Cornell University

CSL

Prediction-Guided Performance-Energy Trade-off for Interactive Applications

Daniel Lo

Taejoon Song, G. Edward Suh

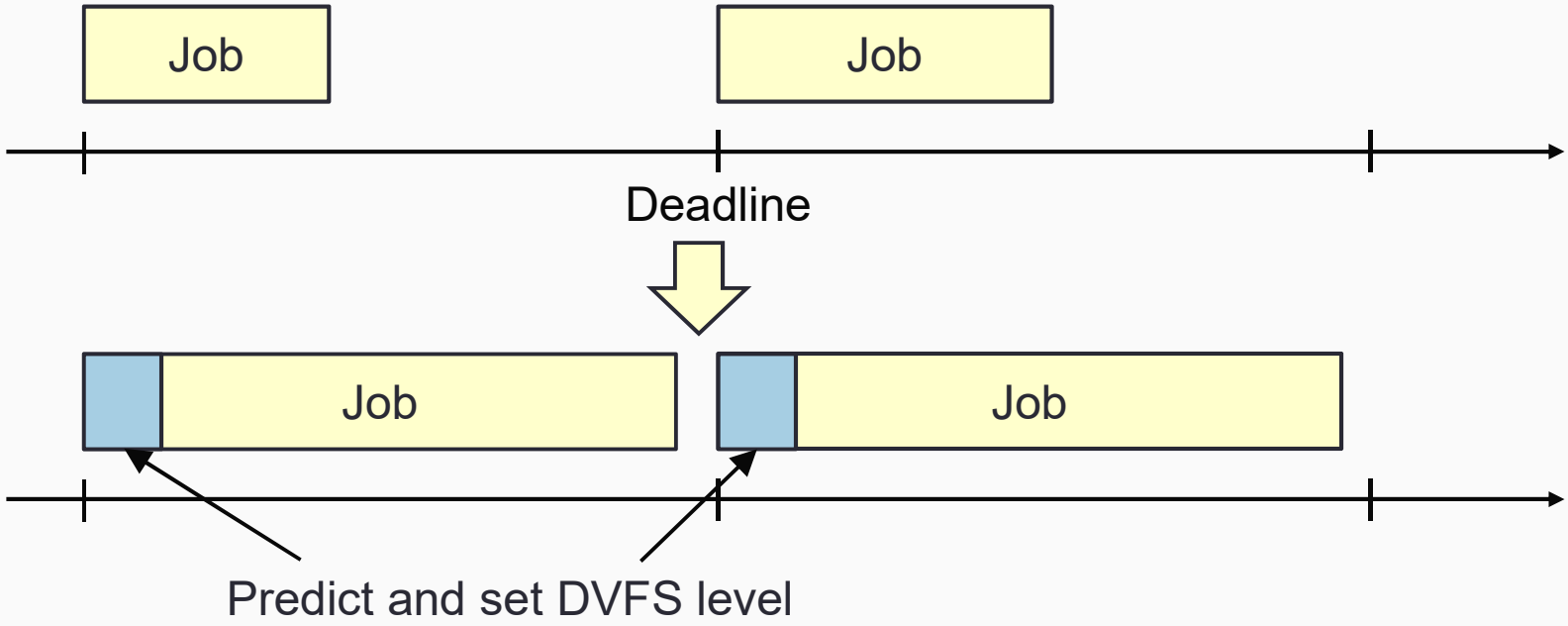
Cornell University

Interactive Applications

- Many modern applications are highly interactive
  - Web browsing
  - Games
  - User interfaces
- These interactions have response-time requirements
- Finishing faster does not improve user experience

