or operations indicated as optional in the drawings may just be one example of components or operations that are optional. Components or operations not marked as optional may, in fact, be optional in some examples.

Furthermore, although specific components of innovations were described, methods, systems, and articles of manufacture consistent with the innovation may include additional or different components. For example, a processor may be implemented as a microprocessor, microcontroller, application specific integrated circuit (ASIC), discrete logic, or a combination of other type of circuits or logic. Similarly, memories may be DRAM, SRAM, Flash or any other type of memory. Flags, data, databases, tables, entities, and other data structures may be separately stored and managed, may be incorporated into a single memory or database, may be distributed, or may be logically and physically organized in many different ways. The components may operate independently or be part of a same program. The components may be resident on separate hardware, such as separate removable circuit boards, or share common hardware, such as a same memory and processor for implementing instructions from the memory. Programs may be parts of a single program, separate programs, or distributed across several memories and processors.

The respective logic, software or instructions for implementing the processes, methods and/or techniques discussed throughout this disclosure may be provided on computer-readable media or memories or other tangible media, such as a cache, buffer, RAM, removable media, hard drive, other computer readable storage media, or any other tangible media or any combination thereof. The tangible media include various types of volatile and nonvolatile storage media. The functions, acts or tasks illustrated in the figures or described herein may be executed in response to one or more sets of logic or instructions stored in or on computer readable media. The functions, acts or tasks are independent of the particular type of instructions set, storage media, processor or processing strategy and may be performed by software, hardware, integrated circuits, firmware, micro code, or any type of other processor, operating alone or in combination. Likewise, processing strategies may include multiprocessing, multitasking, parallel processing and/or any other processing strategy known now or later discovered. In one embodiment, the instructions are stored on a removable media device for reading by local or remote systems. In other embodiments, the logic or instructions are stored in a remote location for transfer through a computer network or over telephone lines. In yet other embodiments, the logic or instructions are stored within a given computer, CPU, GPU, or system.

A second action may be said to be "in response to" a first action independent of whether the second action results directly or indirectly from the first action. The second action may occur at a substantially later time than the first action and still be in response to the first action. Similarly, the second action may be said to be in response to the first action even if intervening actions take place between the first action and the second action, and even if one or more of the intervening actions directly cause the second action to be performed. For example, a second action may be in response to a first action if the first action sets a flag and a third action later initiates the second action whenever the flag is set.

To clarify the use of and to hereby provide notice to the public, the phrases "at least one of <A>, <B>, . . . and <N>" or "at least one of <A>, <B>, . . . <N>, or combinations thereof" or "<A>, <B>, . . . and/or <N>" are defined by the Applicant in the broadest sense, superseding any other implied definitions hereinbefore or hereinafter unless expressly asserted by the Applicant to the contrary, to mean one or more elements selected from the group comprising A, B, . . . and N. In other words, the phrases mean any combination of one or more of the elements A, B, . . . or N 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 of the innovation have been described, it will be apparent to those of ordinary skill in the art that many more embodiments and implementations are possible within the scope of the innovation. Accordingly, the innovation is not to be restricted except in light of the attached claims and their equivalents.

What is claimed is:

1. An external memory system comprising:
a client device; and
a memory appliance;
the client device comprising:
a first communication interface; and
a local memory comprising:
an application logic unit; and
a client logic unit;
the memory appliance comprising:
a processor;
a second communication interface;
a memory configured in an address space addressable by the processor; and
a region access unit;
wherein the client logic unit is configured to cause allocation of a region of the memory that is accessible on the memory appliance by the client device through the first communication interface,
wherein the region access unit is configured to allocate the region of the memory for use as external primary memory of the client device,
wherein the client logic unit is further configured to:
receive a first memory allocation request from the application logic unit to allocate a portion of memory in a virtual address space of the application logic unit;
select, at the client device and in response to the first memory allocation request to allocate the portion of memory, a subset of the region of the memory in the memory appliance; map, at the client device, at least the requested portion of memory to the virtual address space; and, in response to a second memory allocation request, allocate a subset of the portion of memory mapped in the virtual address space of the application logic unit, wherein data in the portion of memory is accessible by the client device via client-side memory access in which the second communication interface of the memory appliance is configured to access the region of the memory in the memory appliance.

