On Computer-Aided Design-Space Exploration for Multi-Cores

Jean-Francois Kempf

October 29, 2012

Jury members:

Oded Maler	Directeur de these
Marius Bozga	Co-Directeur de these
Kim G. Larsen	Rapporteur
Bruce Krogh	Rapporteur
Boudewijn R. Haverkort	Examinateur
Eugene Asarin	Examinateur
Fahim Rahim	Examinateur

Context of the thesis

Minalogic project ATHOLE

- Low-power multi-processors platform for embedded systems.
- Partners: ST, CEA Leti, Thales, CWS, Verimag.
- Verimag: High level modeling and analysis.

This thesis

- Development of a framework for modeling and analysis of embedded systems.
- Development of new probabilistic analysis.

Motivation

Embedded systems are everywhere:

Motivation

Embedded systems:

- Combination of computer hardware and software
- Designed to perform a dedicated function

Motivation

Embedded systems:

- Combination of computer hardware and software
- Designed to perform a dedicated function

Requirements:

- Low cost implying:
  - Short design & development time
  - Limited resources (memory, CPU . . .)
- High performance due to increasing functionality
- Low energy consumption

Embedded Systems Design

- There are several design choices both in hardware and software

Embedded Systems Design

- There are several design choices both in hardware and software
- Each of them has advantages according to different criteria:
  - Timing performance
  - Power consumption
  - Platform cost
  - ...

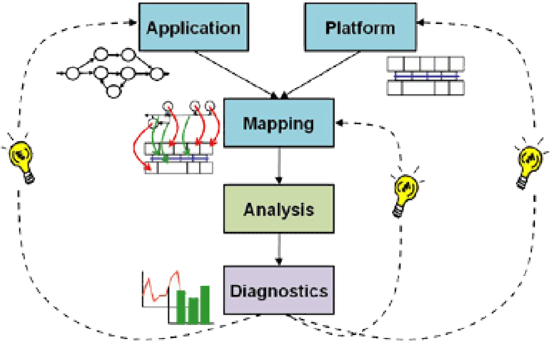
Embedded Systems Design

- There are several design choices both in hardware and software
- Each of them has advantages according to different criteria:
  - Timing performance
  - Power consumption
  - Platform cost
  - ...

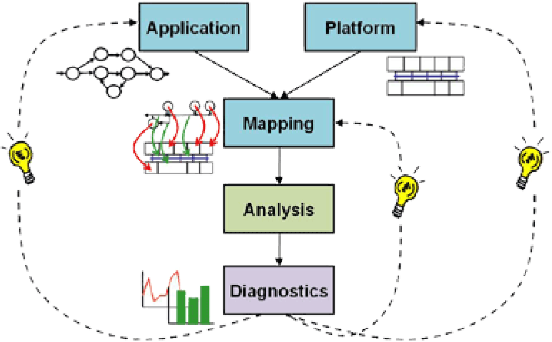
Needs

- **Performance estimation as soon as possible**
  - evaluate quickly different trade-off

Design-Space exploration



Design-Space exploration



In this thesis:
Exploration and Analysis on a high level of abstraction.

High-Level Performance Evaluation

Advantages:

- Works at virtual level:
  - No need of a physical platform
  - No need of a complete implementation
- Models are simplified:
  - Performance analysis is tractable
  - Simulation and analysis is feasible fast
- Evaluation of different alternatives can be done easily

Weakness:

- Parameters estimation
- Model calibration
- Accuracy of the results

Contribution

A framework for high level modeling and analysis

Model:

- **Applications** are modeled at task level
  (task graph with data transfer)

Contribution

A framework for high level modeling and analysis

Model:

- **Applications** are modeled at task level
  (task graph with data transfer)
- **Environment** models the dynamics of task arrival

Contribution

A framework for high level modeling and analysis

Model:

- **Applications** are modeled at task level
  (task graph with data transfer)
- **Environment** models the dynamics of task arrival
- **Architecture** is an abstraction of MPSoC

Contribution

A framework for high level modeling and analysis

Model:

- **Applications** are modeled at task level
  (task graph with data transfer)
- **Environment** models the dynamics of task arrival
- **Architecture** is an abstraction of MPSoC
- **Deployment** : mapping and scheduling

Analysis:

- Simulation
- Statistical analysis
- Formal verification

Uncertainty

To compensate the lack of precision at this level of description:

- Increase the uncertainty margins
- Consider this uncertainty in the analysis

Uncertainty

Modeling uncertainty with timed automata

- Timing informations are modeled as intervals
- Exhaustive reachability analysis
- Analysis is **worst-case oriented** and sometimes intractable.

Uncertainty

Modeling uncertainty with timed automata

- Timing informations are modeled as intervals
- Exhaustive reachability analysis
- Analysis is **worst-case oriented** and sometimes intractable.

We may be more concerned about the **average performance**.

Uncertainty

Modeling uncertainty with timed automata

- Timing informations are modeled as intervals
- Exhaustive reachability analysis
- Analysis is **worst-case oriented** and sometimes intractable.

We may be more concerned about the **average performance.**

Modeling uncertainty probabilistically

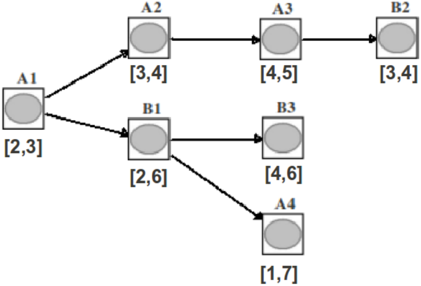
- Duration Probabilistic Automata:
  - Timed automata with probabilistic duration
  - Discrete event simulation and statistical analysis
  - Exact computation of expected termination time

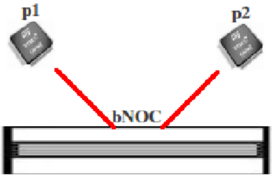
A simple example

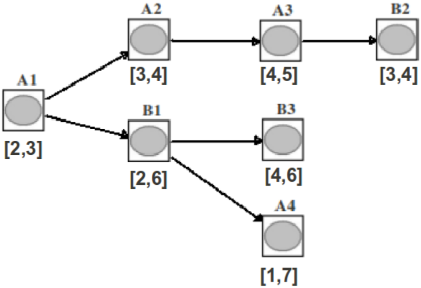
We show, with this example, the importance of considering the *uncertainty* in the analysis.