Prediction-based DVFS Control

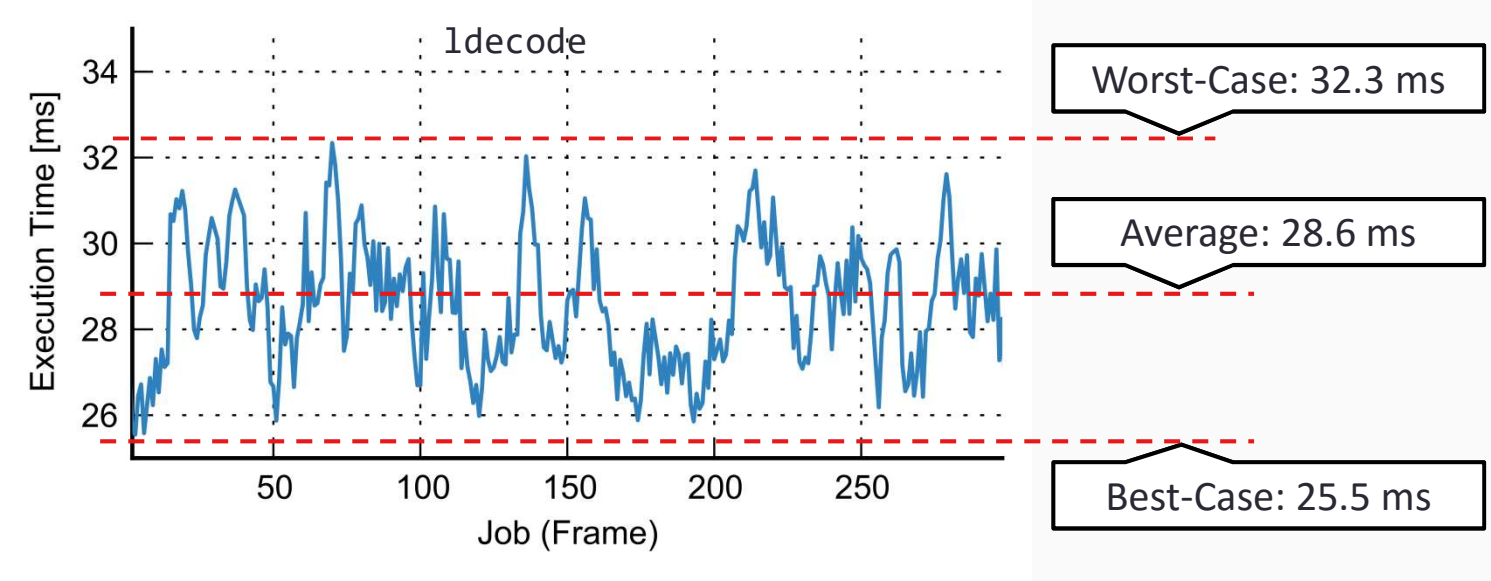
- Run jobs slower in order to save energy while preserving user experience



- Goal: Pick DVFS level for each job to minimize energy and meet response-time requirement

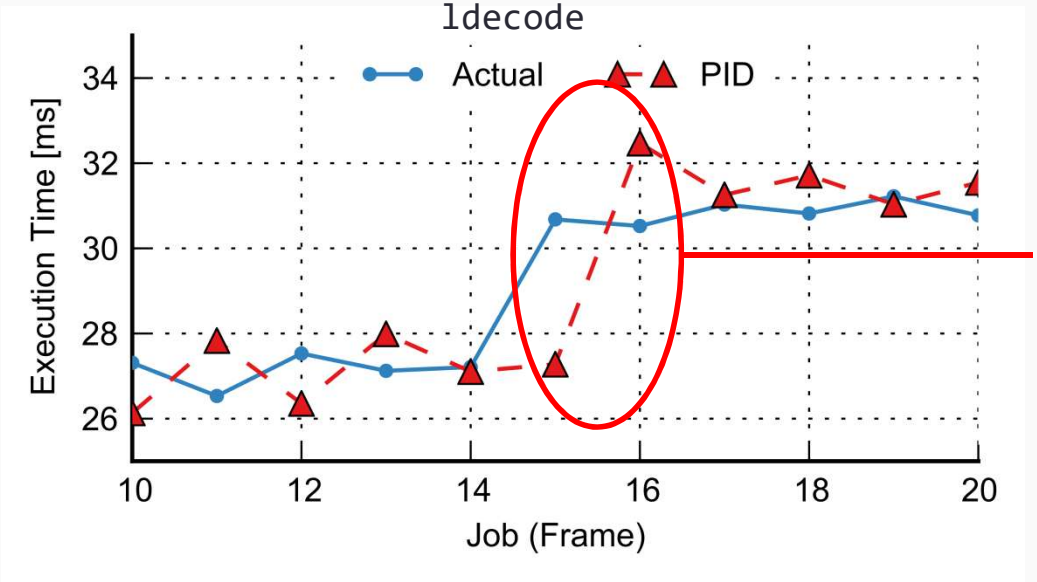
Execution Time Variation

- Challenge: Execution times for an application vary from job to job
  - Optimizing for the worst-case wastes energy
  - Optimizing for the average-case misses deadlines