2. The system of claim **1**, wherein the client logic unit is configured to map the at least the portion of memory to the virtual address space before the subset of the region of the memory is selected.

3. The system of claim **1**, wherein the application logic unit includes an operating system, a kernel, a device driver, a virtual machine, a hypervisor, a container, or a jail included in the client device.

4. The system of claim **1**, wherein the client logic unit presents a data interface to the application logic unit to read and/or write data to a specific location within the region of the memory.

5. The system of claim 4, wherein the data interface includes an API, a block-level interface, a character-level interface, a memory-mapped interface, a memory allocation interface, a memory swapping interface, a memory caching interface, a hardware-accessible interface, or a graphics processing unit (GPU) accessible interface.

6. The system of claim 5, wherein the memory-mapped interface includes and/or utilizes an interface which provides character-level, byte-level, and/or block-level access to data of the region of the memory.

7. The system of claim 1, wherein the external memory is accessible from the client device via a memory swapping interface, a swap device, and/or a swap file included in the client device.

8. The system of claim 1, wherein the processor is configured to:

memory map at least a portion of a file to the region of the memory, wherein a virtual address addressable by the processor is generated, and the at least a portion of the file is accessible through the region of the memory at the virtual address; and

register the virtual address with the second communication interface, wherein registration of the virtual address provides client-side memory access to the region of the memory, wherein the client-side memory access provides the client device access to the region of the memory over a network.

9. The system of claim 8, wherein the file is a pseudo file, a block device file, and/or a character device file.

10. The system of claim 8, wherein, the second communication interface is configured to generate an I/O fault in response to an attempt to access a portion of the region that is not present.

11. The system of claim 1, wherein the processor is further configured to perform a batch portion invalidation and/or reclaim of portions of the region of the memory and indicate to the second communication interface that the portions of the region of the memory are reclaimed and/or invalidated.

12. A method comprising:

allocating a region of memory in a memory appliance as external memory for a client device, wherein external memory is memory that is external to the client device but treated as primary memory at the client device;

receiving, at the client device from a component in the client device, a first memory allocation request to allocate a portion of memory in a virtual address space of a process and/or a thread;

selecting, at the client device in response to the first memory allocation request to allocate the portion of memory, a subset of the region of memory in the memory appliance to be the portion of memory allocated;

mapping at least a portion of a file to a corresponding portion of the virtual address space of the process and/or the thread, wherein the portion of the file corresponds to the selected subset of the region of memory, and wherein the file is a pseudo file;

allocating a subset of the portion of memory in the virtual address space of the process and/or the thread in response to a second memory allocation request;

reading or writing data at an offset within the file, thereby causing a read from or a write to a corresponding offset within the region of memory in the memory appliance, wherein the read or the write is performed via client-side memory access in which a communication interface of the memory appliance is configured to access the region of memory in the memory appliance.

13. The method of claim 12, wherein the second memory allocation request to allocate the portion of memory is made through a memory allocation interface.

14. The method of claim 12, further comprising performing the read or the write in a page fault handler in response to a page fault.

15. The method of claim 12, further comprising making the external memory accessible by an application logic unit through a memory swapping interface.

16. The method of claim 15, further comprising emulating a swap file with the memory swapping interface.

17. The method of claim 12, further comprising compressing the data prior to transmission between the client device and the memory appliance.

18. The method of claim 12, further comprising recoverably reducing a size of the data prior to transmission between the client device and the memory appliance.

19. The method of claim 12, further comprising compressing the data, in response to the write, by the client device, a communication interface, and/or a logic included with the client device.

20. The method of claim 12, further comprising recoverably reducing a size of the data, in response to the write, by the client device, a communication interface, and/or a logic included with the client device.

* * * * *