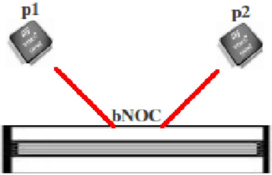
Outcome

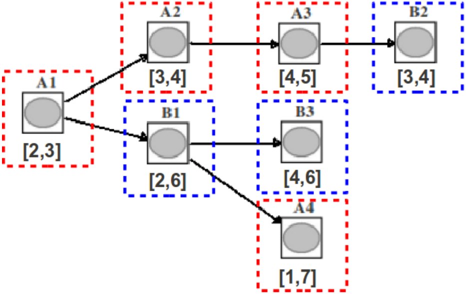
- Timing analysis based exclusively on **worst case execution times** might not catch the **worst behavior**

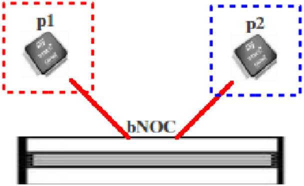




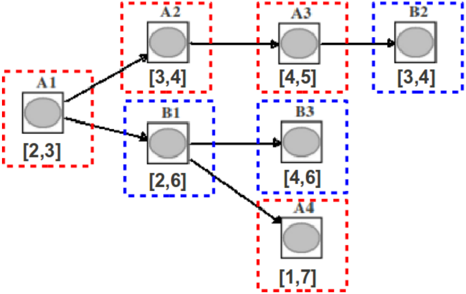

A1
[2,3]
A2
[3,4]
A3
[4,5]
B2
[3,4]
B1
[2,6]
B3
[4,6]
A4
[1,7]

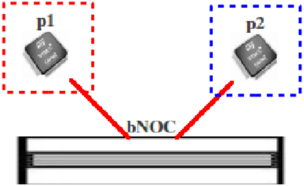

p1
p2
bNOC





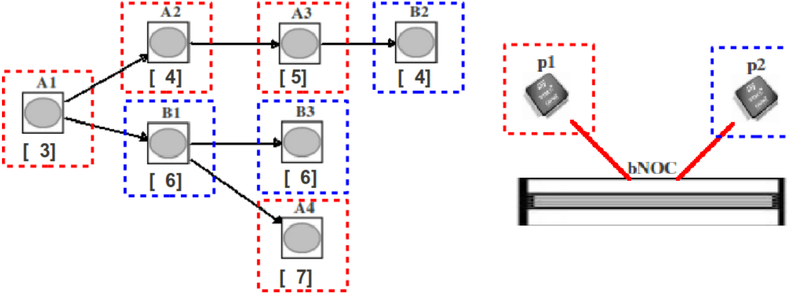
- FIFO scheduling (non preemptive)





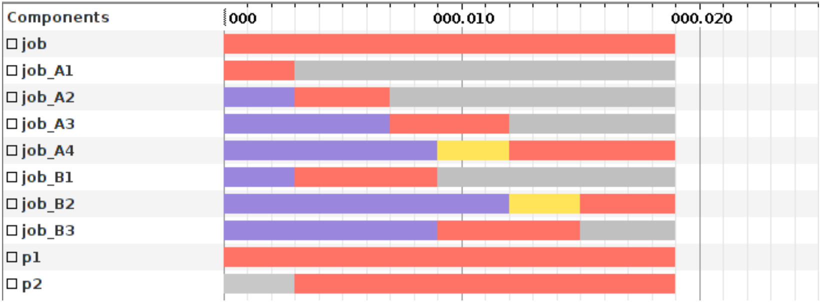
- FIFO scheduling (non preemptive)
- Question:
  - **What is the maximal response time of the job ?**

Corner-Case Analysis



- Naively, to get the maximal response time, one might do an analysis based on *worst-case execution time* for all tasks.

Corner-Case Analysis

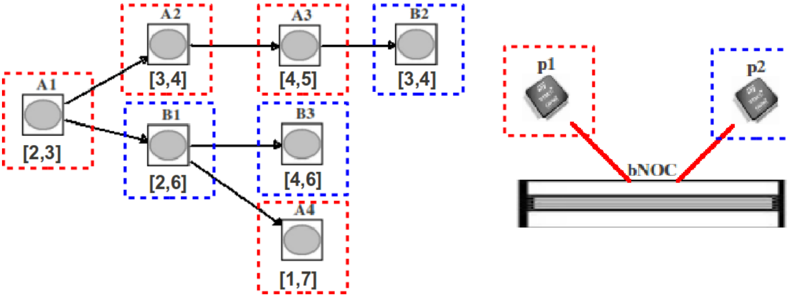


- Analysis gives a response time of 19 timeunits

One might conclude:

- **The worst response time for the job is 19**

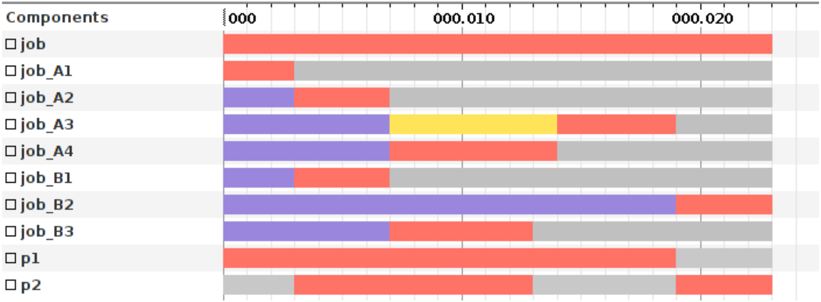
Reachability Analysis with Uncertainty



We use now timed automata reachability analysis:

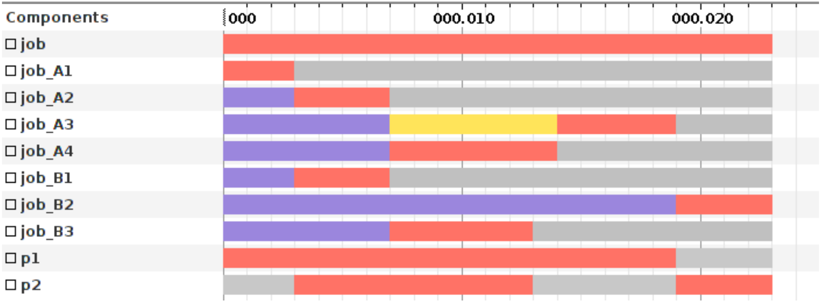
- Explore all possible behaviors.
- Retrieve the execution trace leading to the worst response time.

Reachability Analysis with Uncertainty



- We get a worst response time of 23 timeunits.

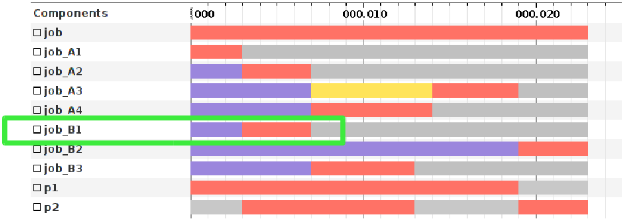
Reachability Analysis with Uncertainty

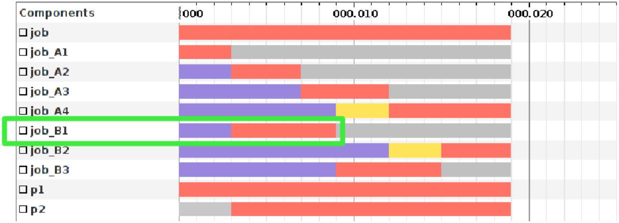


- We get a worst response time of 23 timeunits.

The worst response time for the job is 23

Explanations





B1 takes less time $\Rightarrow$ A4 start before A3 (on critical path).

Quantitative Estimation

Uncertainty plays also an important role when we care more about the **average performance**

Quantitative Estimation

Uncertainty plays also an important role when we care more about the **average performance**

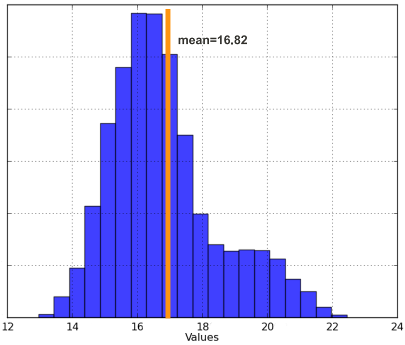
- Assumption: execution times are distributed uniformly.

Quantitative Estimation

Uncertainty plays also an important role when we care more about the **average performance**

- Assumption: execution times are distributed uniformly.

With simulation we get more quantitative information:



Example: Conclusion

- Analysis based on deterministic values (lower and upper) might give incorrect bounds on the global response time.

Example: Conclusion

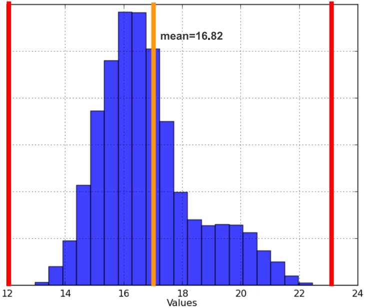
- Analysis based on deterministic values (lower and upper) might give incorrect bounds on the global response time.
- Timed automata reachability analysis gives us correct bounds but no quantitative information.

Example: Conclusion

- Analysis based on deterministic values (lower and upper) might give incorrect bounds on the global response time.
- Timed automata reachability analysis gives us correct bounds but no quantitative information.
- Stochastic simulation does not catch tight bounds but gives more quantitative information about average performance.

Example: Conclusion

- Analysis based on deterministic values (lower and upper) might give incorrect bounds on the global response time.
- Timed automata reachability analysis gives us correct bounds but no quantitative information.
- Stochastic simulation does not catch tight bounds but gives more quantitative information about average performance.



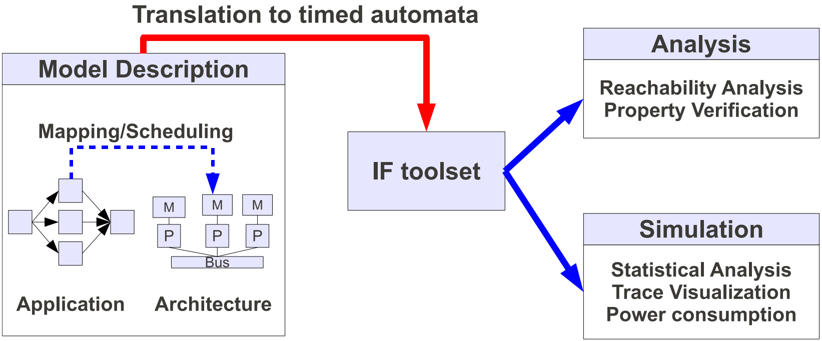
Outline

- DeSpEx: Design Space Explorer
  - A framework for modeling and analysis at high level of abstraction.
- Duration Probabilistic Automata:
  - Exact analysis of expected termination time
  - Synthesis of expected time optimal schedulers