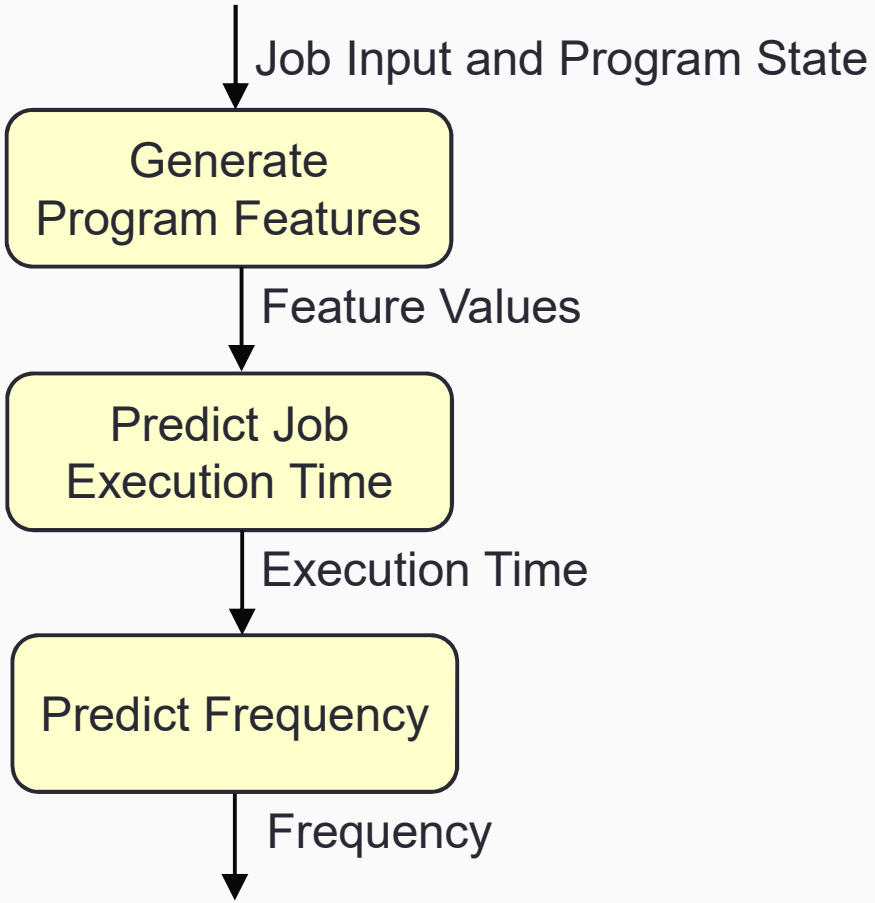
Design Objectives

- Proactive control – History-based DVFS control is too slow to account for variations
  - Predict DVFS level based on job inputs and program state



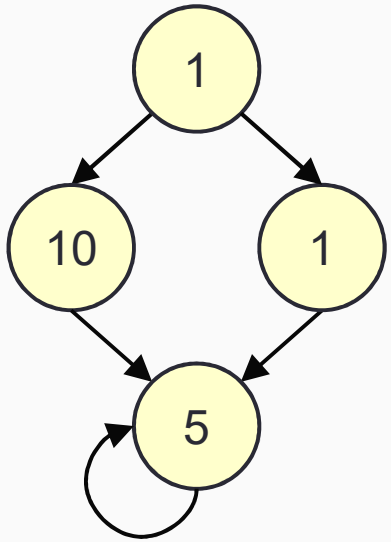
- General – applicable to a wide range of applications
- Automated – minimal programmer effort

Prediction Approach



Feature Selection

- To first order, execution time correlates with number of instructions executed
- Number of instructions depends on control flow
  - Conditionals taken/not taken
  - Loop counts
  - Function calls



Instrumentation

- Instrument program source to count these features

	Original Code	Instrumented Code

Conditional	`if (condition) {`
	`  …`
	`}`
		`if (condition) {`
		`  feature[0]++;`
		`  …`
		`}`

Loop	`for (i=0; i<n; i++)`
	`{`
	`  …`
	`}`
	
	`while (n = n->next)`
	`{`
	`  …`
	`}`
		`feature[1] += n;`
		`for (i=0; i<n; i++)`
		`{`
		`  …`
		`}`
		
		`while (n = n->next)`
		`{`
		`  feature[2]++;`
		`  …`
		`}`

Prediction Slice

- Problem: Instrumented job will take as long as original job to run
- Solution: Use program slicing to create minimal job to calculate features

