

AMENDED CLAIMS

received by the International Bureau on 07 March 2016 (07.03.16)

1. (Original) A computing system including:

a storage system storing one or more graph-based program specifications, at least a first graph-based program specification including a plurality of components corresponding to tasks and directed links between ports of the components, the first graph-based program specification including:

(1) a first type of link configuration between ports of linked components, corresponding to transfer of control or transfer of a single data element, and

(2) a second type of link configuration between ports of linked components, corresponding to transfer of multiple data elements;

a compiler configured to generate a target program specification from the first graph-based program specification, the target program specification including control code representing at least one control graph including graph nodes representing the components in the first graph-based program specification, where two or more of the graph nodes are connected based on links having the first type of link configuration; and

one or more computing nodes, each including at least one processor, with at least a first of the computing nodes being configured to:

initiate execution of the target program specification, and

manage computing resources for links having the second type of link configuration, the computing resources including at least one of: (1) a buffer for storing data elements provided by an output port, or (2) a buffer for storing data elements provided to an input port.

2. (Original) The computing system of claim 1, wherein the computing resources include at least a first buffer for storing data element provided to an input port.

3. (Original) The computing system of claim 2, wherein the first buffer stores unordered data elements, where a first data element is retrieved from the first buffer for processing by a first instance of a first of the components without blocking retrieval of any second data element from the first buffer for processing by a second instance of the first component until after the first instances completes processing the first data element.

4. (Original) The computing system of claim 2, wherein managing computing resources includes filling the first buffer in a manner that does not cause the first buffer to grow beyond a capacity of the first of the computing nodes.

5. (Original) The computing system of any preceding claim, wherein the control graph includes: a first set of connected graph nodes, and a second set of connected graph nodes; where no graph node in the first set of graph nodes is directly connected to any graph node in the second set of graph nodes.

6. (Original) The computing system of claim 5, wherein the control code enables execution of tasks corresponding to components represented by the first set of graph nodes concurrently with tasks corresponding to components represented by the second set of graph nodes.

7. (Original) The computing system of claim 1, wherein the compiler is configured to identify one or more sets of one or more tasks corresponding to respective components.

8. (Original) The computing system of claim 7, wherein the target program specification includes control code representing a control graph for each of the identified sets.

9. (Original) The computing system of claim 7, wherein each set of one or more tasks has a single collection of data elements as a data source that provides different data elements to different instances of the set of one or more tasks.

10. (Original) The computing system of claim 7, wherein each set of one or more tasks corresponds to at least one directed acyclic graph of one or more nodes, where each node of the graph corresponds to a task or an identified set and each directed edge of the graph corresponds to a link having the first type of link configuration.

11. (Original) The computing system of claim 10, wherein the directed acyclic graph of one or more nodes has a single root node.

12. (Original) The computing system of claim 10, wherein at least a first graph of one or more nodes includes at least one node corresponding to a second graph of one or more nodes nested within the first graph.

13. (Original) The computing system of any preceding claim, wherein the control code implements a state machine that controls execution of the target program specification.

14. (Original) The computing system of claim 13, wherein the control graph includes a first node used by the state machine to manage control signals that are used to begin execution of one or more components represented by nodes in the control graph.

15. (Original) The computing system of claim 14, wherein the compiler is configured to: inspect link configurations of components to determine any component that are not linked to an upstream component by a link having the first type of link configuration, and connect any nodes representing such components to the first node.

16. (Original) The computing system of claim 14, wherein the control graph includes a second node used by the state machine to determine when all components represented by nodes in the control graph have finished execution.

17. (Original) A method including:

storing one or more graph-based program specifications, at least a first graph-based program specification including a plurality of components corresponding to tasks and directed links between ports of the components, the first graph-based program specification including:

- (1) a first type of link configuration between ports of linked components,
corresponding to transfer of control or transfer of a single data element,
and
- (2) a second type of link configuration between ports of linked components,
corresponding to transfer of multiple data elements;

generating a target program specification from the first graph-based program specification, the target program specification including control code representing at least one control graph including graph nodes representing the components in the first graph-based program specification, where two or more of the graph nodes are connected based on links having the first type of link configuration; and

using at least one computing node,

initiating execution of the target program specification, and

managing computing resources for links having the second type of link configuration, the computing resources including at least one of: (1) a buffer for storing data elements provided by an output port, or (2) a buffer for storing data elements provided to an input port.

18. (Original) Software stored in a non-transitory form on a computer-readable medium, the software including instructions for causing a computing system to:

store one or more graph-based program specifications, at least a first graph-based program specification including a plurality of components corresponding to tasks and directed links between ports of the components, the first graph-based program specification including:

(1) a first type of link configuration between ports of linked components,
corresponding to transfer of control or transfer of a single data element,
and

(2) a second type of link configuration between ports of linked components,
corresponding to transfer of multiple data elements;

generate a target program specification from the first graph-based program specification,
the target program specification including control code representing at least one
control graph including graph nodes representing the components in the first
graph-based program specification, where two or more of the graph nodes are
connected based on links having the first type of link configuration; and

using at least one computing node,

initiate execution of the target program specification, and

manage computing resources for links having the second type of link
configuration, the computing resources including at least one of: (1) a
buffer for storing data elements provided by an output port, or (2) a buffer
for storing data elements provided to an input port.

19. (New) The computing system of claim 1, wherein the graph nodes in the control
graph are connected based on links having the first type of link configuration, but not based on
links having the second type of link configuration.

20. (New) The computing system of claim 1, wherein the computing resources
include buffers allocated for links having the second type of link configuration, but not for links
having the first type of link configuration.