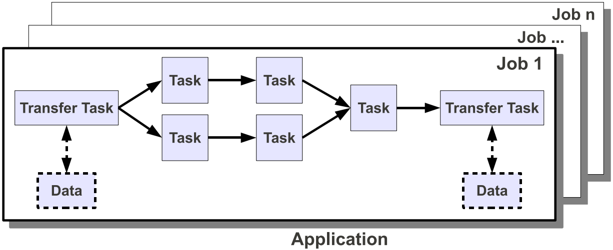
DeSpEx
Objectives

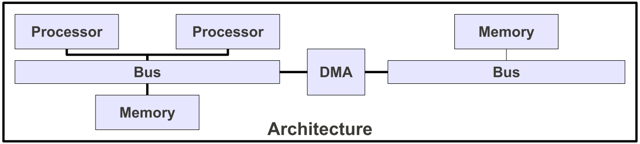
- Provide HW/SW designers with a framework for rapid design space exploration
- High level language for model description
- Formal semantic provided by *timed automata*
- Performance evaluation using formal methods and stochastic simulation

Framework Overview

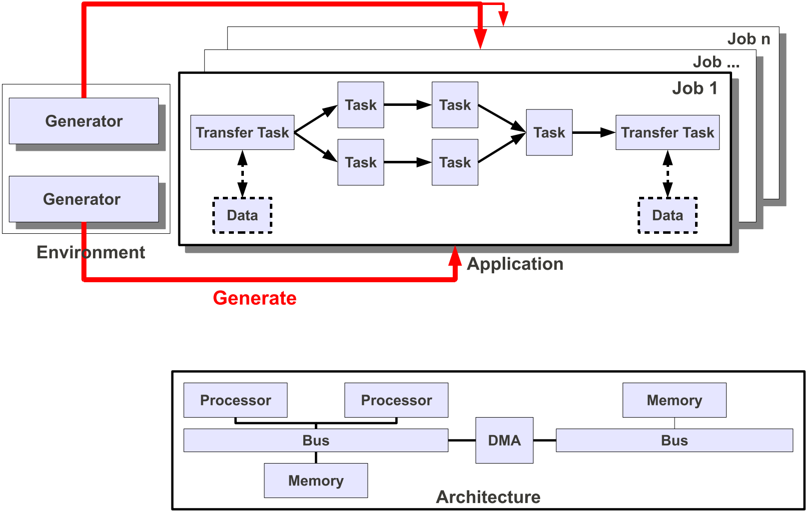


Model overview

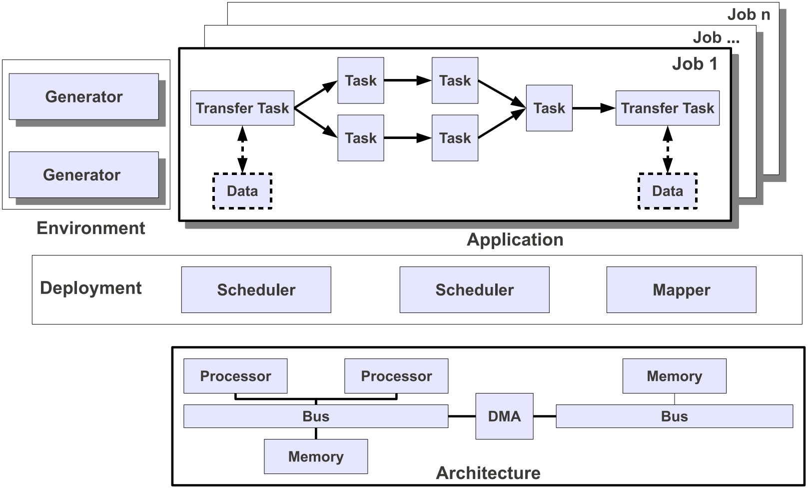




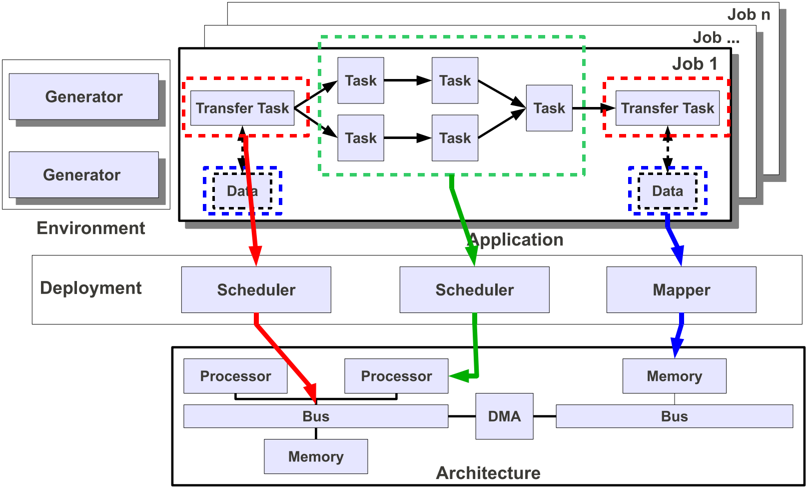
Model overview



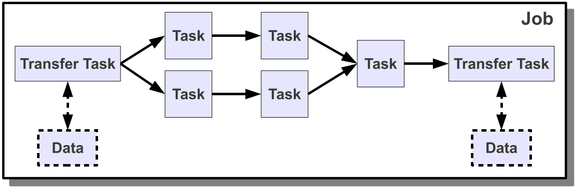
Model overview



Model overview

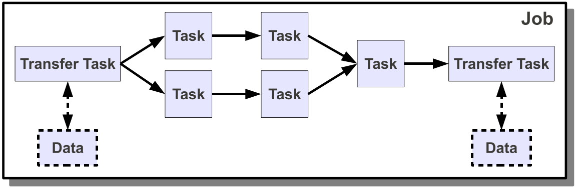


Application



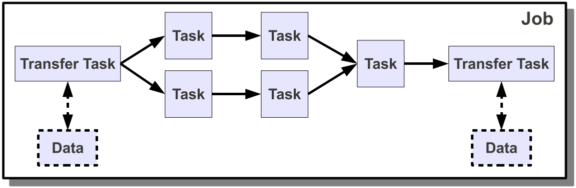
- Basic unit of work whose instance arrive to be execute