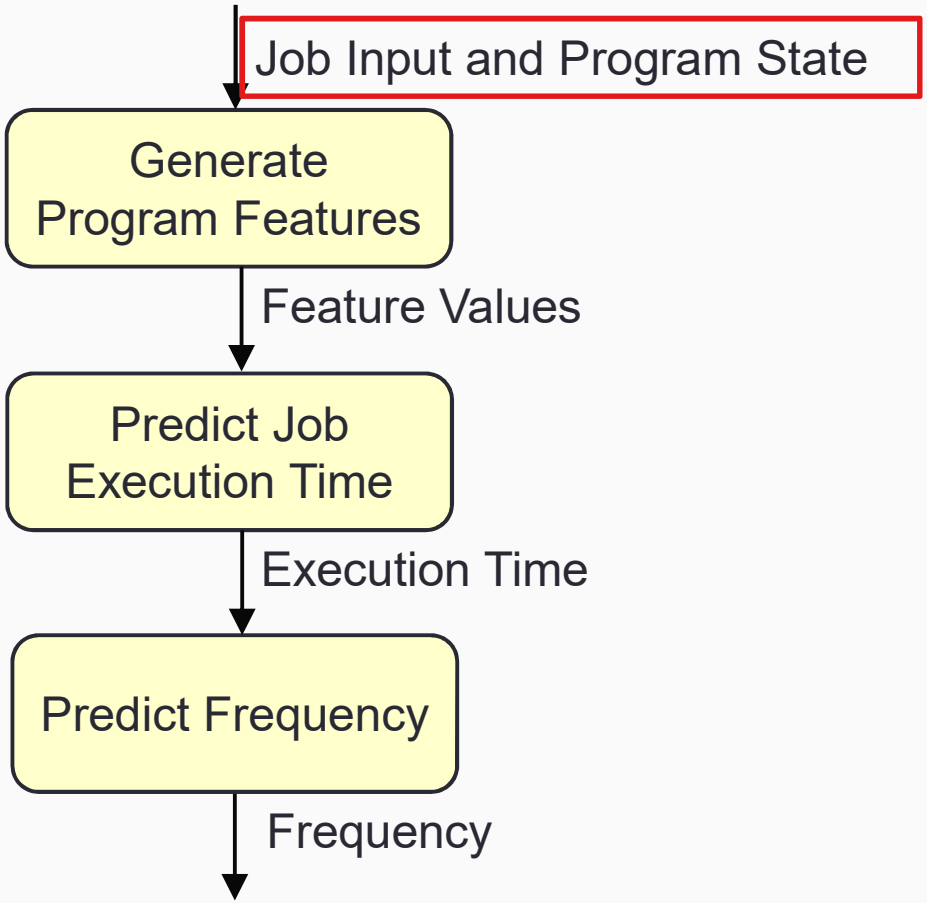
Instrument Code

```
if (x)
{
  feature[0]++;
  compute();
}

feature[1] += n;
for (i=0; i<n; i++)
{
  compute2();
}
```

Program Slice

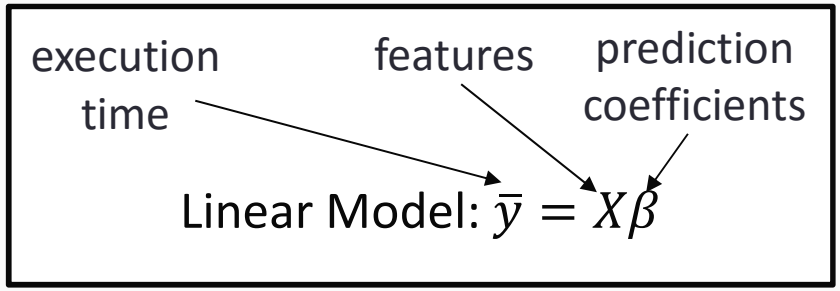
```
if (x)
{
  feature[0]++;
}


feature[1] += n;
```

