Item 285
git_comments:

git_commits:

1. **summary:** Do not provide a stale negative cache (#7422)
   message: Do not provide a stale negative cache (#7422)

github_issues:

1. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
2. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
3. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-

age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier

4. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
5. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each

request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier

6. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
7. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
8. **title:** stale negative cache is not updated by fresh negative response
   body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-

age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
label: code-design

9. **title:** stale negative cache is not updated by fresh negative response
body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier
10. **title:** stale negative cache is not updated by fresh negative response
body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated

maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier

11. **title:** stale negative cache is not updated by fresh negative response
    body: Hello, ATS does not update existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. ``` CONFIG proxy.config.http.negative_revalidating_enabled INT 1 CONFIG proxy.config.http.negative_revalidating_lifetime INT 3600 ``` If the backend replies 502 result having `Cache-Control: max-age=60, public`, it is well cached in ATS, that is good. When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : - The client cannot get a newer 'negative' response. - Once the negative cache becomes stale, all incoming requests will involve requests to the backend. It is of course good 'not updating positive stale cache by negative fresh response', but there is no reason 'not to update negative stale cache by negative fresh response'. [background] My idea is switching between 'normal' backend and 'maintenance mode' backend using HAProxy behind ATS. Usually normal backend replies 200 status with `Cache-Control: max-age=300, public`. When we switch to the 'maintenance mode' backend, it replies a beautiful maintenance information page having 502 status with `Cache-Control: max-age=60, public` for whatever HTML requests. By using this short max-age cache, we would like to avoid heavy load on the maintenance mode backend. Thanks to 502 status of 'maintenance mode' backend, ATS tries to return a stale cache of 'positive' response, if exists, until long-enough `negative_revalidating_lifetime` reaches, i.e. we can provide a normal site as much as possible. And when `negative_revalidating_lifetime` reaches, or positive cache does not exist (for explicit `no-store` requests etc.), the client will get a beautiful maintenance information page. But because of the behaviour describing here, the client cannot get the updated maitenance information page during maintenance mode and also the maintenance mode backend needs to respond for each request in vain. Here is an example timeline. ``` 00:00 ATS fetches 200 response from the normal backend and replies it 00:01 (we switch to the maintenance mode backend) 00:03 ATS replies a normal cache 00:10 cache is already stale, but ATS still replies a normal cache thanks to negative_revalidating_lifetime 01:15 ATS fetches from the maintenance mode backend, then caches it and replies it 01:16 (we update the maintenance page on the maintenance backend) 01:17 'negative' cache is already stale, thus ATS fetches again from the maintenance mode backend, but replies the old existing stale 'negative' cache without updating it 01:20 'negative' cache is always stale, thus ATS fetches again from the maintenance mode backend. ``` If ATS supports `stale-if-error`, our responses would be: - from 'normal' backend: `Cache-Control: max-age=...,stale-if-error=<7 days>` - from 'maintenance' backend: `Cache-Control: max-age: 60` (without `stale-if-error`) but it is not supported for now, thus we are trying this approach with using `negative_revalidating_lifetime`. I verified this behaviour on ATS version 7.1.1 and 8.0.2. Thanks in advance ! Kazuhiko /cc @vpelletier

github_issues_comments:

