Tradeoff exploration between reliability, power consumption, and execution time

Ismail Assayad, Alain Girault, and Hamoudi Kalla

DREAM seminar, UC Berkeley
1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
Outline

1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
Problem and motivations

Problem statement

Statically schedule an application task graph onto a homogeneous distributed memory architecture, with a maximal reliability, a minimal power consumption, and a minimal length / WCET / $c_{max}$ / makespan

- Three antagonistic criteria: reliability, power consumption, and length
- Reliability is crucial for dependable systems
- Length is crucial for real-time systems
- Power consumption is crucial for autonomous systems
- Plenty of industrial applications: satellite, automotive, consumer electronics, low-power, ...

Algorithm task graph
Application, distributed architecture, and static schedule

Algorithm task graph

Distributed architecture graph
Application, distributed architecture, and static schedule

Algorithm task graph

Distributed architecture graph

A (partial) static distributed schedule
Application, distributed architecture, and static schedule

Algorithm task graph

Distributed architecture graph

A (partial) static distributed schedule

One point in the 3D space
Reliability model

Definition

The reliability measures the service continuity Probability that the system functions correctly during a given time interval.

Reliability model [Lloyd & Lipow, 1962] [Shatz & Wang, IEEE TR’89]

\[ R(X, P) = e^{-\lambda_P L(X, P)} \]

- \( \lambda_P \) is the failure rate of component \( P \) per time unit
- \( L(X, P) \) is the length / WCET of operation \( X \) onto \( P \)
- All the HW components are fail-silent
- All the failures are transient (implies the “hot” failure model)
- All the failure occurrences are statistically independent events
Energy and power consumption model

Model taken from [Zhu, Melhem, Mossé & Elnohazy, ICPADS’04]

\[ P = P_s + h(P_{ind} + P_d) \]
\[ P_d = C_{ef} V^2 f \]

- \( P_s \) is the static power
- \( h = 1 \) (circuit active) or 0 (inactive)
- \( P_{ind} \) is the frequency independent active power
- \( P_d \) is the frequency dependent active power
- \( C_{ef} \) is the switch capacitance
- \( V \) is the supply voltage
- \( f \) is the operating frequency

Same model for communications [Luo, Peh & Jha, DATE’03]
Outline

1. Context and overview

2. Multicriteria optimization for power and reliability

3. Power reduction techniques

4. Reliability improvement techniques

5. Our tricriteria scheduling algorithm TSH

6. Simulations

7. Conclusion
Intuition: our criteria are antagonistic

- Replicating operations **improves the reliability but increases the schedule length**

- Lowering the operating voltage **reduces the power consumption but increases the execution times, and thus the schedule length**

- Lowering the operating voltage **reduces the power consumption but increases the failure rate per time unit, and thus reduces the reliability**

---

**How to understand the tradeoffs?**

Study the non-dominated solutions, aka the Pareto optima.
Pareto optima

Assume a simple bicriteria minimization problem with $Z_1$ and $Z_2$

- $x^1$, $x^2$, $x^3$, $x^4$, and $x^5$ are Pareto optima
- $x^1$ and $x^5$ are weak optima
- $x^2$, $x^3$, and $x^4$ are strong optima
- The set of all Pareto optima is the Pareto curve
The end-user must have the choice!

- Solutions to our problems are static schedules that can be plotted in the 3D space (length, reliability, power)

- Only the end-user can choose between the solution (300, 0.999, 1.0) and the solution (200, 0.99, 2.0)

- We must provide him/her with a set of non-dominated solutions

  ➤ The Pareto surface in this 3D space

**First principle**

We must provide the Pareto surface.
How to compute the Pareto surface?

Usual approaches to multicriteria optimization [T’kindt & Billaut, 2006]:

1. Aggregation of the criteria into a single one
   - transform the problem into a classical single criterion optimization problem.

2. Hierarchization of the criteria
   - optimize one criteria at a time.

3. Transformation of \( n - 1 \) criteria into constraints
   - find the solution that optimizes the \( n \)-th criterion among all the solutions that satisfy the \( n - 1 \) constraints.
Aggregation method

Aggregation of the criteria into a single one
transform the problem into a classical single criterion optimization problem.

\[ \alpha Z_1 + \beta Z_2 \]
Hierarchization method

Hierarchization of the criteria

- optimize one criteria at a time.

\[ Z_2 \]

1: minimize \( Z_2 \)

2: minimize \( Z_1 \)
How to compute the Pareto surface?

1. Aggregation of the criteria into a single one
   - DOES NOT WORK

2. Hierarchization of the criteria
   - DOES NOT WORK

3. Transformation of $n - 1$ criteria into constraints
   - ONLY CHOICE — IT WORKS

Second principle
To compute the Pareto surface, we must transform two criteria into constraints and minimize the third criterion.
Building the Pareto surface incrementally (1)

- In two dimensions: \( \varepsilon \)-constraint method [Haines, Lasdon & Wismer, TSMC’71]
- Generalization to \( n \) dimensions: [Laumanns, Thiele & Zitzler, EJOR’06]
- SAT-solving based method: [Legriel, Le Guernic, Cotton & Maler, TACAS’10]

Here, we just generate a regular grid (anyway our mono-criterion minimization algorithm is not optimal)
Building the Pareto surface incrementally (II)

Building a grid with the constrains

One schedule $S$ in the 3D space
How to statically schedule large graphs?

- Optimal multiprocessor scheduling is NP-complete [Garey & Johnson, 1979]

- For large graphs, only approximated methods are feasible

- Typically list scheduling [Hu, OR’61]:
  
  - **Ready operations** are those for which all predecessors are scheduled
  
  - Ready operations are sorted by a “smart” cost function and the first operation is scheduled
  
  - There is no backtracking

- (We haven’t tried other scheduling methods, e.g., genetic)
The problem with monotonous measures

\[ E(S) \]

\[ E_{\text{obj}} \]

Each newly scheduled operation adds a strictly positive term

\[ \Rightarrow \text{Energy is strictly increasing} \]
The problem with monotonous measures

Each newly scheduled operation adds a strictly positive term

- Energy is strictly increasing

Each newly scheduled operation multiplies a term strictly below 1

- Reliability is strictly decreasing
The problem with monotonous measures

Each newly scheduled operation adds a strictly positive term

\[ E(S) \]

\[ E_{obj} \]

\[ 0 \]

\[ 1 \]

\[ 2 \]

\[ 3 \]

\[ 4 \]

\[ 5 \]

\[ 6 \]

\[ 7 \]

\[ \ldots \]

\[ \text{operation number} \]

Energy is strictly increasing

Each newly scheduled operation multiplies a term strictly below 1

\[ R(S) \]

\[ R_{obj} \]

\[ 0 \]

\[ 1 \]

\[ 2 \]

\[ 3 \]

\[ 4 \]

\[ \text{operation number} \]

Reliability is strictly decreasing

Third principle

The two criteria transformed into constraints must be invariant w.r.t. time.
Summary of contributions

- **Energy** is not invariant, but the **power consumption** is
- **Reliability** is not invariant, but the **failure rate per time unit** is

**Contribution**

Tricriteria scheduling algorithm (length, GSFR, power)

GSFR generalizes the failure rate per time unit to a distributed schedule

[Girault & Kalla, IEEE TDSC’09]

- Length and GSFR are antagonistic
- Length and power consumption are antagonistic
- GSFR and power consumption are antagonistic
Outline

1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
DFS and DVS

Recall the power model

\[ P = P_s + h(P_{\text{ind}} + P_d) \quad \text{with} \quad P_d = C_{ef} V^2 f \]

The cost of putting the system to sleep (i.e., putting \( h = 0 \)) is very high, so we only play with \( P_d \)

DFS = Dynamic Frequency Scaling

It lowers the frequency of the processor, i.e., it increases the cycle period

DVS = Dynamic Voltage Scaling

It lowers the voltage of the processor

DVS \Rightarrow DFS \quad \text{(but the reverse is not true)}

When the voltage is lowered, the cycle period is increased proportionally
Dynamic Voltage and Frequency Scaling – DVFS

- Recall the power model:
  \[ P = P_s + h(P_{\text{ind}} + P_d) \]
  \[ P_d = C_{ef} V^2 f \]

- DVFS plays on the frequency dependent power \((P_d)\) only

- Square relationship between \(V\) and \(P_d\)
  \[ \Rightarrow \text{DVFS decreases } E \]

- But \(P_{\text{ind}}\) grows linearly as \(f\) decreases

- Thanks to the square relationship, the overall gain is significant

Total energy w.r.t. the voltage level.
Impact of DVFS on the failure rate

Exponential relationship between the fault probability and the frequency (lower voltage makes the chip sensitive to smaller energy particles)

\[ \lambda(f) = \lambda_0 \cdot 10^{\frac{b(1 - f)}{1 - f_{\text{min}}}} \]

- \( \lambda_0 \) is the nominal failure rate per time unit
- \( b > 0 \) is a constant
- \( f \) is the frequency scaling factor
- \( f_{\text{min}} \) is the lowest operating frequency
- The failure rate is maximal at \( f_{\text{min}}/V_{\text{min}} \): \( \lambda_{\text{max}} = \lambda(f_{\text{min}}) = \lambda_0 \cdot 10^b \)
(1) Energy consumption of a schedule $S$:

$$E(S) = \sum_{j=1}^{\left|P\right|} \left( P_{\text{ind}}^p \cdot L(S) + \sum_{o_i \in p_j} P_d^j \cdot L(o_i, p_j, f_{i,j}) \right) \quad // \text{processors}$$

$$+ \sum_{k=1}^{\left|L\right|} \left( P_{\text{ind}}^\ell \cdot L(S) + \sum_{d_i \in \ell_k} P_d^k \cdot L(d_i, \ell_k, f_{\ell}) \right) \quad // \text{links}$$

(2) Resulting power consumption:

$$P(S) = E(S) / L(S)$$
Power consumption of a multiprocessor schedule

\[ P_1 \left( C_{ef}^p, P_{ind}^p \right) \]
\[ L_{12} \left( C_{ef}^\ell, P_{ind}^\ell \right) \]
\[ P_2 \left( C_{ef}^p, P_{ind}^p \right) \]

- On \( P_1 \): \( E(P_1) = P_{ind}^p \cdot L + C_{ef}^p \left( V_1^2 f_1 L(X, P_1, f_1) + V_2^2 f_2 L(Z, P_1, f_2) \right) \)
- On \( P_2 \): \( E(P_2) = P_{ind}^p \cdot L + C_{ef}^p V_3^2 f_3 L(Y, P_2, f_3) \)
- On \( L_{12} \): \( E(L_{12}) = P_{ind}^\ell \cdot L + C_{ef}^\ell V_\ell^2 f_\ell L(X \triangleright Y, L_{12}, f_\ell) \)
Outline

1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
Invariant criteria to measure the reliability

Recall the reliability model: \[ R(X, P) = e^{-\lambda P \ L(X,P)} \]

**GSFR = global system failure rate per time unit**

The GSFR is the failure rate per time unit of the global system \( S \), seen as if it were a single HW component.

\[ \text{GSFR}(S) = \Lambda(S) = \frac{-\log R(S)}{U(S)} \quad \text{with} \quad U(S) = \sum_{(o_i, p_i) \in S} L(o_i/p_i) \]

Consistent with the “hot” failure model

Note that the usual reliability formula holds: \[ R(S) = e^{-\Lambda(S) \ U(S)} \]
Variation of the exact replication factor in function of $\Lambda_{obj}$

$|\mathcal{P}| = k = 4$

Exact replication factor of operation $i$

Number of operation $i$

$\bigcirc \quad |\mathcal{P}| = k = 4$

Assayad, Girault, and Kalla (INRIA) (length, reliability, power) scheduling August 14th, 2012 29 / 47
Computing the reliability: Reliability Block-Diagrams (RBD)

S if (I1,P2) not fail and (I1,A,L2-3) not fail and (A,P3) not fail then succees D

A

P3 L2-3 P2 L1-2 P1 L1-4 P4 L3-4

A

P3 L2-3 P2 L1-2 P1 L1-4 P4 L3-4

A

I1

A

I1

I1
In general, the reliability computation exponential in the RBD size

(aka terminal-pair problem, NP-complete [Ball, IEEE TR’86])
Making the RBD serial-parallel

Simple algorithm graph:

We insert routing operations in the algorithm task graph:

They incur an additional overhead on the schedule length, because there is less concurrency between the communications.

Since there are also less communications, this additional overhead is less than 4% on average [Girault & Kalla, IEEE TDSC’09]
RBD of a schedule without replication

The RBD is:

\[
\begin{align*}
R &= R(X, P1) \cdot R(X \triangleright Y, L12) \cdot R(Y, P2) \\
&= e^{-\lambda_1 L(X, P1)} \cdot e^{-\lambda_{12} L(X \triangleright Y, L12)} \cdot e^{-\lambda_2 L(Y, P2)} \\
\Lambda &= \frac{-\log R}{U} = \frac{\lambda_1 L(X, P1) + \lambda_{12} L(X \triangleright Y, L12) + \lambda_2 L(Y, P2)}{L(X, P1) + L(X \triangleright Y, L12) + L(Y, P2)}
\end{align*}
\]
RBD of a schedule with replication

\[ \text{RBD of a schedule with replication} \]

\[ \text{P1} \quad \text{L13} \quad \text{P3} \quad \text{L23} \quad \text{P2} \quad \text{L34} \quad \text{P4} \quad \text{L35} \quad \text{P5} \]

\[ \begin{align*}
&X^1 \\
&X \triangleright Y \\
&X \triangleright Y \\
&R \\
&X \triangleright Y \\
&\quad \\
&X \triangleright Y \\
&\quad \\
&Y^1 \\
&\quad \\
&Y^2
\end{align*} \]

\[ \text{block } B_1 \quad \text{block } B_2 \quad \text{block } B_3 \]

\[ \begin{align*}
&(X^1/P1) \\
&(X \triangleright Y/L13) \\
&(X^2/P2) \\
&(X \triangleright Y/L23)
\end{align*} \]

\[ \begin{align*}
&(X \triangleright Y/L34) \\
&(R/P3) \\
&(X \triangleright Y/L35)
\end{align*} \]

\[ \begin{align*}
&(Y^1/P4) \\
&(Y^2/P5)
\end{align*} \]
Outline

1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
State of the art in tricriteria scheduling

- [Pop, Poulsen & Izosimov, CODES-ISSS’07] : (length, reliability, energy), heterogeneous architecture, reliable bus, re-execution of failed operations.

- [Gan, Gruian, Pop & Madsen, ASP-DAC’11] : set of communicating periodic real-time tasks, heterogeneous architecture, critical tasks must be replicated (given), minimize the energy under the reliability constraint and the deadline constraints. Tabu-search heuristics with random moves.

Tricriteria Scheduling Heuristics (TSH)

Contribution

TSH = tricriteria scheduling algorithm (length, GSFR, power)

- Inputs: $Alg$, $Arc$, WCETs, $P_{obj}$, $\Lambda_{obj}$.
- Output: a distributed static schedule $S$ of $Alg$ on $Arc$ s.t. $L(S)$ is minimal, $P(S) \leq P_{obj}$, and $\Lambda(S) \leq \Lambda_{obj}$
- Ready list scheduling heuristics
- Ready operations are sorted by a smart cost function
- Cost function: power-efficient dependable schedule pressure
- The most urgent operation is scheduled on a subset of processors s.t. $GSFR < \Lambda_{obj}$, $P < P_{obj}$, and the increase in schedule length is minimal
Cost function for the list scheduling heuristics

Schedule pressure of operation $o_i$ on processor $p_j$:
[Grandpierre, Lavarenne & Sorel, CODES’99]

\[
\sigma^{(n)}(o_i, p_j) = ETS^{(n)}(o_i, p_j) + LTE^{(n)}(o_i) - CPL^{(n)}
\]

- $ETS^{(n)}(o_i, p_j) =$ Earliest Time at which $o_i$ can Start on $p_j$
- $LTE^{(n)}(o_i) =$ Latest start Time from End of $o_i$
  (length of the longest path from $o_i$ to Alg’s output operations)
- $CPL^{(n)} =$ Critical Path Length of $S^{(n)}$
Generalized schedule pressure

Recall the schedule pressure of operation $o_i$ on processor $p_j$:

$$\sigma^{(n)}(o_i, p_j) = ETS^{(n)}(o_i, p_j) + LTE^{(n)}(o_i) - CPL^{(n)}$$

The generalized schedule pressure of $o_i$ on the set $\mathcal{P}_k$ is:

$$\sigma^{(n)}(o_i, \mathcal{P}_k) = ETS^{(n)}(o_i, \mathcal{P}_k) + LTE^{(n)}(o_i) - CPL^{(n)}$$

$$ETS^{(n)}(o_i, \mathcal{P}_k) = \max_{p_j \in \mathcal{P}_k} ETS^{(n)}(o_i, p_j)$$
Power-efficient schedule pressure

Power-efficient schedule pressure:

Find the set $Q_{best}^{(n)}(o_i) = Q_j$ such that:

$$
\sigma^{(n)}(o_i, Q_j) = \min_{Q_k \in \mathcal{Q}} \left\{ \sigma^{(n)}(o_i, Q_k) \mid \Lambda_B(o_i, Q_k) \leq \Lambda_{obj} \right. \\
\left. \wedge (E^{(n+1)} - E^{(n)}) \leq P_{obj}(L^{(n+1)} - L^{(n)}) \right\}
$$

