

including any one element alone or the one element in combination with one or more of the other elements which may also include, in combination, additional elements not listed.

While various embodiments have been described, it will be apparent to those of ordinary skill in the art that many more embodiments and implementations are possible. Accordingly, the embodiments described herein are examples, not the only possible embodiments and implementations.

What is claimed is:

1. An apparatus comprising: a processor configured to execute a virtualization instance, the processor further configured to:

allocate a region of an external primary memory, the region of the external primary memory accessible on a memory appliance by the apparatus over a network, wherein the external primary memory is memory that is external to the apparatus but is primary memory to the apparatus;

allocate at the apparatus a subset of the previously allocated region of the external primary memory to be a slab of the external primary memory in response to a request to allocate the slab, the request comprising an invocation of a programmatic method of a memory allocation interface;

map, at the apparatus, at least the slab of the external primary memory to a virtual address space; and

access the external primary memory on the memory appliance over the network with client-side memory access in which a communication interface of the memory appliance is configured to access the external primary memory, wherein the external primary memory is accessible by the virtualization instance through the virtual address space.

2. The apparatus of claim 1, wherein the processor is configured to add external primary memory to the virtualization instance during execution of the virtualization instance.

3. The apparatus of claim 1, wherein the processor is configured to remove a subset of the external primary memory from the virtualization instance.

4. The apparatus of claim 1, wherein the processor is further configured to perform a snapshot of primary memory of the virtualization instance, the primary memory of the virtualization instance including the external primary memory.

5. The apparatus of claim 4, wherein performance of the snapshot of the primary memory of the virtualization instance includes initiation of a copy of at least one of a file, a filesystem, or a subvolume to which the external primary memory is mapped.

6. The apparatus of claim 4, wherein the virtualization instance executes in a first process, and wherein performance of the snapshot of the primary memory of virtualization instance includes creation of a forked process of the first process, and further includes maintenance of a first view of the primary memory of the virtualization instance for the first process and a second view of the primary memory of the virtualization instance for the forked process, wherein the forked process has access to the snapshot of the primary memory of the virtualization instance through the second view of the primary memory.

7. The apparatus of claim 4, wherein performance of the snapshot of the primary memory of the virtualization instance includes initiation of a snapshot of memory on a memory appliance that backs the external primary memory.

8. The apparatus of claim 4, wherein performance of the snapshot of the primary memory of the virtualization instance includes creation of a copy of contents of a region and/or an allocation of the external primary memory that resides on a memory appliance.

9. The apparatus of claim 1, wherein the programmatic method of the memory allocation interface includes `malloc()`.

10. The apparatus of claim 1, wherein the programmatic method of the memory allocation interface includes `mmap()`.

11. The apparatus of claim 1, wherein the at least the slab includes the region of the external primary memory.

12. The apparatus of claim 1, wherein the invocation of the memory allocation interface is from the virtualization instance.

13. A method comprising:

allocating a first region of an external primary memory for a virtualization instance, the first region of the external primary memory accessible on a memory appliance by a first client over a network, wherein the virtualization instance executes on the first client, and wherein the external primary memory on the memory appliance is accessible over the network with client-side memory access in which a communication interface of the memory appliance is configured to access the external primary memory;

allocating at the first client a subset of the previously allocated first region of the external primary memory to be a slab of the external primary memory in response to a request to allocate the slab, the request comprising an invocation of a programmatic method of a memory allocation interface;

map, at the first client, at least the slab of the external primary memory to a virtual address space, wherein the external primary memory is accessible by the virtualization instance through the virtual address space; and migrating the virtualization instance from the first client to a second client by:

copying data stored in the first region of the external primary memory to a second region, and wherein the first region and the second region are included in the memory appliance; and

resuming operation of the virtualization instance on the second client, wherein the second region is external primary memory for the virtualization instance executing on the second client, and wherein the second region of the external primary memory and the data copied thereto is accessible by the virtualization instance executing on the second client with client-side memory access.

14. The method of claim 13, wherein the memory appliance is a single memory appliance and the first region and the second region are included in the single memory appliance, and wherein the data is copied from the first region to the second region without being transferred to the first client or the second client.

15. The method of claim 13, wherein the memory appliance comprises a first memory appliance and a second memory appliance, wherein the first region is in the first memory appliance and the second region is in the second memory appliance, and the data is copied from the first region to the second region without being transferred to the first client or the second client.

16. The method of claim 15, wherein the data is copied from the first region to the second region using client side memory access.

55

17. The method of claim 13, wherein the virtualization instance includes a virtual machine, a container, a jail, or a zone.

18. A method for limiting local primary memory usage, the method comprising:

backing a plurality of memory portions by a file, a subset of the memory portions included in local primary memory, wherein the memory portions are mapped to an address space of a virtualization instance;

including, in the local primary memory, a plurality of portion-tracking data structures that correspond to the memory portions;

unmapping a page table entry for a selected memory portion from the address space of the virtualization instance;

updating a portion-tracking data structure corresponding to the selected memory portion to indicate that the page table entry for the selected memory portion was dirty when unmapped;

flushing or shooting down a translation lookaside buffer (TLB) for the page table entry;

marking, after the flushing or the shooting down the translation lookaside buffer, the selected memory portion corresponding to the page table entry dirty based on the portion-tracking data structure indicating that the page table entry for the selected memory portion was dirty when unmapped; and

writing the selected memory portion to the file based on the selected memory portion being marked dirty.

19. A method for limiting local primary memory usage, the method comprising:

backing a plurality of memory portions by a file, a subset of the memory portions included in local primary memory;

including, in the local primary memory, a plurality of portion-tracking data structures that correspond to the memory portions;

unmapping a page table entry for a selected memory portion from a virtual address space;

updating a portion-tracking data structure corresponding to the selected memory portion to indicate that the page table entry for the selected memory portion was dirty when unmapped;

56

flushing or shooting down a translation lookaside buffer (TLB) for the page table entry;

marking, after the flushing or the shooting down the translation lookaside buffer, the selected memory portion corresponding to the page table entry dirty based on the portion-tracking data structure indicating that the page table entry for the selected memory portion was dirty when unmapped; and

writing the selected memory portion to the file based on the selected memory portion being marked dirty.

20. The method of claim 19, wherein the file is memory mapped external primary memory.

21. The method of claim 19 further comprising allocating the subset of the memory portions in response to one or more invocations of a programmatic method of a memory allocation interface.

22. The method of claim 19, wherein content of the file is accessed via client side memory access.

23. A method comprising:

allocating an external primary memory for a client, wherein the external primary memory is external to the client but is primary memory to the client, wherein the external primary memory is accessible over a communication interface of the client with client-side memory access in which a communication interface of a memory appliance on which the external primary memory resides is configured to access the external primary memory, and wherein a slab of the external primary memory is allocable at the client, in response to a request at the client to allocate the slab, by the client selecting a subset of the external primary memory previously allocated and mapping the slab of the external primary memory to a virtual address space of the client, wherein the client includes a virtual machine on a first physical machine, the external primary memory is allocated to the virtual machine and includes data accessed by the virtual machine from the first physical machine; and

migrating the virtual machine from the first physical machine to a second physical machine without copying the data by accessing the data in the external primary memory by the virtual machine from the second physical machine.

* * * * *