SQUEEZED: a host memory ballooning daemon

Version: Document Revision 0.1
Date: 9th November 2009
Comments are welcome!

David Scott: dave.scott@eu.citrix.com

1 Introduction

We wish to:

1. allow VM memory to be adjusted dynamically without having to reboot; and
2. "squeeze" a few more VMs onto a host to cover the interval between another host failing and more capacity being brought online.

SQUEEZED is a per-host memory ballooning daemon. It performs two tasks:

1. it exports a simple host memory management interface to the XAPI toolstack through which XAPI can *reserve* memory for new domains;
2. it applies a *ballooning policy* to all domains running on a host.

The daemon currently includes a simple ballooning policy (see Section 4) and the intention is that this can be replaced later with more sophisticated policies (e.g. *xenballoond*[1])). Although the only client is the XAPI toolstack, the interface can in theory be used by other clients.

In the short-term this document will allow the assumptions and the design to be reviewed. In the longer term this document will become part of the XAPI toolstack software design notes.

The rest of this document is structured as follows. Section 2 lists assumptions made by the ballooning daemon on other parts of the system; these assumptions need careful review and may not be valid. Section 3 describes the interface between the toolstack and the ballooning daemon. Section 4 describes the simple built-in ballooning policy and Section 5 describes how SQUEEZED models memory. The main loop of the daemon is described in Section 6 and a detailed example is described in Section 7. Section 8 describes the structure of the daemon itself and finally Section 9 lists some known issues.

2 Environmental assumptions

1. The SQUEEZED daemon runs within a XEN domain 0 and communicates to xenstored via a Unix domain socket. Therefore SQUEEZED is granted full access to xenstore, enabling it to modify every domain's `memory/target`.
2. The SQUEEZED daemon calls `setmaxmem` in order to cap the amount of memory a domain can use. This relies on a patch to xen[2] which allows `maxmem` to be set lower than `totpages`. See Section 6.2 for more information.
3. The SQUEEZED daemon assumes that only domains which write `control/feature-balloon` into xenstore can respond to ballooning requests. It will not ask any other domains to balloon.
4. The SQUEEZED daemon assumes that the memory used by a domain is: (i) that listed in `domain_getinfo` as `totpages`; (ii) shadow as given by `shadow_allocation_get`; and (iii) a small (few KiB) of miscellaneous XEN structures (e.g. for domains, vcpus) which are invisible.
5. The SQUEEZED daemon assumes that a domain which is created with a particular `memory/target` (and `startmem`, to within rounding error) will reach a stable value of `totpages` before writing `control/feature-balloon`.[3]The daemon writes this value to `memory/memory-offset` for future reference.

[1] `http://wiki.xensource.com/xenwiki/Open_Topics_For_Discussion?action=AttachFile&do=get&target=Memory+Overcommit.pdf`

[2] `http://xenbits.xen.org/xapi/xen-3.4.pq.hg?file/c01d38e7092a/max-pages-below-tot-pages`

[3] The `control/feature-balloon` key is probably the wrong signal.

- The SQUEEZED daemon does not know or care exactly what causes the difference between `totpages` and `memory/target` and it does *not* expect it to remain constant across XEN releases. It only expects the value to remain constant over the lifetime of a domain.

6. The SQUEEZED daemon assumes that the balloon driver has hit its target when difference between `memory/target` and `totpages` equals the `memory-offset` value.

   - Corrollary: to make a domain with a responsive balloon driver currenty using `totpages` allocate or free x, it suffices to set `memory/target` to x + `totpages` + `memory-offset` and wait for the balloon driver to finish. See Section 5 for more detail.

7. The SQUEEZED daemon must maintain a "slush fund" of memory (currently 9MiB) which it must prevent any domain from allocating. Since (i) some XEN operations (such as domain creation) require memory within a physical address range (e.g. $<$ 4GiB) and (ii) since XEN preferentially allocates memory outside these ranges, it follows that by preventing guests from allocating *all* host memory (even transiently) we guarantee that memory from within these special ranges is always available. See Section 6.1 for more details.

8. The SQUEEZED daemon assumes that it may set `memory/target` to any value within range: `memory/dynamic-max` to `memory/dynamic-min`

9. The SQUEEZED daemon assumes that the probability of a domain booting successfully may be increased by setting `memory/target` closer to `memory/static-max`.

