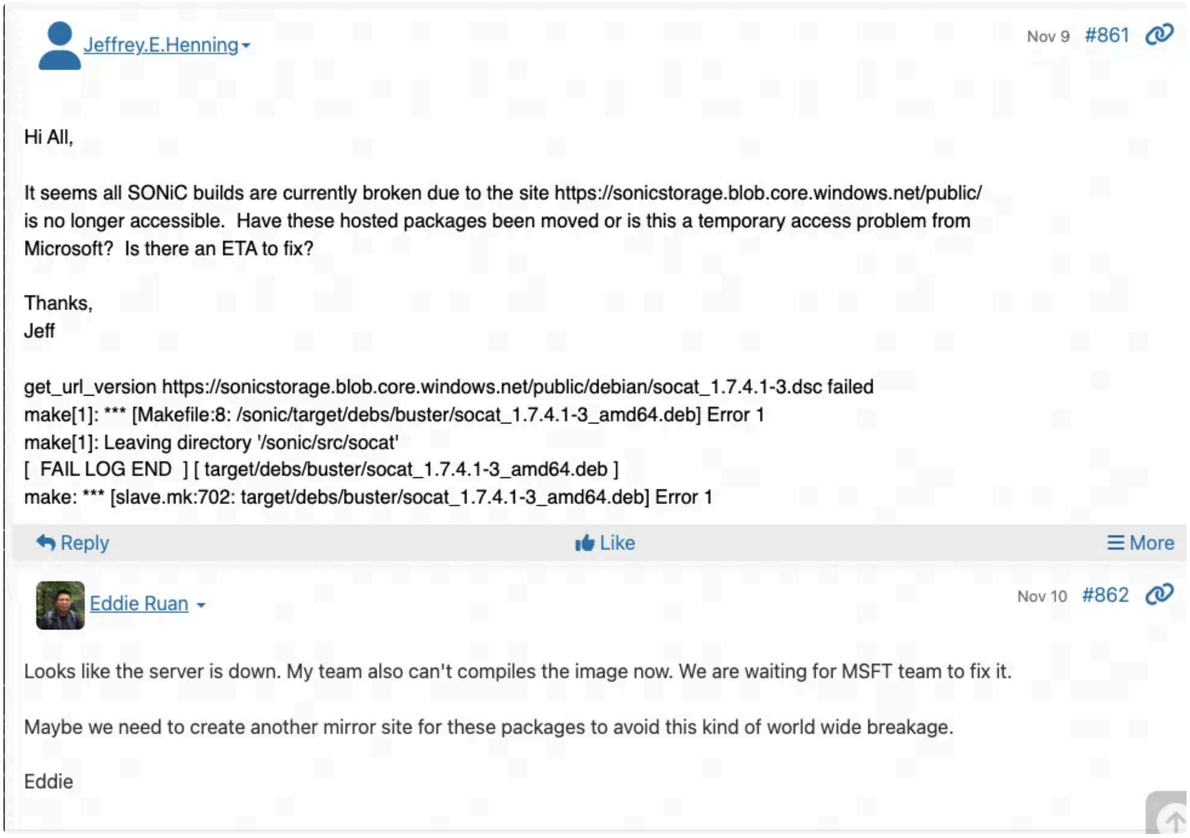
Enhancements on SONiC Compiling

Problems to be solved

A few issues have been identified in the SONiC compilation process:

Problem 1: A recent **sonic-storage** failure on 11/9/2024 caused widespread SONiC compilation failures globally. This issue occurred because certain dependent download URLs lack backup mirrors. If a URL becomes inaccessible, the compilation process fails immediately.



Jeffrey.E.Henning Nov 9 #861

Hi All,

It seems all SONiC builds are currently broken due to the site https://sonicstorage.blob.core.windows.net/public/ is no longer accessible. Have these hosted packages been moved or is this a temporary access problem from Microsoft? Is there an ETA to fix?