Prediction Approach



Execution Time Model

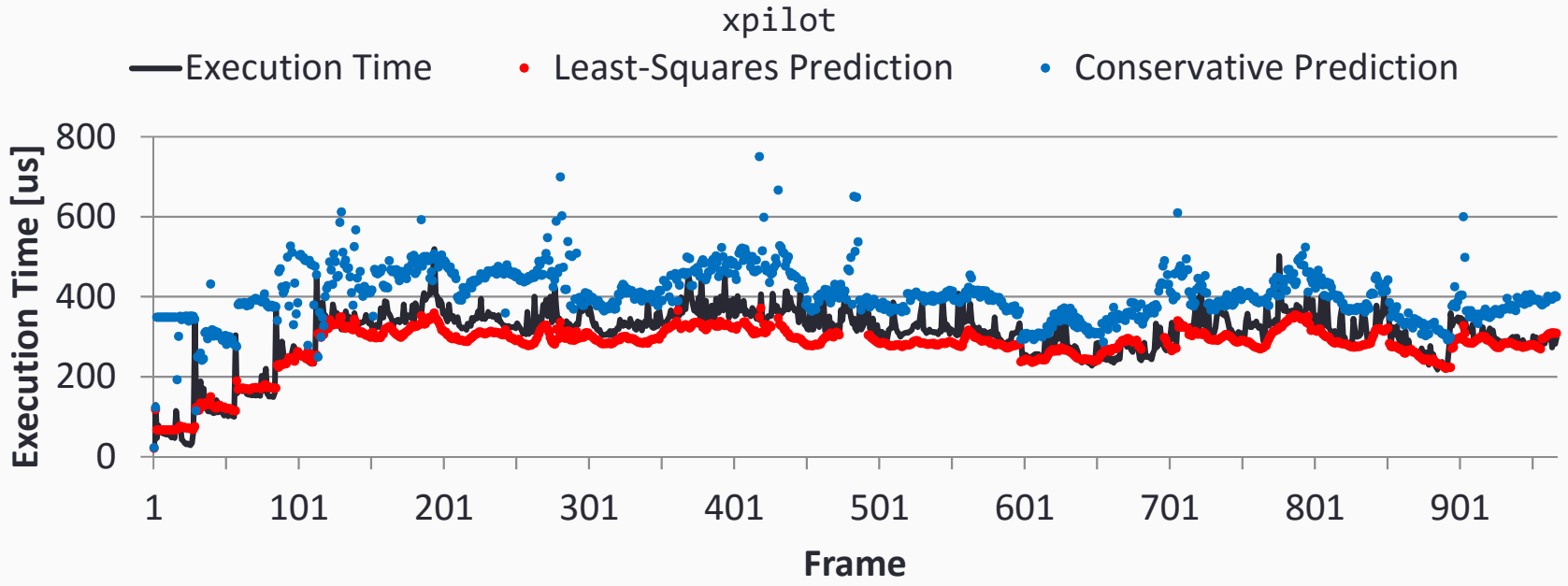
- Linear model to map features to execution time
  - Captures first-order effect of control flow on execution time
  - Fast to evaluate



- Use profiling data to fit model coefficients
  - Multiple possible ways to calculate coefficients

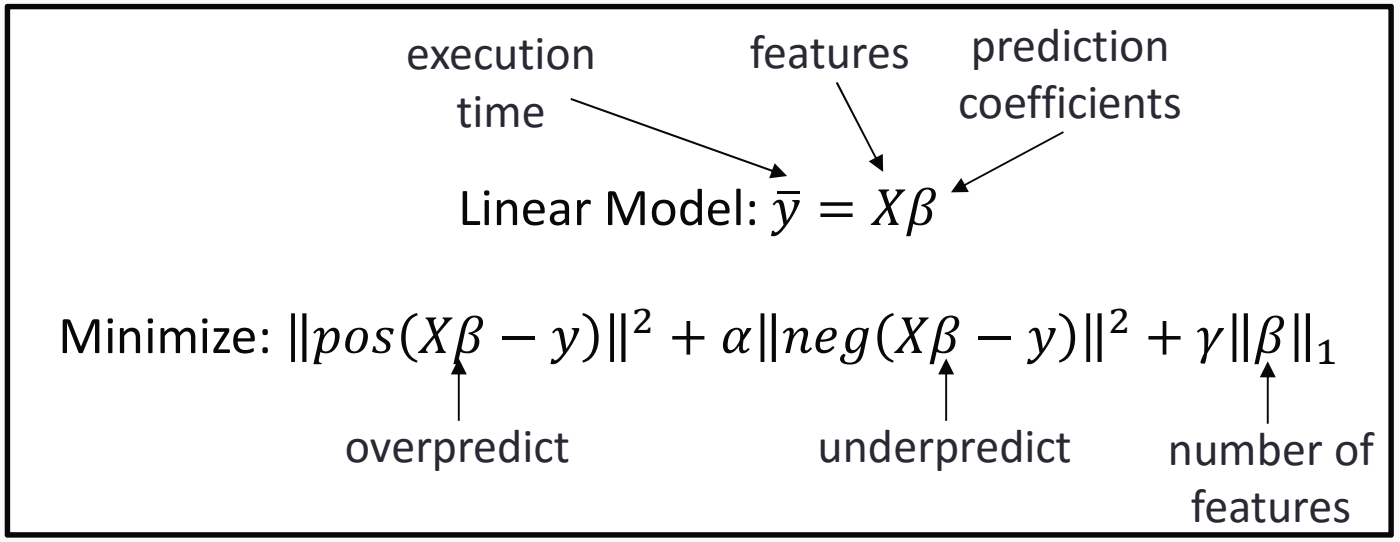
Conservative Prediction

- Least-squares prediction leads to both under-prediction and over-prediction
  - Under-prediction results in deadline misses
- Want to skew prediction towards over-prediction