10. The SQUEEZED daemon assumes that, if a balloon driver has not made any visible progress after 5 seconds, it is effectively *inactive*. Active domains will be expected to pick up the slack.

3 Toolstack interface

This section begins by describing the concept of a *reservation* and then describes the toolstack interface in pseudocode.

A *reservation* is: an amount of host free memory tagged with an associated *reservation id*. Note this is an internal SQUEEZED concept and XEN is completely unaware of it. When the daemon is moving memory between domains, it always aims to keep

$$host\ free\ memory >= s + \sum_i reservation_i$$

where s is the size of the "slush fund" (currently 9MiB) and $reservation_i$ is the amount corresponding to the ith reservation.

As an aside: Earlier versions of SQUEEZED always associated memory with a XEN domain. Unfortunately this required domains to be created before memory was freed which was problematic because domain creation requires small amounts of contiguous frames. Rather than implement some form of memory defragmentation, SQUEEZED and XAPI were modified to free memory before creating a domain. This necessitated making memory *reservations* first-class stand-alone entities.

Once a *reservation* is made (and the corresponding memory is freed), it can be *transferred* to a domain created by a toolstack. This associates the *reservation* with that domain so that, if the domain is destroyed, the *reservation* is also freed. Note that SQUEEZED is careful not to count both a domain's *reservation* and its `totpages` during e.g. domain building: instead it considers the domain's allocation to be the maximum of *reservation* and `totpages`.

The size of a *reservation* may either be specified exactly by the caller or the caller may provide a memory range. If a range is provided the daemon will allocate at least as much as the minimum value provided and as much as possible up to the maximum. By allocating as much memory as possible to the domain, the probability of a successful boot is increased.

Clients of the SQUEEZED provide a string name when they log in. All untransferred reservations made by a client are automatically deleted when a client logs in. This prevents memory leaks where a client crashes and loses track of its own reservation ids.

The interface looks like this:

```
string session_id login(string client_name)

string reservation_id reserve_memory(string client_name, int kib)
int amount, string reservation_id reserve_memory_range(string client_name, int min, int max)

void delete_reservation(string client_name, string reservation_id)

void transfer_reservation_to_domain(string client_name, string reservation_id, int domid)
```

The XAPI toolstack has code like the following: (in `http://www.xen.org/files/XenCloud/ocamldoc/index.html?c=xapi&m=Vmops`)

```
r_id = reserve_memory_range("xapi", min, max);
try:
   d = domain_create()
   transfer_reservation_to_domain("xapi", r_id, d)
with:
   delete_reservation("xapi", r_id)
```

The interface is currently implemented using a trivial RPC protocol over xenstore where requests and responses are directories and their parameters and return values are keys in those directories.

4 Ballooning policy

This section describes the very simple default policy currently built-into SQUEEZED.

Every domain has a pair of values written into xenstore: `memory/dynamic-min` and `memory/dynamic-max` with the following meanings:

`memory/dynamic-min` : the lowest value that SQUEEZED is allowed to set `memory/target`. The administrator should make this as low as possible but high enough to ensure that the applications inside the domain actually work.

`memory/dynamic-max` : the highest value that SQUEEZED is allowed to set `memory/target`. This can be used to dynamically cap the amount of memory a domain can use.

If all balloon drivers are responsive then SQUEEZED daemon allocates memory proportionally, so that each domain has the same value of:

$$\frac{\texttt{memory/target} - \texttt{memory/dynamic-min}}{\texttt{memory/dynamic-max} - \texttt{memory/dynamic-min}}$$

So:

- if memory is plentiful then all domains will have `memory/target` = `memory/dynamic-max`
- if memory is scarce then all domains will have `memory/target` = `memory/dynamic-min`

Note that the values of `memory/target` suggested by the policy are ideal values. In many real-life situations (e.g. when a balloon driver fails to make progress and is declared *inactive*) the `memory/target` values will be different.

Note that, by default, domain 0 has `dynamic_min` = `dynamic_max`, effectively disabling ballooning. Clearly a more sophisticated policy would be required here since ballooning down domain 0 as extra domains are started would be... problematic.

5 The memory model used by squeezed

This section describes the model used internally by SQUEEZED.