Application



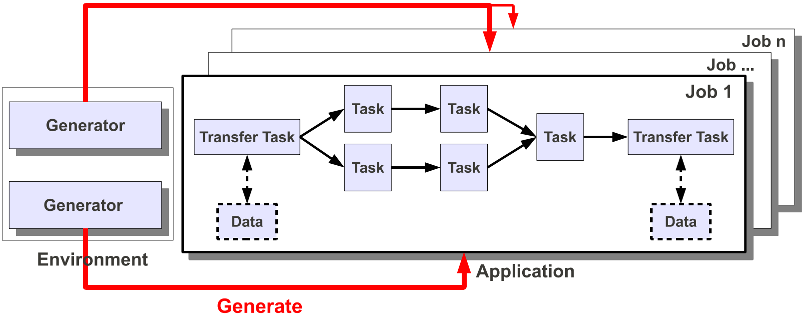
- Basic unit of work whose instance arrive to be execute
- Generalized *task-graph* with data transfer
  - Task are related by precedence
  - Distinction between computation and transfer
  - Tasks are characterized by:
    - an amount of work (computation)
    - a quantity of data to communicate (transfer)

Application



- Basic unit of work whose instance arrive to be execute
- Generalized *task-graph* with data transfer
  - Task are related by precedence
  - Distinction between computation and transfer
  - Tasks are characterized by:
    - an amount of work (computation)
    - a quantity of data to communicate (transfer)
- Extended to allow *loops* modeling
  - execute a finite number of iterations sequentially

Environment



For performance evaluation we need a way to model arrival of workload:

- **Input generators**:
  - process that generates a timed sequence of job instances subject to some logical and timing constraints.

Generators

- Periodic $t_k = O + k \cdot P$

Generators

- Periodic $t_k = O + k \cdot P$
- Periodic with Uncertainty $t_k + P \leq t_{k+1} \leq t_k + P + J$

Generators

- Periodic $t_k = O + k \cdot P$
- Periodic with Uncertainty $t_k + P \leq t_{k+1} \leq t_k + P + J$
- Periodic with Jitter $O + k \cdot P \leq t_k \leq O + k \cdot P + J$

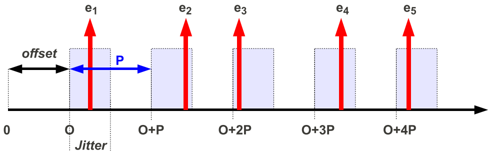
Generators

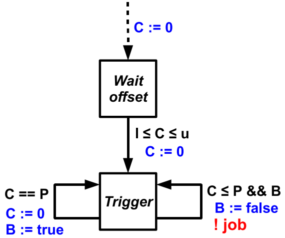
- Periodic $t_k = O + k \cdot P$
- Periodic with Uncertainty $t_k + P \leq t_{k+1} \leq t_k + P + J$
- Periodic with Jitter $O + k \cdot P \leq t_k \leq O + k \cdot P + J$
- Bounded Variability At most n event in any time interval of length Δ

Generators

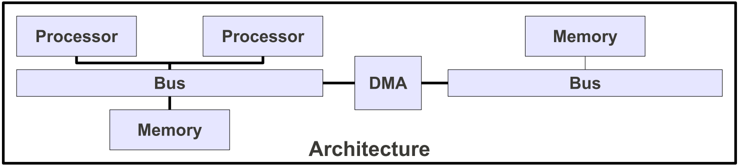
- Periodic — $t_k = O + k \cdot P$
- Periodic with Uncertainty — $t_k + P \leq t_{k+1} \leq t_k + P + J$
- Periodic with Jitter — $O + k \cdot P \leq t_k \leq O + k \cdot P + J$
- Bounded Variability — At most n event in any time interval of length Δ

Example: Periodic Generator with Jitter





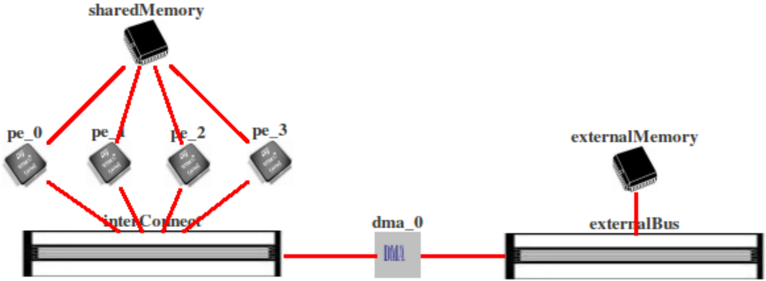
Architecture



- **Processors**: one or a fixed number of *frequencies* (sets the number cycles/instructions per time units)
- **Memory**: *access latency* $[l, u]$ and a *rate* (defines the number of *data units* treated per time unit)
- **Bus**: static routing table and a *bandwidth* (defines the number of *data units* communicated per time unit)
- For each component:
  - *power consumption* value associated with its different states