1. I would like to propose the following patch that perfectly solves this issue. ``` --- trafficserver-8.1.1/proxy/http/HttpTransact.cc.orig 2020-12-01 00:30:26.000000000 +0100 +++ trafficserver-8.1.1/proxy/http/HttpTransact.cc 2021-01-11 11:35:41.946893735 +0100 @@ -5752,6 +5752,7 @@ HttpTransact::is_stale_cache_response_returnable(State *s) { HTTPHdr *cached_response = s->cache_info.object_read->response_get(); + HTTPStatus cache_response_code = cached_response->status_get(); // First check if client allows cached response // Note does_client_permit_lookup was set to @@ -5760,6 +5761,12 @@ if (!s->cache_info.directives.does_client_permit_lookup) { return false; } + // We don't serve stale negative cache. + if (cache_response_code == HTTP_STATUS_INTERNAL_SERVER_ERROR || cache_response_code == HTTP_STATUS_GATEWAY_TIMEOUT || + cache_response_code == HTTP_STATUS_BAD_GATEWAY || cache_response_code == HTTP_STATUS_SERVICE_UNAVAILABLE) { + TxnDebug("http_trans", "[is_stale_cache_response_returnable] stale negative cache"); + return false; + } // Spec says that we can not serve a stale document with a // "must-revalidate header" // How about "s-maxage" and "no-cache" directives? ```
2. I think this was fixed in #7401 ? If so, we should close this.
3. > I think this was fixed in #7401 ? If so, we should close this. Thanks for looking into this. I believe, however, that the issue described here is orthogonal to #7401. #7401 fixed an issue that was introduced recently (a few weeks ago) via #7361 , but the issue described here is seen in 8.0.2 (per Kazuhiko's previous comment): > I verified this behaviour on ATS version 7.1.1 and 8.0.2. 8.0.2 is a couple years old: https://github.com/apache/trafficserver/releases/tag/8.0.2
4. Thanks for comments. I just tried ATS 8.1.1 and confirmed that the issue still existed but could be fixed by my patch above.
5. @fdiary could you open a PR with your patch on the master branch?
6. @masaori335 Sure. I just opened a PR : #7422
7. Thanks for the PR. Let me clarify a bit. > When it becomes stale, i.e. more than 60 seconds later, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. Is this because `proxy.config.http.negative_revalidating_lifetime` is not used at `is_stale_cache_response_returnable()` as you pointed out #7425? I tried to repro the scenario. And seems like the stale negative response is updated if `proxy.config.http.cache.max_stale_age` is short enough.
8. **body:** > And seems like the stale negative response is updated if proxy.config.http.cache.max_stale_age is short enough. Indeed, `is_stale_cache_response_returnable` refers `max_stale_age` but not `negative_revalidating_lifetime`. But if we have short `max_stale_age`, good 'positive' cache will be also unavailable shortly, that is not good. So regardless which value

`is_stale_cache_response_returnable` refers, I believe that the proposal here is still meaningful to fix this issue, i.e. to provide a good 'positive' cache for long period but to make updating 'negative' cache possible and also to reduce requests to the 'negative response' backend.
label: code-design

9. I agree. It'd be useful if we can change the period of serving stale contents by the cached status code.

github_pulls:

1. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
2. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
3. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
4. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
5. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
6. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
7. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
8. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
9. **title:** Do not provide a stale negative cache
   body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend.

With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417

10. **title:** Do not provide a stale negative cache
    body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417
11. **title:** Do not provide a stale negative cache
    body: ATS does not update an existing cache of 'negative' response even if ATS fetches a fresh equally-negative response from the backend. When a negative cache becomes stale, ATS fetchs a new version from the backend, which, if it is equally negative to the cached version, will be discarded and the stale entry will be sent to the client. That means : * The client cannot get a newer 'negative' response. * Once the negative cache becomes stale, all incoming requests will involve requests to the backend. With this PR, ATS will no longer provide a stale negative cache if the revalidated response is also negative, so that we can fix these issues still providing a stale 'positive' cache if available. This closes #7417

github_pulls_comments:

1. @fdiary can you run `make clang-format` and amend your commit with the changed files?
2. @randall : "can you run `make clang-format` and amend your commit with the changed files?" Oops, I just did so.
3. @masaori335 Thanks !
4. Cherry-picked to v9.0.x branch.
5. Cherry-picked to v9.1.x branch.
6. Can we get a backport PR for 8.1.x for this?

github_pulls_reviews:

1. This might have a pretty big impact in other places where calls this function. @sudheerv Is this change safe with `CACHE_WL_FAIL_ACTION_STALE_ON_REVALIDATE`? https://github.com/apache/trafficserver/blob/690ec54950611a1955b34911b73a222365998667/proxy/http/HttpTransact.cc#L7215-L7216
2. @fdiary How about move this change under the if-block of doing negative revalidating to minimize side effects? https://github.com/apache/trafficserver/blob/b6b61fbd3e232dbef010dc8b712bb2076510bd7f/proxy/http/HttpTransact.cc#L4347 My concern is this changes behavior of setting `2` and `3` of `proxy.config.http.cache.open_write_fail_action`[*1]. It's a config to do stale-while-revalidate, so if we never serve negative stale cache, all simultaneous requests might go to the origin server. [*1] https://docs.trafficserver.apache.org/en/latest/admin-guide/files/records.config.en.html#proxy-config-http-cache-open-write-fail-action
3. @masaori335 Thanks for your proposal, that is reasonable. I force-updated the change in this PR.

jira_issues:

jira_issues_comments: