Reconfiguration of a Distributed Information Fusion System

Éric Benoit, Marc-Philippe Huget, Patrice Moreaux, Olivier Passalacqua

LISTIC, Polytech'Savoie, Université de Savoie, France
Outline

1. Context
   - Information Fusion and Information Fusion Systems
2. Example
   - System reconfiguration
3. Model
4. Framework architecture
5. Reconfiguration
   - Error detection and configuration improvement
6. Conclusion
   - Contributions and future work
An Information Fusion Process (IFP) is defined by:
- information sources
- a process

An Information Fusion System (IFS):
- executes an IFP,
- consists of hardware and software elements,
- is often distributed,
- is more and more dynamic:
  - the resources.
Reconfiguration example

A video of the user's head is produced by his smart-phone.

A filter is applied on the video by computers in room 3.

The original and the filtered videos are initially displayed in room 1 and then in room 2.

Reconfiguration appends when the user moves from room 1 to room 2.
Fusion process

The IFP is described by a discrete data-flow graph. Nodes are fusion elements (data sources, fusion nodes, data sinks).

Compatible with usual information fusion models.
A configuration defines:
- an assignment of the fusion elements,
- a path through the network for each data transfer.
Two sub-systems

Search for a new configuration.

Deployment and execution of a configuration.

Reconfiguration of a Distributed Information Fusion System // olivier.passalacqua@univ-savoie.fr
An execution node is a set of host systems. A host system is a set of execution frameworks.

Reconfiguration of a Distributed Information Fusion System // olivier.passalacqua@univ-savoie.fr
Need for reconfiguration (1)

Configuration errors and failures:
  * failure state
    - the system cannot produce some of the results,

Two kinds of errors are detected:
  * Inter-execution framework errors:
    - an execution framework "disappears",
  * Intra-execution framework errors:
    - deadlock on a resource,
    - time-out connection inside an execution element.

Error recovery strategy
  * selection of the first configuration compatible with the available resources,
Need for reconfiguration (2)

Configuration improvement:
- even without any error, the control system tries to improve the runtime system permanently.

A formal model based on GSPN is built, at runtime, from the configuration of the system.
To improve the runtime system:
- the control system updates the model parameters,
- and computes its rewards.

Configuration improvement strategy
- a "significantly better" configuration is selected,
  - the comparison is based on reward values computed from the configuration model.
Transition between configurations (1)

Best effort policy:
- transition in the case of an **implementation swap**: results in queues are preserved.

- myproject.MyFunction1
  - x1
  - x2
  - x3
  - \( \theta^1 \)
- myproject.MyFunction2
  - y1
  - y2

Reconfiguration of a Distributed Information Fusion System // olivier.passalacqua@univ-savoie.fr
Best effort policy:
- transition in the case of a new assignment:
  only updated execution elements are deleted.
Contributions

Model
- Information Fusion Process description
- Configuration model
  - fusion elements assignment
  - links assignment
- Runtime evaluation of the configuration

Implementation
- Reconfiguration mechanism
  - error recovery
  - configuration improvement
- Transition policy
  - implementation swap
  - new assignment
Future work

Model
- Configuration search algorithm
  - definition of the characteristics of a configuration from its description,

Implementation
- Distributed configuration evaluation
- Extension of the framework architecture
Any questions?
Selection of a new configuration

Constraint programming approach

Constraints
- fusion functions are compatible with a subset of execution framework,
- connections between fusion functions are fixed,

Objectives
- global maximal usage of the set of the execution frameworks,
- global "short" physical communication between fusion functions.

The problem is finally sent to a CSP solver.