Finally, the most urgent operation is:

$$o_{urg} = o_i \in O_{\text{ready}}^{(n)} \text{ s.t. } \sigma^{(n)}(o_i, Q_{best}^{(n)}(o_i)) = \max_{o_j \in O_{\text{ready}}^{(n)}} \left\{ \sigma^{(n)}(o_j, Q_{best}^{(n)}(o_j)) \right\}$$
If the next scheduled operation does not increase the length \( L \) because it fits in a slacks, then it could be impossible to meet the constraint \( P_{\text{obj}} \):

\[
L_{(n+1)} = L^{(n)} \land E_{(n+1)} > E^{(n)} \implies (E^{(n+1)} - E^{(n)}) \not\leq P_{\text{obj}}(L^{(n+1)} - L^{(n)})
\]
Soundness of TSH

Theorem 1: soundness w.r.t. the GSFR constraint

If each operation $o$ of $\mathcal{A}lg$ is scheduled such that $\Lambda_B(o, Q_k) \leq \Lambda_{obj}$, then $\Lambda(S) \leq \Lambda_{obj}$. 
Soundness of TSH

**Theorem 1:** soundness w.r.t. the GSFR constraint

If each operation $o$ of $\mathcal{Alg}$ is scheduled such that $\Lambda_B(o, \mathcal{Q}_k) \leq \Lambda_{obj}$, then $\Lambda(S) \leq \Lambda_{obj}$.

**Theorem 2:** soundness w.r.t. the power constraint

If each operation $o$ of $\mathcal{Alg}$ is scheduled such that 

$$ (E^{(n+1)} - E^{(n)}) \leq P_{obj}(L^{(n+1)} - L^{(n)}) $$

then $P(S) \leq P_{obj}$. 
Outline

1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
Schedule length in function of the power and the GSFR; \(|P|=3; |X|=30\)

Pareto surface for an Alg graph of 30 tasks, over 3 processors, \(\lambda_p = 10^{-5}\), \(\lambda_\ell = 5.10^{-4}\), \(\mathcal{V} = \{0.25, 0.50, 0.75, 1.0\}\)
Schedule length in function of the power and the GSFR; \(|P|=4; |X|=30\)

Pareto surface for an \(A/l_g\) graph of 30 tasks, over 4 processors \(\lambda_p = 10^{-5}\), \(\lambda_l = 5.10^{-4}\), \(V = \{0.25, 0.50, 0.75, 1.0\}\)
Comparison with MILP on a small instance

Problem instance: Alg graph of 5 tasks, over 3 processors $\lambda_p = 10^{-5}$, $\lambda_\ell = 5.10^{-4}$, $\mathcal{V} = \{0.25, 0.50, 0.75, 1.0\}$ $\Rightarrow$ 15% to optimal
Outline

1. Context and overview
2. Multicriteria optimization for power and reliability
3. Power reduction techniques
4. Reliability improvement techniques
5. Our tricriteria scheduling algorithm TSH
6. Simulations
7. Conclusion
The three principles

Offer tradeoff choices ⇔ compute the Pareto surface ⇔ turn two criteria into constraints ⇔ those criteria must be invariant w.r.t. time
Conclusion

The three principles

Offer tradeoff choices ⇔ compute the Pareto surface ⇔ turn two criteria into constraints ⇔ those criteria must be invariant w.r.t. time

The three criteria

- Minimize the schedule length / WCET / $C_{\text{max}}$ / makespan
- Bound the power consumption by $P_{\text{obj}}$
- Bound the global system failure rate by $\Lambda_{\text{obj}}$
Conclusion

The three principles
Offer tradeoff choices ⇔ compute the Pareto surface ⇔ turn two criteria into constraints ⇔ those criteria must be invariant w.r.t. time

The three criteria
- Minimize the schedule length / WCET / $C_{\text{max}}$ / makespan
- Bound the power consumption by $P_{\text{obj}}$
- Bound the global system failure rate by $\Lambda_{\text{obj}}$

The tricriteria scheduling heuristics TSH
- Ready list scheduling heuristics
- Operations are sorted by the power-efficient schedule pressure
- Each run is constrained in a cell of a regular grid
- Produces the Pareto surface in the 3D space (length,GSFR,power)