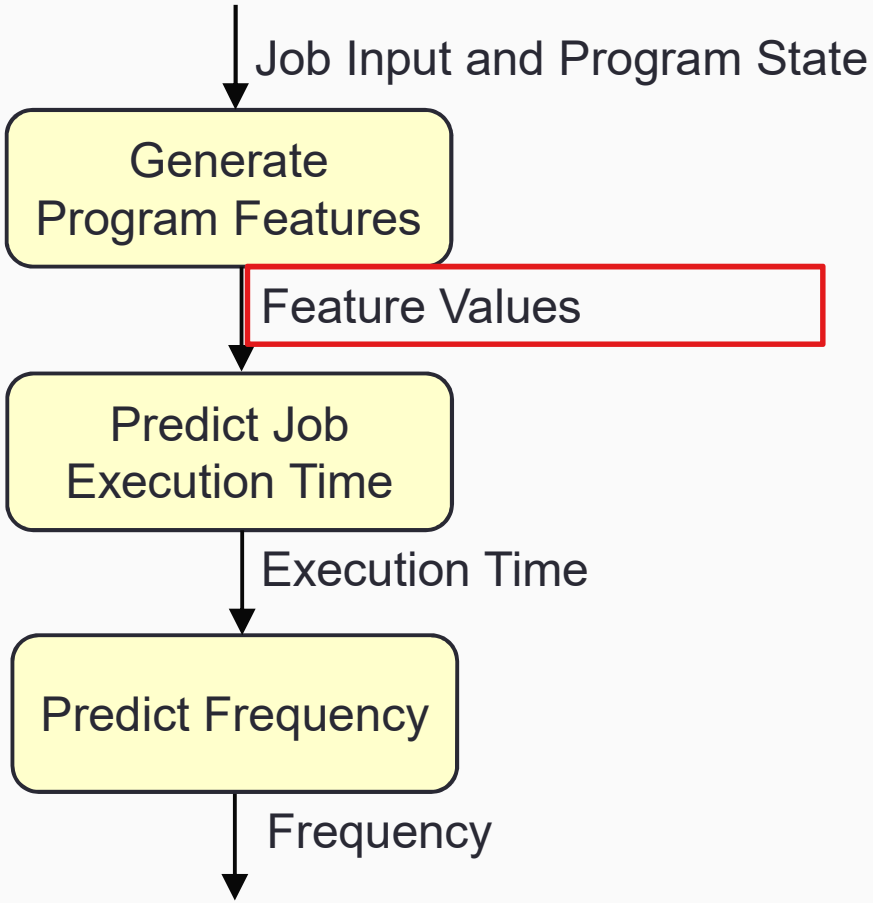


Prediction Model

- Use convex optimization with custom objective
  - Avoid under-prediction – leads to missed deadline
  - Minimize features needed



