Quantum Computing Approach to V&V of Complex Systems Overview

Summary of Quantum Enabled V&V Technology

June 12, 2014

Todd Belote
Chris Elliott
Flight Controls / VMS Integration
Discussion Layout

I. Quantum Computing S/W V&V Overview
  • What is this product and who will use it?
  • Why is this tool unique?
  • Where are the cost savings?
    • LM Aero Flight Control Design Process Improvement

II. Hypothetical Example
  • Limiter Software Component in a Flight Control Application

III. Summary Technical Description of QCITL S/W V&V Method
  • Limiter Example
  • Finite State Model Example in Design Phase

IV. Path Forward
What is this product?

- QVTrace*: This technology is a method for Software Verification & Validation using Quantum Computer Assisted Formal Methods.

Who will use it?

- Target Users are System/Software Design Teams interested in:
  - Reducing development costs
  - Improving final product quality

*Product Developed by Quantum Research Analytics
Why is this Tool Unique?

- Hidden Defects in Safety Critical Software Today
  - If not detected with specific Checkcase, leads to failure in field
  - Quantum Verification & Validation Method detects this issue without need for creating specific Checkcase

- Conventional testing to QVV testing

Safety Critical Software
State Space

Checkcase test spotlight investigates specific area of software state space
Why is this **Tool Unique?**

- Hidden Defects in Safety Critical Software Today
  - If not detected w/ specific Checkcase, leads to failure in field
  - Quantum Verification & Validation Method detects this issue without need for creating specific Checkcase

- Conventional testing to QVV testing

**Matrix of Testing**

**Safety Critical Software**

**State Space**

Checkcase test spotlight investigates specific area of software state space
Why is this Tool Unique?

• Hidden Defects in Safety Critical Software Today
  • If not detected with specific Checkcase, leads to failure in field
  • Quantum Verification & Validation Method detects this issue without need for creating specific Checkcase

• Conventional testing to QVV testing

Safety Critical Software State Space

QVV tests state space in entirety with better scaling using quantum hardware*

*Improving the scaling problem with the QC is a theory and remains to be proven
D-Wave Adiabatic Quantum Computer

Current State-of-the-Art

$H(t) = A(t) \sum_{i=1}^{N} \sigma_{x}^{i} + B(t) \left[ \sum_{i=1}^{N} h_{i} \sigma_{z}^{i} + \sum_{i,j\in E} J_{ij} \sigma_{z}^{i} \sigma_{z}^{j} \right]$
Where are the Cost Savings?

OLD

Requirements (Textual)

Design

Build

Test Planning and Development

Test Execution

NEW

Requirements Modeling

Verification Modeling

Design & Analysis

Test Generation & Analysis

Implementation & Integration

Test Execution & Analysis

Highly Integrated Verification

QE-VV Approach Reduces Overall Cost

Defects Require Rework

Fewer Defects

Less Rework

Lower Risk

Shorter Development Schedule

Copyright 2014, Lockheed Martin Corporation. All rights reserved.
II. Hypothetical Example

Simplified Load Factor Control Block diagram

- Mission and Flight Control Software “Limits” Maximum Load Factor

Limiter: What is it?

Requirement: Ensure a Signal is Bounded by [up, lo]

Requirements (Pseudo Code)

```plaintext
if u > up
    y = up
elseif u < lo
    y = lo
else
    y = u
end
```
Limiter: Bounds May Be Dynamic

Dynamic Upper Bound
Limiter: Failure in the Field

Later detection is expensive and could be dangerous.

Hypothetical Failure:
- External Subsystem Provides Invalid Bound (missed in Stand Alone Test!)
- Likely found in Integration or Flight Test

Can we find this before the Lab? Or Flight!
Quantum Enabled V&V Overview

Implementation

Requirements

SMT Instance
False = Defect Space

Satisfiability
(All-SAT)

Modulo Theory
(Number Domain)

Ising
(Binary Optimization)

Quantum Computer

Feasible?

Consistent Reqs/Implementation

Defect Detected Inconsistency in Reqs/Implementation

Copyright 2014, Lockheed Martin Corporation. All rights reserved.
Quantum Enabled V&V Overview

Requirements + Implementation are melded into a Satisfiability Modulo Theory (SMT) instance describing behavior of entire system.
End to End Overview (Limiter)

THE MELDING...

Requirements: “What we want”

\[ up \geq y \geq lo \]

Implementation: “What we built”

\[
\begin{align*}
(u < up) \land (u > lo) & \rightarrow y = u \\
(u > up) & \rightarrow y = up \\
(u < lo) & \rightarrow y = lo
\end{align*}
\]

Statements are melded together – Substitute requirement expression for output variable in implementation as follows:

\[
\begin{align*}
(u < up) \land (u > lo) & \rightarrow (up \geq u \geq lo) \\
(u > up) & \rightarrow (up \geq up \geq lo) \\
(u < lo) & \rightarrow (up \geq lo \geq lo)
\end{align*}
\]

where \(s1 = (u < up) \land (u > lo)\) and \(s2 = (u > up) \land (u > lo)\)

where \(s3 = (u > up)\) and \(s4 = (u > up) \land (u > up)\)

where \(s5 = (u < lo)\) and \(s6 = (u > lo) \land (u > lo)\)

Melded Problem Now Represents Requirements and Code
Quantum Enabled V&V Overview

Boolean SAT extracted from SMT instance. Transformed to Ising Formula and Solved for ALL satisfying assignments.

Satisfiability (All-SAT)  Modulo Theory (Number Domain)

Quantum Computer

Ising (Binary Optimization)

Feasible?