The SQUEEZED daemon considers a ballooning-aware domain (i.e. one which has written the `feature-balloon` flag into xenstore) to be a 6-tuple:

$$\textit{ballooning domain} = (\texttt{dynamic-min}, \texttt{dynamic-max}, \texttt{target}, \texttt{totpages}, \texttt{memory-offset}, \texttt{maxmem})$$

where

`dynamic-min` : policy value written to `memory/dynamic-min` in xenstore by a toolstack (see Section 4)

`dynamic-max` : policy value written to `memory/dynamic-max` in xenstore by a toolstack (see Section 4)

`target` : balloon driver target written to `memory/target` in xenstore by SQUEEZED

`totpages` : instantaneous number of pages used by the domain as returned by `domain_getinfo`

`memory-offset` : constant difference between `target` and `totpages` when the balloon driver believes no ballooning is necessary:

$$\texttt{memory-offset} \stackrel{def}{=} \texttt{totpages} - \texttt{target} \textit{ when idle}$$

`maxmem` : upper limit on `totpages`:

$$\texttt{totpages} <= \texttt{maxmem}$$

For convenience we define a `totpages'` to be the target value necessary to cause a domain currently using `totpages` to maintain this value indefinitely.

$$\texttt{totpages'} \stackrel{def}{=} \texttt{totpages} - \texttt{memory-offset}$$

The SQUEEZED daemon believes that:

- a domain should be ballooning iff `totpages' <> target` (unless it has become *inactive*);
- a domain has hit its target iff `totpages' = target` (to within 1 page);
- if a domain has `target` $\leftarrow x$ then, when ballooning is complete, it will have `totpages` = `memory-offset` $+ x$; and therefore
- to cause a domain to free y it sufficies to set `target` $\leftarrow$ `totpages` $-$ `memory-offset` $- y$.

The SQUEEZED daemon considers non-ballooning aware domains (i.e. those which have not written `feature-balloon`) to be represented by pairs of:

$$\textit{other domain} = (\texttt{totpages}, \textit{reservation})$$

where

`totpages` : instantaneous number of pages used by the domain as returned by `domain_getinfo`

reservation : memory initially freed for this domain by SQUEEZED after a `transfer_reservation_to_domid` call

Note that non-ballooning aware domains will always have `startmem` = `target` since the domain will not be instructed to balloon. Since a domain which is being built will have 0 <= `totpages` <= *reservation*, SQUEEZED computes:

$$unused(i) \stackrel{def}{=} i.reservation - i.\texttt{totpages}$$

and subtracts this from its model of the host's free memory, ensuring that it doesn't accidentally reallocate this memory for some other purpose.

The SQUEEZED daemon believes that:

- all guest domains start out as non-ballooning aware domains where `target` = *reservation* = `startmem`;
- some guest domains become ballooning-aware during their boot sequence i.e. when they write `feature-balloon`

The SQUEEZED daemon considers a host to be a 5-tuple:

$$host = (ballooning\ domains, other\ domains, s, \texttt{physinfo.free\_pages}, reservation_i)$$

where

ballooning domains : a list of *ballooning domain* values representing domains which SQUEEZED will instruct to balloon;

other domains : a list of *other domain* values which includes both domains which are still booting and will transform into *ballooning domains* and those which have no balloon drivers.

s : a "slush fund" of low memory required for XEN;

`physinfo.free_pages` : total amount of memory instantanously free (including both `free_pages` and `scrub_pages`)

$reservation_i$: a set of memory *reservations* not allocated to any domain

The SQUEEZED daemon considers memory to be unused (i.e. not allocated for any useful purpose) as follows:

$$unused\ memory = \texttt{physinfo.free\_pages} - \Sigma_i reservation_i - s - \Sigma_{i \in other\ domains} unused(i)$$

6 The main loop

The main loop [4] is triggered by either:

1. the arrival of an allocation request on the toolstack interface; or
2. the policy engine – polled every 10s – deciding that a target adjustment is needed.

Each iteration of the main loop[5] generates the following actions:

1. Domains which were active but have failed to make progress towards their target in 5s are declared *inactive*. These domains then have:

$$\texttt{maxmem} \leftarrow min(\texttt{target}, \texttt{totpages})$$

[4] `change_host_free_memory` in `http://xenbits.xen.org/xapi/xen-api.hg?file/3e8c0167940d/ocaml/xenops/squeeze.ml`

[5] `one_iteration` in `http://xenbits.xen.org/xapi/xen-api.hg?file/3e8c0167940d/ocaml/xenops/squeeze.ml`

2. Domains which were inactive but have started to make progress towards their target are declared *active*. These domains then have:

$$\texttt{maxmem} \leftarrow \texttt{target}$$

3. Domains which are currently active have new targets computed according to the policy (see Section 4). Note that inactive domains are ignored and not expected to balloon.