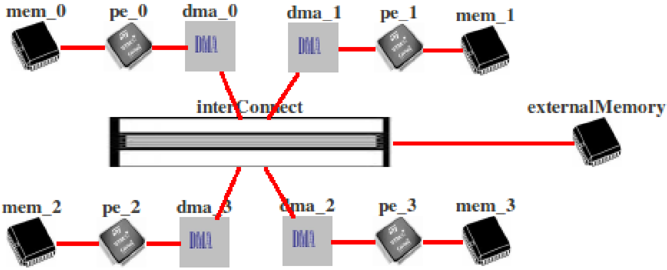
Architecture

Example: Abstract Model inspired from P2012 Architecture



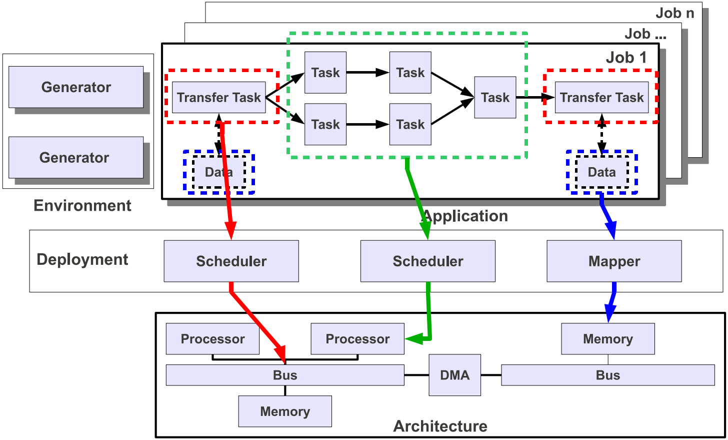
Architecture

Example: Abstract Model inspired from Cell Architecture



Deployment

Defines the mapping and the scheduling of the application on the architecture



Deployment

Task Scheduling

We provide several non-preemptive scheduling scheme:

- **FIFO**: tasks are pushed into processor according to their arrival order
- **Priority Scheduling**: tasks are pushed into processor according to their priority
- **Frequency Scaling Scheduling**: a task can switch the frequency of a processor before starting its execution

Deployment

Task Scheduling

We provide several non-preemptive scheduling scheme:

- **FIFO**: tasks are pushed into processor according to their arrival order
- **Priority Scheduling**: tasks are pushed into processor according to their priority
- **Frequency Scaling Scheduling**: a task can switch the frequency of a processor before starting its execution

Each scheduling scheme can be defined in two ways:

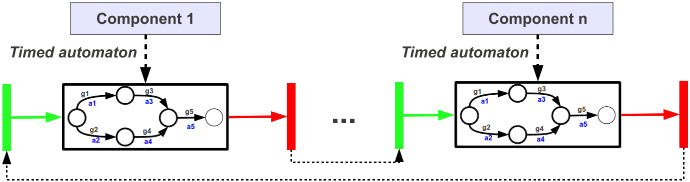
- *Global* scheduling: one scheduler in charge of all processing elements
- *Local* scheduling: each processing element has its own scheduler

Evaluation

- The aim of this modeling framework is provide design space exploration for performance evaluation

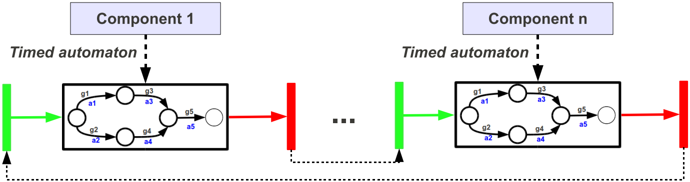
Evaluation

- The aim of this modeling framework is provide design space exploration for performance evaluation
- The composition of all automata yields a global timed automaton which captures the semantics of the system



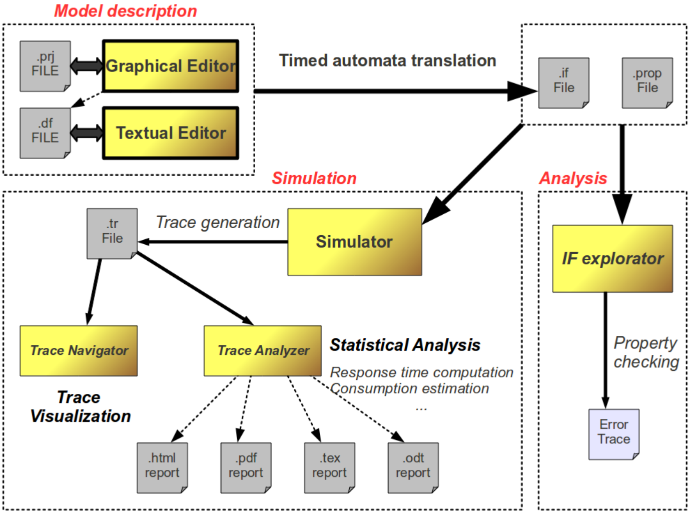
Evaluation

- The aim of this modeling framework is provide design space exploration for performance evaluation
- The composition of all automata yields a global timed automaton which captures the semantics of the system



- All our analysis methods are based on a unified semantic model provided by timed automata