Thanks,
Jeff

```
get_url_version https://sonicstorage.blob.core.windows.net/public/debian/socat_1.7.4.1-3.dsc failed
make[1]: *** [Makefile:8: /sonic/target/debs/buster/socat_1.7.4.1-3_amd64.deb] Error 1
make[1]: Leaving directory '/sonic/src/socat'
[  FAIL LOG END  ] [ target/debs/buster/socat_1.7.4.1-3_amd64.deb ]
make: *** [slave.mk:702: target/debs/buster/socat_1.7.4.1-3_amd64.deb] Error 1
```

Eddie Ruan Nov 10 #862

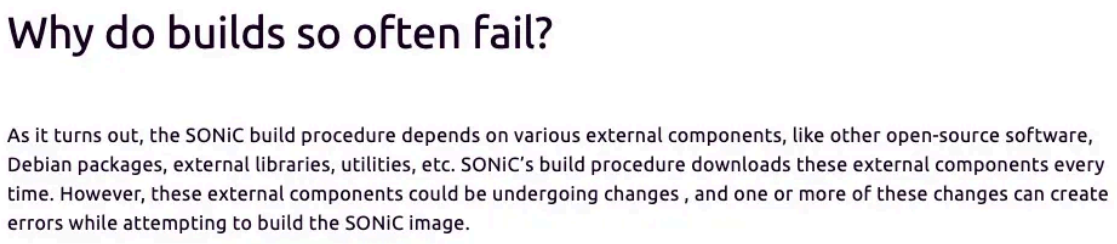
Looks like the server is down. My team also can't compiles the image now. We are waiting for MSFT team to fix it.

Maybe we need to create another mirror site for these packages to avoid this kind of world wide breakage.

Eddie

Problem 2: Some dependency packages do not have specified versions. Changes to these underlying dependencies can lead to failures in SONiC compilation, particularly in release builds. This issue can be addressed by locking the versions of all dependency packages and implementing a mechanism to use mirror sited. The following URL provides some insight information on this problem.

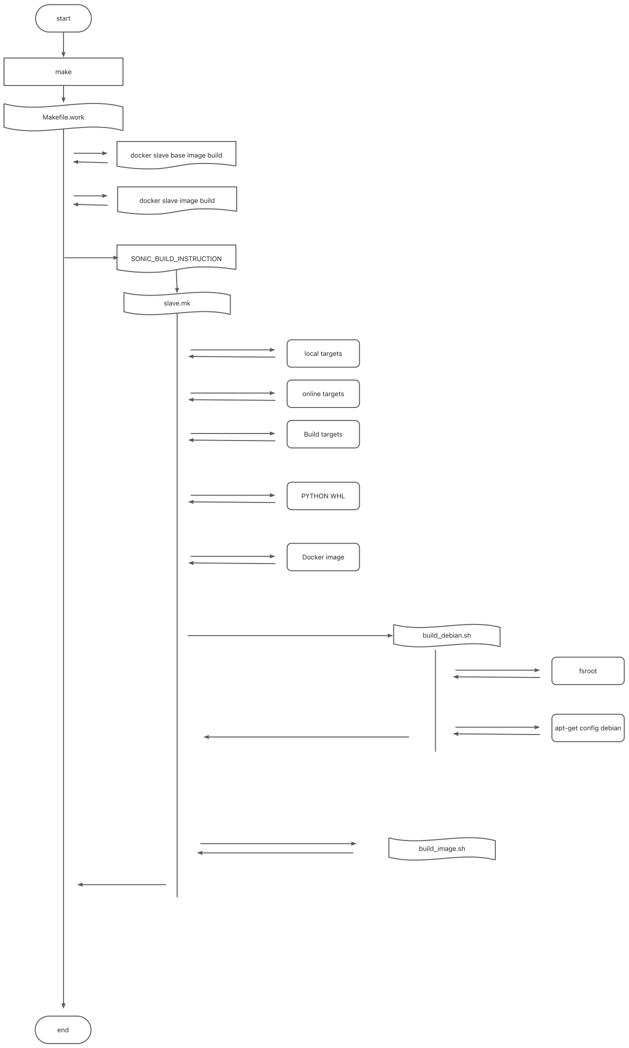
https://www.capgemini.com/insights/expert-perspectives/sonic-nos-production-builds/ <https://www.capgemini.com/insights/expert-perspectives/sonic-nos-production-builds/>



Why do builds so often fail?

As it turns out, the SONiC build procedure depends on various external components, like other open-source software, Debian packages, external libraries, utilities, etc. SONiC's build procedure downloads these external components every time. However, these external components could be undergoing changes , and one or more of these changes can create errors while attempting to build the SONiC image.

Current SONiC Compile Work flow



External Dependency Analysis

1. Use docker pull to get docker base images

   https://github.com/sonic-net/sonic-buildimage/blob/master/dockers/docker-base-buster/Dockerfile.j2#L8 <https://github.com/sonic-net/sonic-buildimage/blob/master/dockers/docker-base-buster/Dockerfile.j2#L8>

Bash 复制代码

```
FROM {{ prefix }}{{DOCKER_BASE_ARCH}}/debian:buster
```

REGISTRY_SERVER could be used to provide an alternative docker image hub location.