Note that domains remain classified as *inactive* only during one run of the main loop. Once the loop has terminated all domains are optimistically assumed to be *active* again. Therefore should a domain be classified as *inactive* once, it will get many later chances to respond.

See Section 6.1 for more detail on how targets are updated and Section 6.2 for more detail about `maxmem`.

The main loop has a notion of a host free memory "target", similar to the existing domain memory `target`. When we are trying to free memory (e.g. for starting a new VM), the host free memory "target" is increased. When we are trying to distribute memory among guests (e.g. after a domain has shutdown and freed lots of memory), the host free memory "target" is low. Note the host free memory "target" is always at least several MiB to ensure that some host free memory with physical address $<$ 4GiB is free (see Section 6.1 for related information).

The main loop terminates when all *active* domains have reached their targets (this could be because all domains responded or because they all wedged and became inactive); and the policy function hasn't suggested any new target changes. There are three possible results:

1. Success if the host free memory is near enough its "target";
2. Failure if the operation is simply impossible within the policy limits (i.e. `dynamic_min` values are too high;
3. Failure if the operation failed because one or more domains became *inactive* and this prevented us from reaching our host free memory "target".

Note that, since only *active* domains have their targets set, the system effectively rewards domains which refuse to free memory (*inactive*) and punishes those which do free memory (*active*). This effect is countered by signalling to the admin which domains/VMs aren't responding so they can take corrective action. To achieve this, the daemon monitors the list of *inactive* domains and if a domain is *inactive* for more than 20s it writes a flag into xenstore `memory/uncooperative`. This key is seen by the XAPI toolstack which currently generates an alert to inform the admin.

6.1 Two-phase target setting

Consider the scenario shown graphically in Figure 1. In the initial state (at the top of the diagram), there are two domains, one which has been requested to use more memory and the other requested to use less memory. In effect the memory is to be transferred from one domain to the other. In the final state (at the bottom of the diagram), both domains have reached their respective targets, the memory has been transferred and the host free memory is at the same value it was initially. However the system will not move atomically from the initial state to the final: there are a number of possible transient in-between states, two of which have been drawn in the middle of the diagram. In the left-most transient state the domain which was asked to *free* memory has freed all the memory requested: this is reflected in the large amount of host memory free. In the right-most transient state the domain which was asked to *allocate* memory has allocated all the memory requested: now the host's free memory has hit zero.

If the host's free memory hits zero then Xen has been forced to give all memory to guests, including memory $<$ 4GiB which is critical for allocating certain structures. Even if we ask a domain to free memory via the balloon driver there is no guarantee that it will free the *useful* memory. This leads to an annoying failure mode where operations such as creating a domain free due to `ENOMEM` despite the fact that there is apparently lots of memory free.

