Some information about UC4

“Use case 4, replacing hosts: given a schema instance describing h hosts and n au's, an existing assignment of n au's to hosts, a set h_d of hosts being deleted and h_a of hosts being added, such that #h_a => #h_d for each h_di, sum(resources(h_di)) < sum(resources(h_ai)); generate a additional "invitations" (config file sections) for hosts *all* hosts, *keeping the au assignments fixed for hosts in h-h_d*, such that 1a,b are satisfied.”

Like UC5, this case is complex and is broken up into 4 situations, **but these situations are standalone situations and are not cumulative:**

*Add 1 host, remove 0 hosts
*Add 1 host, remove 1 host
*Add 2 hosts, remove 1 host
*Add 2 hosts, remove 2 hosts

The basic idea with this use case is that for a set of hosts that have been invited to harvest some AU’s, zero or more of these hosts will be removed and replaced by one or more hosts, so long as the resources available (number of hosts harvesting and storage commitment) do not decrease. However, a host replacing another host does not necessarily inherit that replaced host’s AU assignments.

I have two problems with how the use case is stated:

*The number of added hosts is greater than or equal to the number of deleted hosts (| |A| | ≥ | |D| |)

The reason for this dictate is to make sure that AU’s that were harvested by a sufficient number of hosts (k=4) do not drop into an insufficient number. The only problem I have with this issue is that hosts periodically go down for many reasons, and often times substitutes are hard do not come online quickly. These use cases do however apply this restriction.

*For each deleted host (D_i), the sum(resources(D)) ≤ sum(resources(A))

My interpretation of what this statement means is that an added host cannot cause a reduction in committed space available. There is a good reason for this restriction--even if you replaced a 100GB committed space computer with 30 10GB committed space computers, even though the total committed space increases, AU’s that were larger than 10GB could not be held.

However, because the statement uses the subscript i for both the deleted host and the added host, what it means is that committed space of a specific added host must be greater than those of a specific removed host. This implication makes no sense because as I have observed, because even though a host may “replace” another host, it does not mean that it automatically inherits its AU invitations and holdings.

The main idea, I think is really this:

For every D_i, there must be at least as many if not more machines with equal or committed resources to it in the set H - D + A [(H \cap D') U A] as there were in the set H.

What this means, is that if there are x machines with at least y committed space originally, where y committed space is the amount of committed space of a host being removed, there need to be at least x machines with at least y committed space after additions and deletions. This is the interpretation I have used for this use case.
**Data-PASS SSP Use Case 4 – Situation 1**

**SCHEMA UC4**

Let there be $h=6$ hosts, where:

$$H = \{\text{lockss-0, lockss-1, lockss-2, props, dris, haar}\}$$

Let there be $n=4$ AU's, where $AU \equiv \{\text{AU0, AU1, AU2, AU3}\}$

Let there be 2 groups to be defined:

1. The set of hosts to be removed, $D$
2. The set of hosts to be added, $A$

Such that:

1. $|D| \leq |A|$
2. For every $D_i$, $i = 0 \rightarrow |D| - 1$, there must be at least as many if not more hosts in the set $H - D + A \cap (H \cap D)$ with equal or greater committed space, as there were in $H$.

Then:

Generate invitations for all hosts such that:

1. The pre-assigned AU's in hosts $H - D \cap H$ are not disturbed.
2. For each AU in $H - D$, there are at least $k=4$ hosts harvesting.
3. For each host, the sum of max size of AU's harvested is less than the storage commitment.

**Situation 1:** Add 1 host, Remove 0 hosts

- $A = \{\text{fong}\}$
- $D = \{\}$

**Conditions:**

- $|D| \leq |A|$ ($0 \leq 1$)
- There are no deleted hosts, so no committed space requirements apply.