Prediction Approach

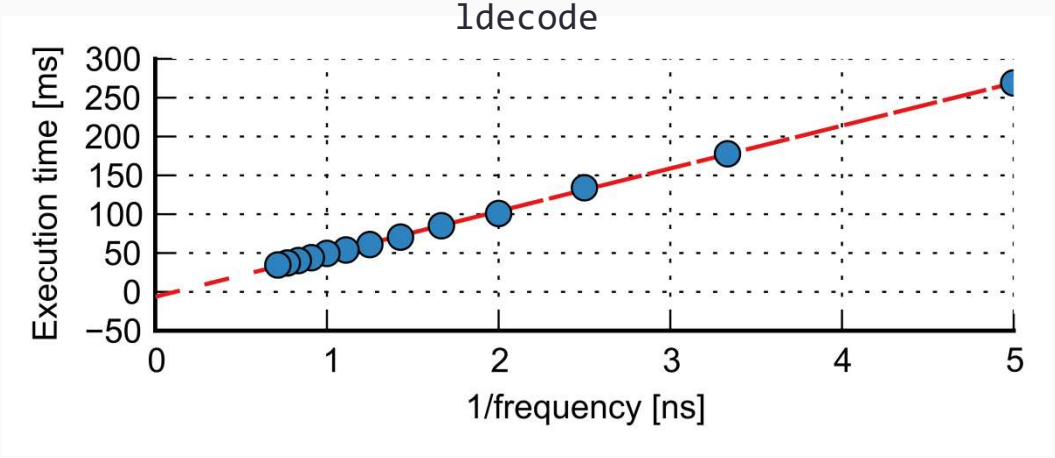


DVFS Model

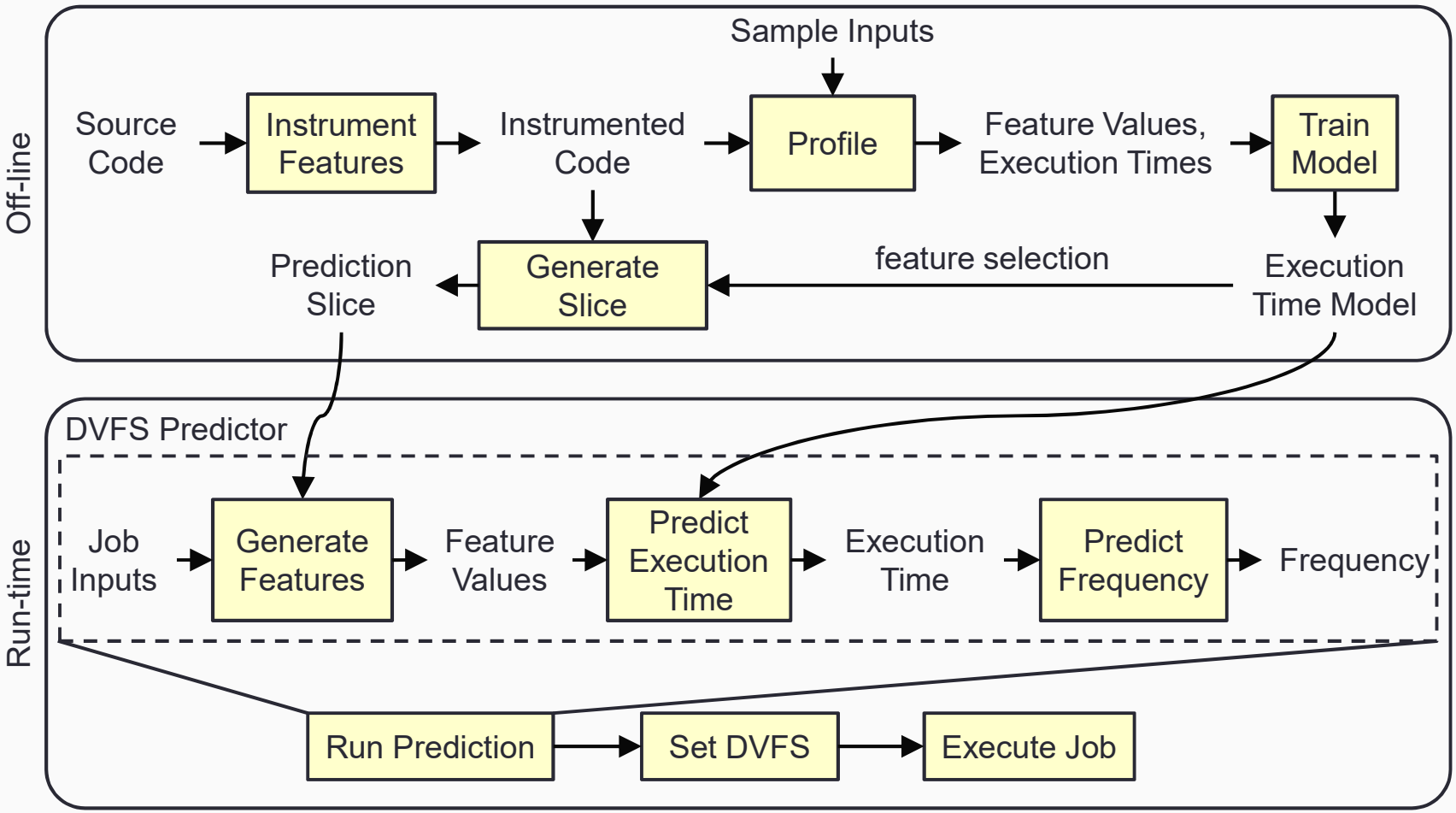
- Need a model of how execution time scales with frequency
  - Simple linear model

$$t = T_{mem} + N_{dependent}/f$$

- Empirical data shows that this is reasonable
  - $r^2 > 0.99$ for all benchmarks