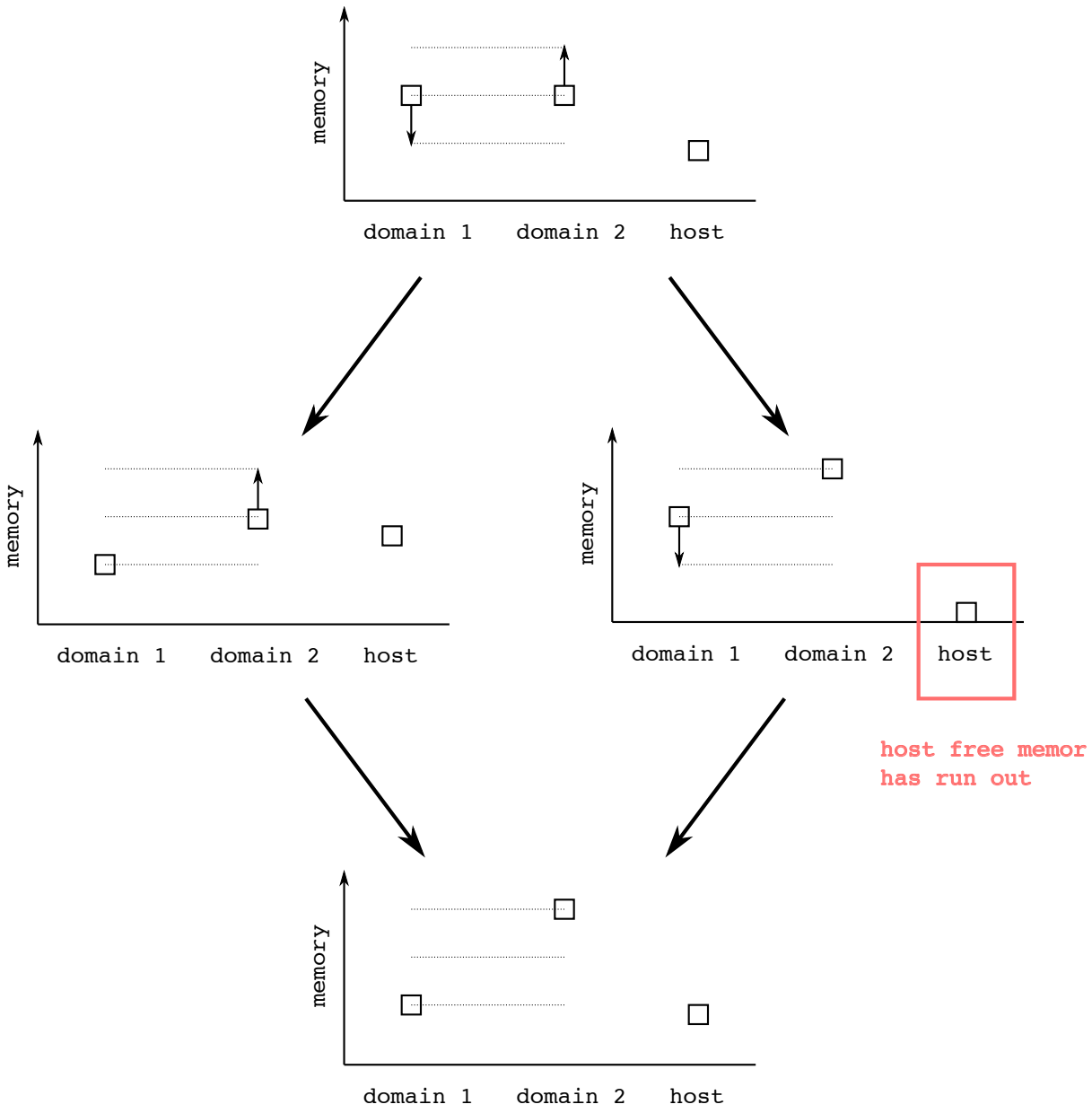

Figure 1: The diagram shows how a system with two domains can evolve if domain `memory/target` values are increased for some domains and decreased for others, at the same time. Each graph shows two domains (domain 1 and domain 2) and a host. For a domain, the square box shows its `totpages'` and the arrow indicates the direction of the `memory/target`. For the host the square box indicates total free memory. Note the highlighted state where the host's free memory is temporarily exhausted.

The solution to this problem is to adopt a two-phase `memory/target` setting policy. The SQUEEZED daemon forces domains to free memory first before allowing domains to allocate, in-effect forcing the system to move through the left-most state in the diagram above.

6.2 Use of `maxmem`

The XEN domain `maxmem` value is used to limit memory allocations by the domain. The rules are:

1. if the domain has never been run and is paused then `maxmem` ← `reservation` (for information about reservations see Section 3);
   - these domains are probably still being built and we must let them allocate their `startmem`
   - **FIXME**: this "never been run' concept pre-dates the `feature-balloon` flag: perhaps we should use the `feature-balloon` flag instead.
2. if the domain is running and the balloon driver is thought to be working then `maxmem` ← `target`; and
   - there may be a delay between lowering a target and the domain noticing so we prevent the domain from allocating memory when it should in fact be deallocating.
3. if the domain is running and the balloon driver is thought to be inactive then `maxmem` ← $min($`target`$,$ `actual`$)$.
   - if the domain is using more memory than it should then we allow it to make progress down towards its target; however
   - if the domain is using less memory than it should then we must prevent it from suddenly waking up and allocating more since we have probably just given it to someone else
   - **FIXME**: should we reduce the target to leave the domain in a neutral state instead of asking it to allocate and fail forever?