**Generate $|H - D + A|$ invitations:**

- *Preserve invitations for hosts in $H - D$:
  - +lockss-0: AU0, AU3
  - +lockss-1: AU0, AU1, AU2, AU3
  - +lockss-2: AU0, AU1, AU2, AU3
  - +props: AU1, AU2
  - +dris: AU1, AU2
  - +haar: AU0, AU3

- *For each AU, $k=4$ hosts are harvesting:
  - +AU0: lockss-0, lockss-1, lockss-2, haar ($k=4$)
  - +AU1: lockss-1, lockss-2, props, dris ($k=4$)
  - +AU2: lockss-1, lockss-2, props, dris ($k=4$)
  - +AU3: lockss-0, lockss-1, lockss-2, haar ($k=4$)

*Generate new invitations:

- +fong: $a$
Situation 2: Add 1 host, remove 1 host

A = \{fong\}
D = \{lockss-0\}

Conditions:
1. \(|D| \leq |A| (1 \leq 1)
2. For every \(D_i, i = 0 \rightarrow |D| - 1\), there must be at least as many if not more hosts in the set \(H - D + A\ [(H \cap D) U A]\) with equal or greater committed space, as there were in \(H\).

Then:
- Generate invitations for all hosts such that:
  1. (Z) The pre-assigned \(AU\)'s in hosts \(H - D\ [H \cap D]\) are not disturbed.
  2. (A) For each \(AU\), there are at least \(k=4\) hosts harvesting.
  3. (B) For each host, the sum of max size of \(AU\)'s harvested is less than the storage commitment.

*Preserve invitations for hosts in \(H-D\):
Data-PASS SSP Use Case 4 – Situation 3

SCHEMA UC4

Let there be h=6 hosts, where:
H = {lockss-0, lockss-1, lockss-2, props, dris, haar}

Let there be n=4 AU's, where AU ≡ {AU0, AU1, AU2, AU3}

Let there be 2 groups to be defined:
* The set of hosts to be removed, D
* The set of hosts to be added, A

Such that:
1. ||D|| ≤ ||A|| (1 ≤ 2)
2. For every D_i, i = 0→||D|| - 1, there must be at least as many if not more hosts in the set H - D + A ((H ∩ D') U A) with equal or greater committed space, as there were in H.

Then:
Generate invitations for all hosts such that:
(2) The pre-assigned AU's in hosts H - D [H ∩ D'] are not disturbed.
(A) For each AU, there are at least k=4 hosts harvesting.
(B) For each host, the sum of max size of AU's harvested is less than the storage commitment.

Machine Ownership Key
ICPSR
Odum Institute
New host
Deleted host

5/28/2008
### SCHEMA UC4

Let there be \( h \) hosts, where:

\[
H = \{ \text{lockss-0, lockss-1, lockss-2, props, dris, haar} \}
\]

Let there be \( n \) AUs, where \( AU \in \{ \text{AU0, AU1, AU2, AU3} \} \)

Let there be 2 groups to be defined:

1. *The set of hosts to be removed, D*
2. *The set of hosts to be added, A*

Such that:

1. \( ||D|| \leq ||A|| \) (2 ≤ 2)
2. For every \( D_i, i = 0 \rightarrow ||D|| - 1 \), there must be at least as many if not more hosts in the set \( H - D + A \) \((H \cap D') \cup A) \) with equal or greater committed space, as there were in \( H \).

Then:

1. Generate invitations for all hosts such that:
   (2) The pre-assigned AUs in hosts \( H - D \) \((H \cap D') \) are not disturbed.
   (B) For each host, the sum of max size of AU's harvested is less than the storage commitment.

### Situation 4: Add 2 hosts, remove 2 hosts

*\( A = \{ \text{fong, lockss-3} \} \)

*\( D = \{ \text{lockss-1, haar} \} \)

**Conditions**

1. \( ||D|| \leq ||A|| \) (2 ≤ 2)
2. For every \( D_i, i = 0 \rightarrow ||D|| - 1 \), there must be at least as many if not more hosts in the set \( H - D + A \) \((H \cap D') \cup A) \) with equal or greater committed space, as there were in \( H \).

For lockss-1 [1700],

- 2 machines had committed space equal or greater than 1700 in \( H \).
- 2 machines had committed space equal or greater than 1700 in \( H - D + A \).

For haar [200],

- 3 machines had committed space equal or greater than 200 in \( H - D + A \).

Generate \( ||H - D + A|| \) (6) invitations:

### Machine Ownership Key

- **ICPSR**
- **Odum Institute**
- **New host**
- **Deleted host**

---

**Data-PASS SSP Use Case 4 – Situation 4**

v 1.1.0

5/28/2008