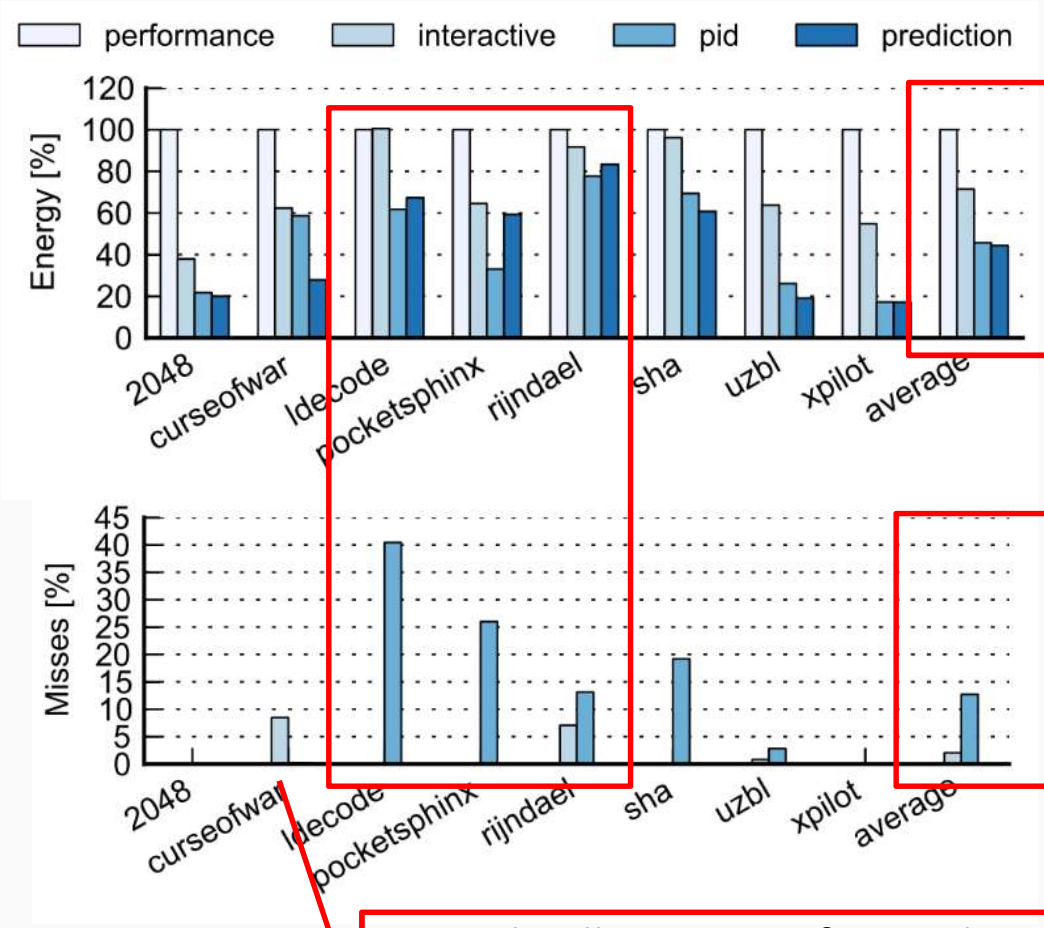
Full Approach



Evaluation Setup

- ODROID-XU3 development board
  - ARM Cortex-A7 core (100 MHz steps from 200 MHz to 1.4 GHz)
  - Ubuntu 14.04
- Baseline controllers
  - performance – always max frequency
  - interactive – Linux governor based on utilization
  - pid – PID-based controller
- Benchmarks
  - web browser (uzbl)
  - speech recognition (pocketsphinx)
  - video decoding (ldecode)
  - games (2048, curseofwar, xpilot)
  - MiBench kernels (rijndael, sha)

Results



prediction energy savings
- vs. performance: 65%
- vs. interactive: 38%
- vs. PID: 2%

2% deadline misses for interactive
13% deadline misses for PID

0.1% deadline misses for prediction

Conclusion

- Opportunities exist to improve the energy usage of interactive applications
- History-based DVFS controllers are not quick enough to respond to job-to-job variations in execution time
- Prediction-based DVFS control outperforms traditional power governors and PID-based controllers
  - Lower energy usage
  - Almost no deadline misses