Orchestrator Telemetry Processing Pipeline

A multi-platform architecture is used to build and manage a telemetry-processing pipeline.

NASA's Jet Propulsion Laboratory, Pasadena, California

Orchestrator is a software application infrastructure for telemetry monitoring, logging, processing, and distribution. The architecture has been applied to support operations of a variety of planetary rovers. Built in Java with the Eclipse Rich Client Platform, Orchestrator can run on most commonly used operating systems. The pipeline supports configurable parallel processing that can significantly reduce the time needed to process a large volume of data products.

Processors in the pipeline implement a simple Java interface and declare their required input from upstream processors. Orchestrator is programmatically constructed by specifying a list of Java processor classes that are initiated at runtime to form the pipeline. Input dependencies are checked at runtime. Fault tolerance can be configured to attempt continuation of processing in the event of an error or failed input dependency if possible, or to abort further processing when an error is detected.

This innovation also provides support for Java Message Service broadcasts of telemetry objects to clients and provides a file system and relational database logging of telemetry. Orchestrator supports remote monitoring and control of the pipeline using browser-based JMX controls and provides several integration paths for pre-compiled legacy data processors.

At the time of this reporting, the Orchestrator architecture has been used by four NASA customers to build telemetry pipelines to support field operations. Example applications include high-volume stereo image capture and processing, simultaneous data monitoring and logging from multiple vehicles. Example telemetry processors used in field test operations support include vehicle position, attitude, articulation, GPS location, power, and stereo images.

This work was done by Mark Powell, David Mittman, Joseph Joswig, Thomas Crockett, and Jeffrey Norris of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-44561.

Scheme for Quantum Computing Immune to Decoherence

The spintronic encodings of this scheme are more efficient than those of a prior scheme.

NASA's Jet Propulsion Laboratory, Pasadena, California

A constructive scheme has been devised to enable mapping of any quantum computation into a spintronic circuit in which the computation is encoded in a basis that is, in principle, immune to quantum decoherence. The scheme is implemented by an algorithm that utilizes multiple physical spins to encode each logical bit in such a way that collective errors affecting all the physical spins do not disturb the logical bit. The scheme is expected to be of use to experimenters working on spintronic implementations of quantum logic.

Spintronic computing devices use quantum-mechanical spins (typically, electron spins) to encode logical bits. Bits thus encoded (denoted qubits) are potentially susceptible to errors caused by noise and decoherence. The traditional model of quantum computation is based partly on the assumption that each qubit is implemented by use of a single two-state quantum system, such as an electron or other spin-½ particle. It can be surprisingly difficult to achieve certain gate operations — most notably, those of arbitrary one-qubit gates — in spintronic hardware according to this model. However, ironically, certain two-qubit interactions (in particular, spin-spin exchange interactions) can be achieved relatively easily in spintronic hardware.

Therefore, it would be fortunate if it were possible to implement any one-qubit gate by use of a spin-spin exchange interaction. While such a direct representation is not possible, it is possible to achieve an arbitrary 1-qubit gate indirectly by means of a sequence of four spin-spin exchange interactions, which could be implemented by use of four exchange gates. Accordingly, the present scheme provides for mapping any one-qubit gate in the logical basis into an equivalent sequence of at most four spin-spin exchange interactions in the physical (encoded) basis. The complexity of the mathematical derivation of the scheme from basic quantum principles precludes a description within this article; it must suffice to report that the derivation provides explicit constructions for finding the exchange couplings in the physical basis needed to implement any arbitrary one-qubit gate. These constructions lead to spintronic encodings of quantum logic that are more efficient than those of a previously published scheme that utilizes a universal but fixed set of gates.

Given this mapping, universal quantum computation could be achieved in the encoded basis if, in addition, it were also possible to implement a controlled-NOT (CNOT) gate in the encoded basis. It had been demonstrated in prior research that such encoded construction of a CNOT gate is possible. Hence, in the present scheme, the mapping of arbitrary one-qubit gates is augmented with the encoded construction of CNOT gates, making it theoretically possible to perform universal quantum computation in the encoded basis.

This work was done by Colin Williams and Farrokh Vatan of Caltech for NASA's Jet Propulsion Laboratory.

The software used in this innovation is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (818) 393-2827. Refer to NPO-42996.