1. Dynamic Continuous Dataflows and QoS Metrics - Background

**Dynamic Dataflow:**
- Is a dataflow consisting of a number of **dynamic tasks (or paths)** connected using dataflow edge dependencies that continuously processing the incoming data streams and produces.
- It allows flexible re-configuration by allowing dynamic updates to the individual task logic or by updating an executing sub-flow.
- Extremely **high and variable data rates**

**Dynamic Tasks:**
- A set of **alternate implementations** (alternates) for a given logical activity.
- Each alternate exhibits different "Cost" to "Value" ratio.
- **Cost** for each alternate is measured in terms of "Core-Seconds" required to perform one **Unit operation** on a reference CPU core.
- **Value** is measured **relatively** among the alternates of a given Dynamic Task based on a domain specific utility function (Domain Perceived Utility, e.g. Accuracy of Results)

**Application QoS:**
- Relative Application Throughput ($\Omega$): Ratio of observed application throughput to maximum achievable throughput given data rate $r_t$
- **Overall Application Value**: Alternate values aggregated for the application based on the alternate active during that interval

$$\Omega(t) = \frac{\alpha(t)}{\mu(t)}$$

3. Constraint-Driven Adaptive Scheduling (Problem Statement)

**Given:** Optimization Interval, $T$. Monitoring interval $t$.

**Goal:** Select **Active Alternates** and **Resource Mapping** during each interval $t$ so as to **Maximize** overall profit $\Theta$ over $T$

**Constraint:** Average relative throughput $\Omega > \hat{\Omega}$

4. Reactive Scheduling Strategies (Proposed Solutions)

Two Step Process:
1. Get current "representative" resource performance and data rates for next interval
2. Given the representatives, find the "best" resource mapping for the next interval

**Representative Selection Heuristics**

- **Current After-the-fact (CAF)**
- **Windowed Average After-the-fact (WAFF)**
- **Windowed Predictive Average (WPA)**
- **Windowed Predictive Optimistic (WPO)**
- **Windowed Predictive Pessimistic (WPP)**

**Resource Mapping Heuristics**

- **Variable Sized Bin-Packing**
- Genetic Algorithm

5. Predictive Look-Ahead Scheduling (PLASiCC)

**Main Idea:** Find the Largest Interval where the current resource mapping violates the constraint, create a **continuously refined** look-ahead plan with **Late Execution** model

**Example**

(a) Sample Dataflow

(b) Data Rate and VM Performance Predictions

(c) PE Resource Mapping at time $t$

(d) PLASiCC Iterations, Planned Actions and Cumulative $\Theta$

6. Reactive VS PLASiCC VS Genetic Algorithm – Observed Results

**Conclusions:**
- Dynamic Dataflows provide powerful abstractions and flexible runtime adaptation.
- Reactive Scheduling Techniques address Infrastructure Performance and Data Rate variations. But are prone to triggering.
- Predictive Look前瞻 Scheduling generates an evolving plan and mitigates triggering.
- Genetic Algorithms provide near optimal solutions but have high run-time overheads, and hence are suitable only for initial deployment