7 Example operation

The scenario in Figure 2 includes 3 domains (domain 1, domain 2, domain 3) on a host. Each of the domains has a non-ideal `totpages'` value.

Recall we also have the policy constraint that:

$$\texttt{dynamic-min} <= \texttt{target} <= \texttt{dynamic-max}$$

Hypothetically if we reduce `target` by `target`−`dynamic-min` (i.e. by setting `target` ← `dynamic-min`) then we should reduce `totpages` by the same amount, freeing this much memory on the host. In the upper-most graph in Figure 2 the total amount of memory which would be freed if we set each of the 3 domain's `target` ← `dynamic-min` is:

$$d1 + d2 + d3$$

In this hypothetical situation we would now have $x + s + d1 + d2 + d3$ free on the host where s is the host slush fund and x is completely unallocated. Since we always want to keep the host free memory above s, we are free to return $x + d1 + d2 + d3$ to guests. If we use the default built-in proportional policy then, since all domains have the same `dynamic-min` and `dynamic-max`, each gets the same fraction of this free memory which we call g:

$$g \stackrel{def}{=} \frac{x + d1 + d2 + d3}{3}$$

For each domain, the ideal balloon target is now `target` = `dynamic-min` + g. The SQUEEZED daemon sets these targets in two phases, as described in Section 6.1

