Optimality and Improvement of Dynamic Voltage Scaling Algorithms for Multimedia Applications

Prof. Lei He, UCLA, EE Dept
LHE@ee.ucla.edu
http://eda.ee.ucla.edu
Research Overview

Mixed-signal SoC, SiP and 3D
FPGA
Multi-core HW & SW

Signal Integrity
Power and Thermal
Variations and Resilience

Since 2006
- Best paper ISPD06
- 6+3 Best paper nominations at DAC/ICCAD, 1 at ASPDAC, and 1 at CICC
- Two assistant professors in 2010
Numerical Group

- Modeling and simulation
  - Extraction of RLC and thermal parasitics
  - Extraction of PTV variations
  - Macro and behavior modeling and figures of merit for delay, SI/PI, and mismatch
  - SPICE with variation
  - Parallel algorithms

- Mathematical programming for optimization
  - Linear, convex, and nonlinear

- Applications:
  - analog circuits
  - Energy for hybrid cars and green computing
Combinatory and Systems Group

- Placement and routing for SiP and 3D
- Fault/SER tolerance logic (FPGA) and system
- Low power circuits and computer architecture
- ESL for RF/MM based multimedia systems
- Embedded/wearable systems for medical care
- Parallel data mining for cloudy computing
**Outline**

- **Background**
- Optimal Offline Solution
- Effective Online Algorithm
- Simulations and Results
- Conclusions
Proliferation of Multimedia Applications

videoconferencing, emergency services, surveillance, tele-medicine, augmented reality, and distributed gaming, etc.

Wireless communication
Need and Challenge of Power Management

- Multimedia applications are energy sensitive
  - Computationally demanding
  - Stringent deadlines for multiple tasks
  - Processed on energy-limited devices

- DVS (dynamic voltage scaling) algorithms to reduce energy while meeting deadlines

- DVS algorithms need to deal with uncertainty in
  - Job complexity (time-varying workloads)
  - Communication delay (e.g. wireless channel)
Formulation of DVS Problem

- **Given**
  - A sequence of decoding jobs (stochastic complexity, stochastic arrival time, deterministic deadline).
  - A set of voltages including power gating, each with associated clock frequency and power.

- **Find**
  - The time and voltage level for each voltage switch.

- **Minimize**
  - Energy.

- **Subject to**
  - Start a job after it arrives.
  - Finish a job before its deadline.
Existing Online Algorithms

- Uses worst or average case execution time [Pillai ACM Symposium on OS’01] [Choi, ICCAD’02] [Zhu LCTES’07] [Nahrstedt et al., ITMC’06]
  - Ensures hard deadlines not missed
  - Soft deadlines (and slack reclamation) to reduce energy consumption

- Online workload prediction
  - Feedback-based [Zhu LCTES’07]
  - Adaptive linear prediction [Akyol 2007]
  - Buffer-constrained DVS [Maxiaguine et al, ICHSC’05]
  - Stochastic Queuing-based DVS [Foo 2008]

It’s not clear how far online algorithms are away from the optimal solution
Existing Offline Algorithms

- Offline algorithms can be used for optimality study
  - Optimal solution assuming that complexity and arrival time are known based on trace
  - Lower bound of energy for online algorithms
- Existing ILP-based offline algorithms are not scalable
  - [Akyol, 2007] [Zhang, ICCAD’07]
Contributions

- Efficient LP-based (instead of ILP-based) optimal offline DVS algorithm.
  - Enables optimality study of online DVS algorithms.

- Effective online approach by sequential linear programming, namely SLP/r.
  - Consumes 0.3% more energy versus 4% more energy for the best existing work, when both compared with optimal solution.

- Applicable to other delay-sensitive applications with time-varying workloads.
  - e.g. real-time stream queries for financial or medical data and manufacturing process control.
Outline

- Background
- **Optimal Offline Solution**
- Effective Online Algorithm
- Simulations and Results
- Conclusions
DVS Problem in Time – Complexity Space

- **Deadline of each job**
- **Arrival time of each job**
- **Total Complexity**
- **Transmission time**
- **C₁**
- **Complexity of each job**
- **T₁, T₂, T₃**
- **D₁, D₂, D₃**
- **Deadline of each job**
- **Max computation that needs to be done by time t**
- **Minimal computation that needs to be done by time t**

\[
L(t) \quad U(t)
\]
valid scheduling solution: piecewise linear curve between $U(t)$ and $L(t)$

time for voltage switch

slope $\Leftrightarrow$ clock frequency
Compared to multimedia jobs (around $10^9$ clock cycles), voltage switching overhead (around 10 clock cycles) is negligible.

We call time interval with constant $U(t)$ and $L(t)$ as the adaptation interval.

Primary Theorem: for an adaptation interval, an arbitrary ordering of any accumulative computation curve consumes the same energy.

- What really matters is the percentage of time for each voltage level.