Tool Overview

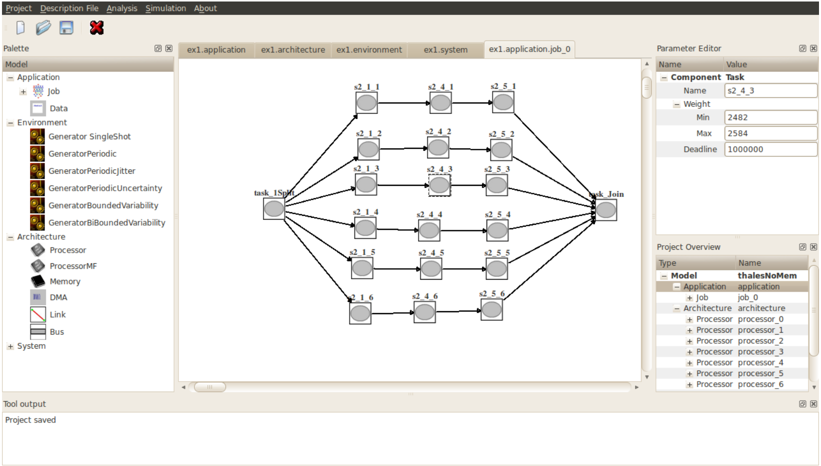


DeSpEx: Textual Model Editor

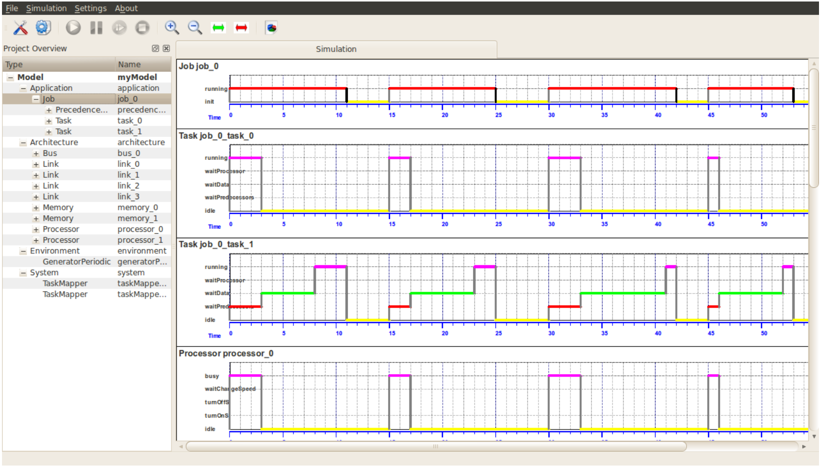
Project Description File Analysis Simulation About

```
Model mymodel
{
   Application
   {
      Data data_1;
      ...
      Job job_0
      {
         Task task_0 {...}
         ...
         Task task_n {...}
         ...
         PrecedenceLink precedencelink_0[task_0 , task_1];
         ...
      }
      ...
      Job job_n {...}
   }
```

DeSpEx: Graphical Model Editor



DeSpEx: Simulation and Trace Visualization



DeSpEx: Trace Visualization (Gantt Chart)

Timing Evaluation

Reachability Analysis

- Check whether some properties are satisfied or not
- In case of property violation generates an error trace, viewable with the GUI

Timing Evaluation

Reachability Analysis

- Check whether some properties are satisfied or not
- In case of property violation generates an error trace, viewable with the GUI

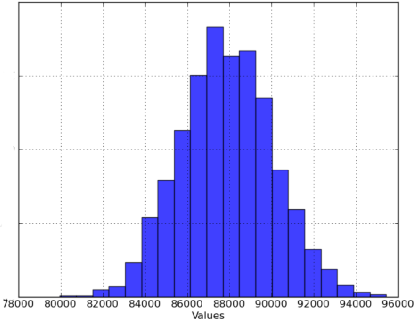
Stochastic Simulation

- Depending on the size of the model it may be difficult (sometimes impossible) to perform reachability analysis
- Timed automata are used to perform discrete event simulation:
  - **Semantic model is the same** as for reachability analysis
  - Randomized reachability exploration
  - We restrict ourself to bounded uncertainty and define a random variable for each uncertainty interval

Timing Evaluation

Trace Analysis

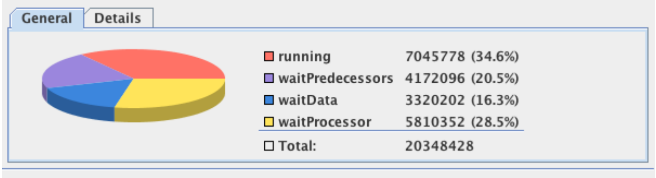
- Stochastic simulation generates timed traces
- We can retrieve quantitative informations with trace analysis:
  - Response time distribution of job



Timing Evaluation

Trace Analysis

- Stochastic simulation generates timed traces
- We can retrieve quantitative informations with trace analysis:
  - Response time distribution of job
  - State distribution for a computation task

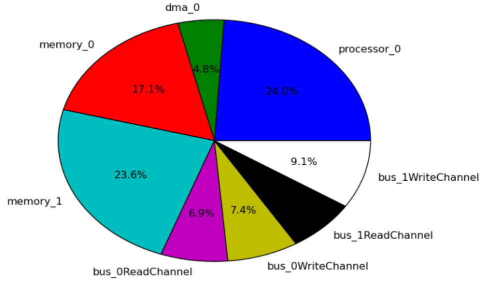


Power Consumption Estimation

- Done via trace analysis
- Each state of an architecture component is associated with a power value
- We can derive from a simulation trace the energy consumption for each component:

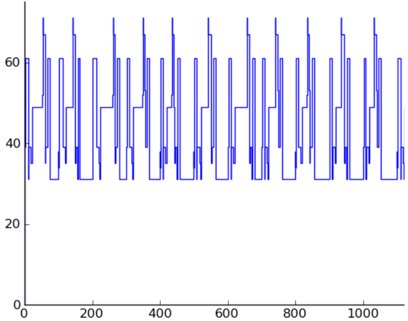
Power Consumption Estimation

- Done via trace analysis
- Each state of an architecture component is associated with a power value
- We can derive from a simulation trace the energy consumption for each component:
  - Energy consumption proportions between HW components:



Power Consumption Estimation

- Done via trace analysis
- Each state of an architecture component is associated with a power value
- We can derive from a simulation trace the energy consumption for each component:
  - Trace of energy consumption in time:



Case Studies

We demonstrate the usage of the tool on two industrial case studies:

- A Radio Sensing Application:
  - Check the feasibility of porting an application, currently running sequentially on a powerful desktop, to an embedded multi-processor machine

Case Studies

We demonstrate the usage of the tool on two industrial case studies:

- A Radio Sensing Application:
  - Check the feasibility of porting an application, currently running sequentially on a powerful desktop, to an embedded multi-processor machine
- Video Processing on P2012:
  - Evaluate different design alternatives

Case Studies

We demonstrate the usage of the tool on two industrial case studies:

- A Radio Sensing Application:
  - Check the feasibility of porting an application, currently running sequentially on a powerful desktop, to an embedded multi-processor machine
- Video Processing on P2012:
  - Evaluate different design alternatives

