CS 516 Compilers and Programming Languages II

Redundancy-2
Project #2 will be posted over the weekend (Sunday night)

Tuesday (next) lecture:
   → Shaleen will talk about polyhedral optimizations
   → Phillip will present work on intermittent energy optimizations

Papers are posted under sakai/resources
• SGS graduate students enrolled in graduate level classes (16:xxx:500 and above) or in undergraduate courses that are approved parts of their graduate curriculum have the option to switch their grading modality from A, B, C, F letter grades to P/NC for any Spring 2020 class through May 22, 2020. The “P” grade replaces the A, B, C letter grades. The “NC” grade replaces the F letter grade.
• P/NC grades will be allowed for the Spring 2020 semester only.
• “P” grades will count towards the credits required for the degree, but they will not enter into GPA calculation.

The information listed above only partially covers issues related to P/NC, or dropping a course. Please reach out to your advisor if you have any questions or concerns.
The RSDG is a dually weighted, directed graph. The weights are cost (\( C \) - resource consumption) and mission value (\( Q \) - quality). For simplicity, we assume that all cost/quality models are linear.

Node and edge are associated with a cost \( C \) (here energy). No label means 0 cost.

Node mission values and execution budgets are provided by the user. Mission values may only be given to user-critical services (nodes).

**Basic structure of 0-1 problem:**

\[
\begin{align*}
\text{maximize} & \quad \sum_{i} \sum_{j} s_{ij} Q_{ij} \\
\text{subject to} & \quad \sum_{j=1}^{n} s_{ij} \leq 1 \\
& \quad \sum_{i} \sum_{j} C_{ij} < \text{Budget} \\
\text{foreach} & \quad \text{Edge}_{\text{AND}}(s_{0} \rightarrow s_{i}), s_{i} - s_{0} \leq 0 \\
\text{foreach} & \quad \text{Edge}_{\text{OR}}(s_{0}, s_{1}, s_{2}, \ldots s_{n} \rightarrow s_{i}), s_{i} - \sum_{i} s_{0} \leq 0
\end{align*}
\]
Example solutions for different provided resource budgets.

**Budget = 29**

- Configuration: \( \{S^1_{1A}, S^1_3, S^2_2, S^1_5\} \)
  - Overall quality = 7
  - Overall energy cost = 29

**Budget = 31**

- Configuration: \( \{S^2_1, S^2_3, S^1_2, S^1_{4B}\} \)
  - Overall quality = 9
  - Overall energy cost = 31
Localization with five quality levels
- Two services with implementation redundancy
- One cloned service

cost and quality weights are not shown
implementation replication / cloning
Service failure inhibits services to be selected, and all other services that rely on the failed services.

Example: WiFi, Camera 2, and IMU 1 fail

⇒ limited configuration space
Possible metrics that describe the redundancy/robustness of a system

(1) Given a set of **failed services** FS (single and multiple failed services):
   • Reduction in configuration space size due to FS
   • Reduction in configuration selections of user critical services due to FS
   • Reduction in available approximation levels for user critical services

(2) **Sensitivity analysis to budget/resource reductions**: Similar metrics can be used to assess the impact of budget reductions

(3) **Impact of failures / limited budgets on quality of optimal configuration.** Enable “what if” experiments

⇒ **Metrics can be used to assess and guide the design of robust/redundant systems**
Basic RAPIDS structure

Possible use of RSDG in a development / execution environment
Basic RAPIDS structure

Possible use of RSDG in a development / execution environment

Ferret from PARSEC benchmark suite (Princeton U.)

3 knobs/services: Labels are value settings, not cost or quality weights

<Continuous Knobs>:

<Discrete Knobs>:
hash {2,4,8} INT;

<Dependencies>:

<Sub-Metrics>:
Coverage, Ranking
Basic RAPIDS structure

**User**

Specifies **quality preferences** and **resource budget** (maximal execution time - soft deadline)

**Framework Runtime**

- Monitors application progress, resource consumption, failures
- Remapping solved as an integer programming problem (**gurobi**)
- Remapping triggered when
  + mismatch of predicted resource usage to actual observed usage beyond tolerance threshold. Possible reasons:
    - errors in prediction model
    - unforeseen change in execution/resource environment
    - system component failures
    - low resource utilization (observed in fixed time intervals)

UI prototype for Ferret
Sample video application execution with custom quality under a fixed budget

Dynamic reconfiguration

- Full Power: Finished 97.53%
- Prefer FrameRate: Finished 98.34%
- Prefer Resolution: Finished 98.21%
- Prefer Brightness: Finished 81.16%

Time (Second)

Frame rate: high ... low
Resolution: high ... low
Brightness: high ... low
Project #2

**Developer**

- **RSDG specification**
  - parse.py
  - Weighted RSDG in xml format
  - quality and cost weights

**User**

- **Execution time budget**
  - (command line argument)

**C++ program**

- Read in RSDG
- Read in budget
- Generate 0-1 problem

**Project Code to be written by students**

- **0-1 integer program RSDG.lp**
- **Gurobi solver**
- **Optimal configuration**
Project #2 - Sample RSDG

Unweighted RSDG:

Maximize
20 S1.4 + 40 S1.3 + 10 S1.2 + 5 S1.1 +
20 S2.2 + 10 S2.1

Subject To
(cost: 50 S1.4 + 40 S1.3 + 21 S1.2 + 8 S1.1 +
25 S2.2 + 20 S2.1 <= 70)

S1.Select: S1.1 + S1.2 + S1.3 + S1.4 = 1
S2.Select: S2.1 + S2.2 = 1

Binary
S1.1 S1.2 S1.3 S1.4 S2.1 S2.2

End

Objective value = 35
S1.3 = 1
S2.2 = 1