CLAIMS

1. A computer system, comprising:

a configuration comprising a physical core enabled to operate in a single thread (ST) mode and a multithreading (MT) mode, wherein the physical core includes multiple physical threads; and

a host program configured to execute in the ST mode on the physical core to perform a method comprising:

issuing a start-virtual-execution (start-VE) instruction to dispatch a guest entity on the physical core, wherein the guest entity includes all or part of a guest virtual machine (VM), wherein the start-VE instruction is executed by the physical core, and wherein the executing comprises:

obtaining a first state description from a location specified by the start-VE instruction, wherein the first state description includes a guest state;

determining, based on the guest state, whether the guest entity includes a single guest thread or multiple guest threads;

based on the guest state and on determining that the guest entity includes multiple guest threads, starting the multiple guest threads on the physical core in the MT mode, wherein the multiple guest threads execute independently of each other; and

based on the guest state and on determining that the guest entity includes a single guest thread, starting the single guest thread on the physical core in the ST mode.

2. The computer system of claim 1, wherein the physical core includes computer instructions that control the use of shared resources between the multiple physical threads when the physical core is in the MT mode.

3. The computer system of claim 1, wherein the host program manages at least a portion of the guest entity as a single logical core when the physical core is in the MT mode.

4. The computer system of claim 1, wherein a thread validity mask is utilized by the host program to indicate the validity of the multiple guest threads in the guest entity.

5. The computer system of claim 1, wherein the method further comprises exiting all of the multiple guest threads in the guest entity prior to returning control to the host program.

6. The computer system of claim 1, wherein based on the guest entity including the multiple guest threads, at least a portion of state data that is common to all the multiple guest threads is stored in a single location.

7. The computer system of claim 1, wherein based on the guest entity including the multiple guest threads, state data for one thread is contained in a first state description, and state data for each of one or more additional threads are contained in an additional state description.

8. The computer system of claim 7, wherein the first state description and the additional state description are stored in at least one of a ring and a list structure.

9. The computer system of claim 1, further comprising performing a no-action exit, wherein the no-action exit includes exiting a guest thread based on a request from another guest thread.

10. A computer implemented method for dispatching multiple threads in a configuration comprising a physical core enabled to operate in a single thread (ST) mode and a multithreading (MT) mode, the physical core including multiple physical threads, and the computer implemented method comprising:

issuing, by a host program executing in the ST mode on the physical core, a start-virtual-execution (start-VE) instruction to dispatch a guest entity on the physical core, wherein the guest entity includes all or part of a guest virtual machine (VM), wherein the start-VE instruction is executed by the physical core, and wherein the executing comprises:

obtaining a first state description from a location specified by the start-VE instruction, wherein the first state description includes a guest state;

determining, based on the guest state, whether the guest entity includes a single guest thread or multiple guest threads;

based on the guest state and on determining that the guest entity includes multiple guest threads, starting the multiple guest threads on the physical core in the MT mode, wherein the multiple guest threads execute independently of each other; and

based on the guest state and on determining that the guest entity includes a single guest thread, starting the single guest thread on the physical core in the ST mode.

11. The computer implemented method of claim 10, wherein the physical core includes computer instructions that control the use of shared resources between the multiple physical threads when the physical core is in the MT mode.

12. The computer implemented method of claim 10, wherein the host program manages at least a portion of the guest entity as a single logical core when the physical core is in the MT mode.

13. The computer implemented method of claim 10, wherein a thread validity mask is utilized by the host program to indicate the validity of the multiple guest threads in the guest entity.

14. The computer implemented method of claim 10, further comprising exiting all of the multiple guest threads in the guest entity prior to returning control to the host program.

15. The computer implemented method of claim 10, wherein based on the guest entity including the multiple guest threads, state data for one thread is contained in a first state description, and state data for each of one or more additional threads are contained in an additional state description.

16. A computer program product for dispatching multiple threads in a configuration comprising a physical core enabled to operate in a single thread (ST) mode and a multithreading (MT) mode, the physical core including multiple physical threads, and the computer program product comprising:

a computer readable storage medium having program instructions embodied therewith, wherein the computer readable storage medium is not a signal, and wherein the program

instructions are readable by a processing circuit to cause the processing circuit to perform a method comprising:

issuing, by a host program executing in the ST mode on the physical core, a start-virtual-execution (start-VE) instruction to dispatch a guest entity on the physical core, wherein the guest entity includes all or part of a guest virtual machine (VM), wherein the start-VE instruction is executed by the physical core, and wherein the executing comprises:

obtaining a first state description from a location specified by the start-VE instruction, wherein the first state description includes a guest state;

determining, based on the guest state, whether the guest entity includes a single guest thread or multiple guest threads;

based on the guest state and on determining that the guest entity includes multiple guest threads, starting the multiple guest threads on the physical core in the MT mode, wherein the multiple guest threads execute independently of each other; and

based on the guest state and on determining that the guest entity includes a single guest thread, starting the single guest thread on the physical core in the ST mode.

17. The computer program product of claim 16, wherein the host program manages at least a portion of the guest entity as a single logical core when the physical core is in the MT mode.

18. The computer program product of claim 16, wherein a thread validity mask is utilized by the host program to indicate the validity of the multiple guest threads in the guest entity.

19. The computer program product of claim 16, wherein the method further comprises exiting all of the multiple guest threads in the guest entity prior to returning control to the host program.

20. The computer program product of claim 16, wherein based on the guest entity including the multiple guest threads, state data for one thread is contained in a first state description, and state data for each of one or more additional threads are contained in an additional state description.