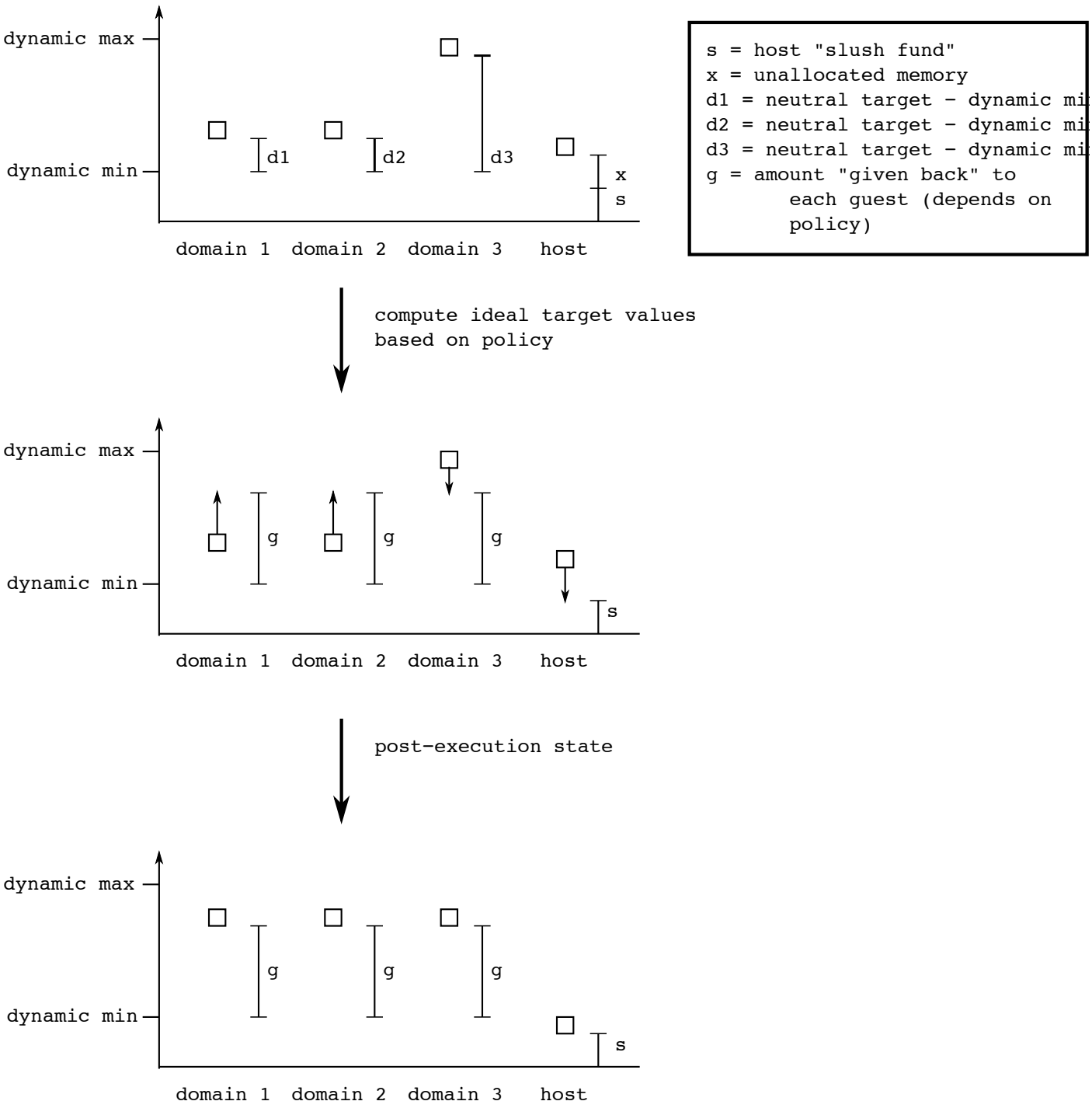

Figure 2: The diagram shows an initial system state comprising 3 domains on a single host. The state is not ideal; the domains each have the same policy settings (`dynamic-min` and `dynamic-max`) and yet are using differing values of `totpages'`. In addition the host has more memory free than desired. The second diagram shows the result of computing ideal target values and the third diagram shows the result after targets have been set and the balloon drivers have responded.

8 The structure of the daemon

The SQUEEZED daemon is a single-threaded daemon which is started by an `init.d` script. It sits waiting for incoming requests on its toolstack interface and checks every 10s whether all domain targets are set to the ideal values (see Section 4). If an allocation request arrives or if the domain targets require adjusting then it calls into the module `ocaml/xenops/squeeze_xen.ml`[6].

The module `ocaml/xenops/squeeze_xen.ml` contains code which inspects the state of the host (through hypercalls and reading xenstore) and creates a set of records describing the current state of the host and all the domains. Note this snapshot of state is not atomic – it is pieced together from multiple hypercalls and xenstore reads – we assume that the errors generated are small and we ignore them. These records are passed into the `ocaml/xenops/squeeze.ml`[7] module where they are processed and converted into a list of *actions* i.e. (i) updates to `memory/target` and; (ii) declarations that particular domains have become *inactive* or *active*. The rationale for separating the XEN interface from the main ballooning logic was to make testing easier: the module `ocaml/xenops/squeeze_test.ml`[8] contains a simple simulator which allows various edge-cases to be checked.

9 Issues

- If a linux domU kernel has the netback, blkback or blktap modules then they away pages via `alloc_empty_pages_and_pagevec()` during boot. This interacts with the balloon driver to break the assumption that, reducing the target by x from a neutral value should free x amount of memory.
- Polling the state of the host (particular the xenstore contents) is a bit inefficient. Perhaps we should move the policy values `dynamic_min` and `dynamic_max` to a separate place in the xenstore tree and use watches instead.
- The memory values given to the domain builder are in units of MiB. We may wish to similarly quantise the `target` value or check that the `memory-offset` calculation still works.
- The XEN patch queue reintroduces the lowmem emergency pool[9]. This was an attempt to prevent guests from allocating lowmem before we switched to a two-phase target setting procedure. This patch can probably be removed.
- It seems unnecessarily evil to modify an *inactive* domain's `maxmem` leaving `maxmem` < `target`, causing the guest to attempt allocations forwever. It's probably neater to move the `target` at the same time.
- Declaring a domain *active* just because it makes small amounts of progress shouldn't be enough. Otherwise a domain could free 1 byte (or maybe 1 page) every 5s.
- Likewise, declaring a domain "uncooperative" only if it has been *inactive* for 20s means that a domain could alternate between *inactive* for 19s and *active* for 1s and not be declared "uncooperative".

[6] `http://www.xen.org/files/XenCloud/ocamldoc/index.html?c=xenops&m=Squeeze_xen`
[7] `http://www.xen.org/files/XenCloud/ocamldoc/index.html?c=xenops&m=Squeeze`
[8] `http://www.xen.org/files/XenCloud/ocamldoc/index.html?c=xenops&m=Squeeze_test`
[9] `http://xenbits.xen.org/xapi/xen-3.4.pq.hg?file/c01d38e7092a/lowmem-emergency-pool`

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for

input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.