![Diagram](Image)
LP Formulation for DVS

\[
\min E = \sum_{i=1}^{W} \sum_{j=0}^{K} (A_{ij} \cdot P_j \cdot \varphi_i)
\]

s. t.

\[
L(I_n) \leq \sum_{i=1}^{n} \sum_{j=0}^{K} (F_j \cdot A_{ij} \cdot \varphi_i) \leq U(I_n), \forall 1 \leq n \leq W
\]

\[
0 \leq A_{ij} \leq 1, \text{ for } 0 \leq j \leq K \text{ and } \sum_{j=0}^{K} A_{ij} = 1
\]
Outline

- Background
- Optimal Offline Solution
- **Effective Online Algorithm**
- Simulations and Results
- Conclusions
rLP Formulation

\[
\min \ E = \sum_{i=1}^{W} \sum_{j=0}^{K} (A_{ij} \cdot P_j \cdot q_i)
\]

s. t.

\[
0 \leq A_{ij} \leq 1, \text{ for } 0 \leq j \leq K \text{ and } \sum_{i=0}^{K} A_{ij} = 1
\]

\[
L(I_n) \leq \sum_{i=1}^{n} \sum_{j=0}^{K} (F_j \cdot A_{ij} \cdot q_i) \leq U(I_n) \quad \forall \ 1 \leq n \leq W
\]

Difference from offline formulation

- \( L(t) \) and \( U(t) \) become stochastic
- \( L(t) \) and \( U(t) \) depend on stochastic complexity of jobs
- A sequence of rLP for a sequence of time windows, each window is bigger than an adaptation interval
Model for Stochastic Complexity of Jobs

- Class-based stochastic model [Foo, 2008]
  - A job class is a particular GOP frame type
  - Near Gaussian distributed complexity
  - Parameters derived offline and transmitted online with low cost

Illustration of SLP/r

- Convert rLP to SLP for robustness

Prediction

U(t) for robust linear programming

L(t) for robust linear programming
Illustration of SLP/r

Prediction $\rightarrow$ rLP

U(t) for robust linear programming

L(t) for robust linear programming

$T_1'$ $T_2'$ $T_3'$ $D_1$ $D_2$ $D_3$ Media Time
Illustration of SLP/r

Prediction $\rightarrow$ rLP $\rightarrow$ Commitment

$L(t)$ for robust linear programming

$U(t)$
real complexity
and arrive time of jobs

$T_1'$ $T_2'$ $T_3'$ $D_1$ $D_2$ $D_3$ Media Time
Illustration of SLP/r

Prediction ➔ rLP ➔ Commitment

U(t)
real complexity
and arrive time of jobs

L(t) for robust linear programming

T1' T2' T3' D1 D2 D3 Media Time
Illustration of SLP/r

Prediction $\rightarrow$ rLP $\rightarrow$ Commitment

Process a new time window

$U(t)$
real complexity
and arrive time of jobs

$L(t)$ for robust linear programming

Media Time $D_1$, $D_2$, $D_3$
Illustration of SLP/r

Prediction $\rightarrow$ rLP $\rightarrow$ Commitment

Process a new time window

$U(t)$
real complexity
and arrive time of jobs

$L(t)$ for robust linear programming

D1, D2, D3, Media Time
Outline

- Background
- Optimal Offline Solution
- Effective Online Algorithm
- Simulations and Results
- Conclusions
Experimental Setup

- $V_{dd}$ between 0.6V and 1.0V with step sizes of 0.1V, plus power gating
  - Our algorithms are applicable to any power model.
- Video sequence consisting of 10 different scenes.
- Compare SLP/r with:
  - Queuing-Based Stochastic Algorithm [Foo, 2008]
  - Deterministic laEDF [Pillai, 2001]
- Monte Carlo simulation of stochastic complexity and arrival time to verify all results
Recap of SLP/r

- Confidence level
  - Linear online prediction function for workload of each job class: \( \text{mean} + k \times \text{standard deviation} \)
  - Deciding trade-off between miss rate and energy

- Granularity of SLP/r
  - Number of jobs to commit before shifting the window
  - Deciding tradeoff between runtime and quality of solution
Comparison of Energy/Miss Rate

- Granularity = 1 job

Optimality study:
- laEDF: 15% more energy.
- Queuing-based: 4% more.

Online algorithm SLP/r
- 1% more energy
Granularity VS Quality

Changing granularity from 1 to 4 jobs

We reduce runtime, energy and miss rate simultaneously

Granularity = 4 jobs:

0.03% miss rate with 0.3% more energy than optimal
Energy VS Granularity/Confidence Level

Minimum energy setting:
Granularity = 4 jobs
Confidence level = 1.5
Granularity = 4 jobs
Confidence level = 1.5,
Miss rate close to zero
Outline

- Background
- Optimal Offline Solution
- Effective Online Algorithm
- Simulations and Results
- Conclusions
Conclusions and Future Work

- An efficient optimal offline DVS algorithm based a tractable LP formulation.
  - Enables optimality study for DVS algorithms.

- An effective online approximation to SLP/r by sequential robust linear programming.
  - Consumes 0.3% more energy versus 4% for the best existing work, when both compared with optimal solution.

- In the future, apply to other delay-sensitive applications with time-varying workloads.
Thanks!

Q & A