Defect Detected Inconsistency in Reqs/Implementation

Consistent Reqs/implementation

No

Ground State

Yes
End to End Overview (Limiter)

Transform the SAT Statement to Ising
- Depict the SAT Statement as a Logical Circuit
- Auxiliary Variables are Added after each Logical Operator

\[(\neg s_1 \lor s_2) \land (\neg s_3 \lor s_4) \land (\neg s_5 \lor s_6) = \text{FALSE}\]

Note, The Output Z is set to FALSE as \(Z = -1\)

This is in terms of the AQC Computational Basis \{-1, +1\} as opposed to the Typical Binary Computational Basis \{0, 1\}
End to End Overview (Limiter)

Using Graph and Energy Equations, Formulate Hamiltonian

Expanding Scalar Equation...

Note: Equation will be normalized in following steps (divided by highest coefficient), due to AQC constraints (Coupler/Edges and Qubits/Node coefficients are bound by +/- 1)

\[ H = a_1 + a_2 + a_3 + 3a_4 + s_2 + s_4 + s_6 - z_1 - 2z_2 - z_3 - z_2' - 2z_3' - 2a_1c_1 - 2a_2c_2 
- 2a_3c_3 - 2a_4c_4 + a_1s_1 + a_1s_2 + a_2s_3 + a_2s_4 + a_3s_5 + a_3s_6 + a_4z_1 - 2a_4z_3 
- 2c_1z_1 - 2c_2z_2 - 2c_3z_3' - 2c_4z_2' - 2s_2z_1 - 2s_4z_2 - 2s_6z_3' - 3z_2z_2' 
+ z_3z_2p - 3z_3z_3p - 2 \]

Note: Constant Terms do not impact the solution that minimizes this equation and are dropped

Scalar Hamiltonian
End to End Overview (Limiter)

Embed Graph to AQC

**Embedding**, Expand Problem to Sparse Form, Sized for the AQC Chip, \( m=512 \) nodes for Vesuvius

---

**Embedding Heuristics**

Problem Graph

Vesuvius Embedding
End to End Overview (Limiter)

**Solve** Ising Problem with AQC (Adiabatic Quantum Computer)

Due to the **Stochastic** Nature of the D-Wave Quantum Annealing Process, the Ising Problem is solved multiple times and a probability distribution of the solution is formulated.

*The lowest energy groundstate is returned as the potential defect space $S$ for the next phase of analysis.*

\[
\text{ISING } s^* = \arg\min_s \left\{ \sum_{(i,j) \in E} s_i J_{i,j} s_j + \sum_{i \in V} h_i s_i \right\}
\]

The solution represents the Satisfying Argument to the SMT instance. If the solution is determined to be feasible (to be discussed), a defect has been identified.

---

**Adiabatic Quantum Computer Stochastic Distribution**
Quantum Enabled V&V Overview

- Step 3:
  - Modulo/Number Theory Problem
  - Classical Solver → Conduct Feasibility Analysis Using AQC Output (Adiabatic Quantum Computer Ground State Solution of SMT Instance)

Feasible?

- No
- Yes

Consistent
Reqs/Implementation

Defect Detected
Inconsistency in
Reqs/Implementation

Quantum Computer

Satisfiability
(All-SAT)

Ising
(Binary Optimization)

Modulo Theory
(Number Domain)
End to End Overview (Limiter)

- Modulo/Number Theory Problem
  - Classical Solver → Conduct Feasibility Analysis Using AQC
    Output (Adiabatic Quantum Computer Ground State Solution of SMT Instance)

Using a Modulo Theory Solver (MTS) determine feasibility of all ground state solutions (37 in this case)
{ False, False, False, True, True, False }
{ ¬s1 & ¬s2 & ¬s3 & s4 & s5 & ¬s6 }

Given these modulo definitions, Statement Always “Unsatisfiable”
not u < lo or not lo <= up or up < u or not up <= up or lo <= lo and lo <= up or lo <= u and u <= up
Therefore we conclude an Solution 1 is INFEASIBLE -> DEFECT FREE

However, Swapped Bounds Problem is Detected in
1 of the 37 Quantum Solutions

SATISFIABLE

Feasible Ground State → BUG Detected!
QE-V&V Example

AUTOPilot

TRANSITION
mode=0;
request=1;
maneuver=0;

[supported && good]

[standby]

STANDBY
mode=1;
request=0;
maneuver=0;

[standby]
[standby && good]

MANEUVER
mode=1;
request=0;
%designfix maneuvers=1;

[APfailure]

NOMINAL
mode=1;
request=1;
maneuver=0;

[supported && good]

SENSOR

NOMINAL
good=1;

[SENfailure]

TRANSITION
good=1;

[request]
[request && mode]

MANEUVER
mode=1;
request=0;
maneuver=1;

[!good]
[standby]

[ supported && good]

[ APfailure]

[ support && good]

FAULT
good=0;
Two Independently Designed Systems Can Often Create Problems
QE-V&V Detects This Anomalous Behavior And Allows Proactive System Design
Path Forward

• LM Aero QE-VV Capability Nearing Point of Testing Realistic Model Examples
  • Established Tool Compatibility with Simulink Primitive Components
  • Models include Nonlinear Algorithms, States, Lookup Tables, and Complex Signal Types
• Focus on Challenge of Optimizing Temporal Unrolling and Property Proving of Models with Multiple States
• Sensitivity Studies to Be Conducted on
  • Probability of Determining All Ground States \(\rightarrow\) V&V Confidence
  • Benchmark Data on QE-VV Method with DW2 in the Loop

Please Contact Us!
TODD BELOTE, todd.r.belote@lmco.com, 817-935-3585
CHRIS ELLIOTT, christopher.m.elliott@lmco.com, 817-935-3054
filename = act\_lin\act\_lin-176.ham, #nodes = 104