Need to

2. Use apt-get to get debian packages；

We have a lot of place to use apt-get to get and install debian packages.

CMake 复制代码

```
echo "deb-src [arch=amd64] http://sonic-debian-archive.trafficmanager.net/debian/ buster main contrib non-free" >> /
```

TODO:

1. Need to check if we need to pin down each debian needed packages' version.
2. Need to find add some mirror site apt server to apt source list.

3. Use wget for download binaries directly ；

wget is a widely used method to download some binaries directly. The occurrance of problem 1 is due to the storage change mistake in https://sonicstorage.blob.core.windows.net/ <https://sonicstorage.blob.core.windows.net/> . There are multiple sources of wget.

CMake 复制代码

```
wget https://archive.debian.org/debian-archive/debian/pool/main/g/golang-glide/golang-glide_0.12.3-2+b3_amd64.deb
```

Dockerfile 复制代码

```
RUN apt-get update && apt-get install -y eatmydata && eatmydata apt-get install -y \
        apt-utils \
        default-jre-headless \
        openssh-server \
        curl \
        wget \
        unzip \
        {{ GZ_COMPRESS_PROGRAM }} \
        git \
        build-essential \
        libtool \
        lintian \
        sudo \
        dh-make \
        dh-exec \
        kmod \
        libtinyxml2-dev \
```

TODO:

1. How to provide alternative mirror sites? Can we use centralized location to maintain all needed binaries getting from via wget.
2. Need to make sure the download binaries versions are pinned down.

4. Install python packages via pip；

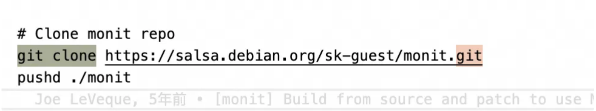
We use "pip3 install" to install some python packages during build via pip3

TODO

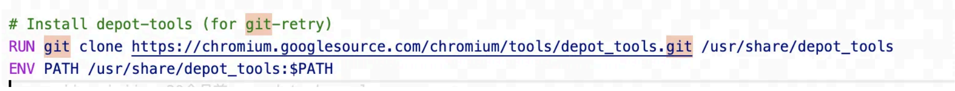
Need to create pip source mirror server for backup and allow pip3 install use "-i" to install packages from a specified source server.

5. Use git clone to pull source codes.

This git clone is not for pulling SONiC submodule's codes, but pulling other sources utililiy codes.



```
# Clone monit repo
git clone https://salsa.debian.org/sk-guest/monit.git
pushd ./monit
   Joe LeVeque, 5年前 • [monit] Build from source and patch to use M
```



```
# Install depot-tools (for git-retry)
RUN git clone https://chromium.googlesource.com/chromium/tools/depot_tools.git /usr/share/depot_tools
ENV PATH /usr/share/depot_tools:$PATH
```

TODO:

1. Need to specify versions for these downloading source codes
2. Need to create a mirror server for hosting these source codes.

6. External dependencies pulled during the compilation of Go projects:

Need to make sure all Go packages have version specified.

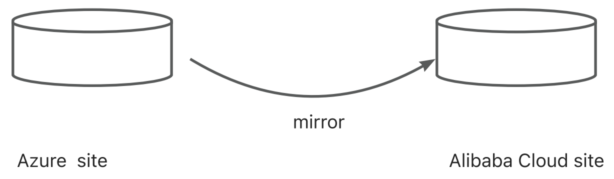
```
import "your-server.com/your-module/your-package"
```

One way to create local cache is the following

1. **First, cache the Go packages required during the compilation process locally.**
2. **Set up a local repository for the Go packages needed in SONiC's compilation process.**
3. **Modify the `import` statements in the Go project to fetch the dependency packages from a specified URL.**

Proposal

1. Create various mirror sites in Alibaba cloud to back up SONiC compiling needed external dependencies.



2. Allow users to specify their external downloading sources from either sites as their primary source
3. Allow hot backup to switch to backup source if the primary source is down.

High Level Approach

1. Need to decide overall build strategy on how to set up versioning on all external dependencies and which parts we would like to create local cache.
2. Create a mirror center in Alibaba Cloud for SONiC compilation.
    a. Some debian packages python packages have already had mirror site in Alibaba cloud.
    b. Need to create mirror for some SONiC specific sources from Azure to Aliaba Cloud, including some local caching.
3. Update sonic-buildimage to allow user to specify their external downloading sources.
4. All host switch over if the primary source is down.

638字