Video Processing on P2012

Goal

- Demonstrate how DeSpEx can be used to solve realistic problems in design space exploration

Video Processing on P2012

Goal

- Demonstrate how DeSpEx can be used to solve realistic problems in design space exploration
- Quantify the performance differences between different design choices

Video Processing on P2012

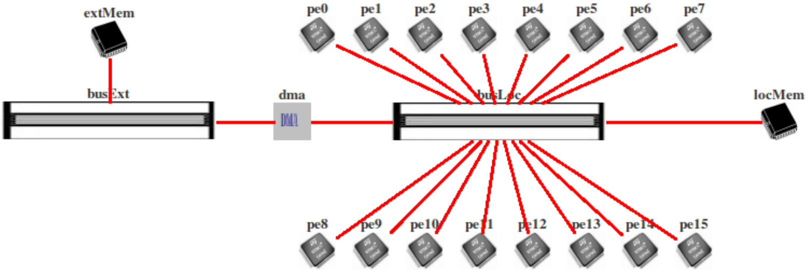
Goal

- Demonstrate how DeSpEx can be used to solve realistic problems in design space exploration
- Quantify the performance differences between different design choices
- Represent available cost/performance trade-offs.

Video Processing on P2012

Architecture Abstraction

- In the context of the ATHOLE project, the architecture is inspired by the P2012 platform
- P2012 is a many-core computing fabric based on multiple clusters
- We restrict our models to one cluster

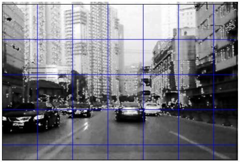


Video Processing on P2012

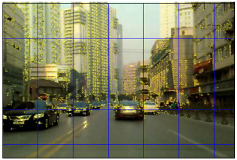
Application

- Augmented reality application called FAST (Features from Accelerated Segment Test)
- Corner detection method
- Algorithm consists in computing the detection on a chunk of an image

A



B



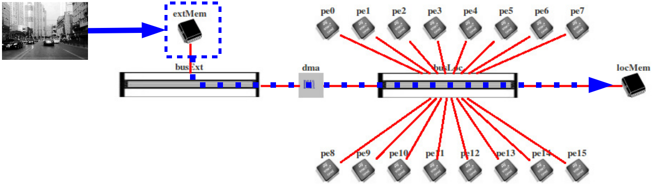
Algorithm

ForEach A[i][j]
Do
    B[i][j]:= compute(A[i][j])
EndDo

Video Processing on P2012

From an architectural point of view:

- The image resides initially in the off-chip
- Needs to be brought to local memory
- Then dispatched to processors for execution



Constraints:

- The whole image does not fit into the local memory
- Several alternatives for its splitting and transfer to local memories

Video Processing on P2012

Goal

Explore different design alternatives

Video Processing on P2012

Goal

Explore different design alternatives

We consider two different implementations:

1. Fetching data by bands.
2. Fetching data by blocks.

Video Processing on P2012

Goal

Explore different design alternatives

We consider two different implementations:

1. Fetching data by bands.
2. Fetching data by blocks.

We consider several platform alternatives:

- Processors number: 1,2,4,8 and 16
- Processors frequency: 200, 400 and 600 MHz

Video Processing on P2012

Bands Treatment

Video Processing on P2012

Bands Treatment

READ Band 1

Video Processing on P2012

Bands Treatment

Processor 1	Processor 2	Processor 3	Processor 4	Processor 5	Processor 6	Processor 7	Processor 8

Video Processing on P2012

Bands Treatment

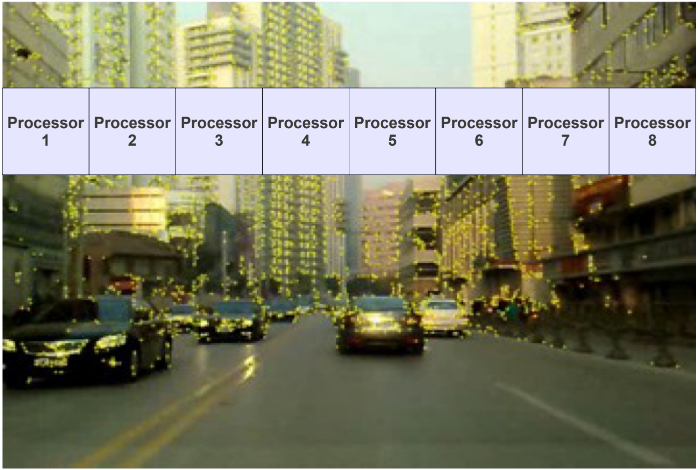
WRITE Band 1

Video Processing on P2012

Bands Treatment

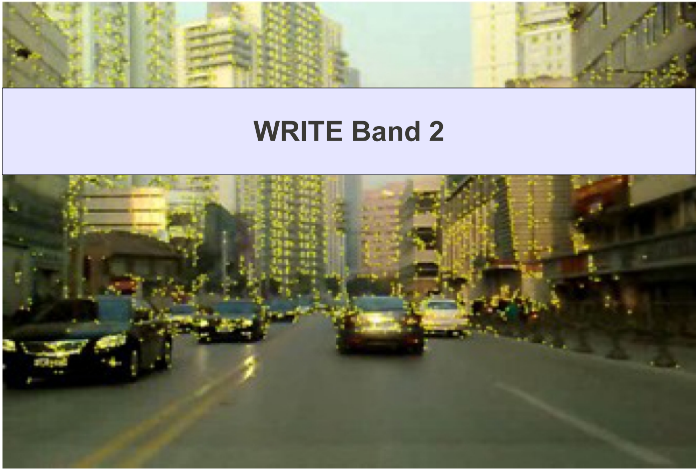
Video Processing on P2012

Bands Treatment



Video Processing on P2012

Bands Treatment

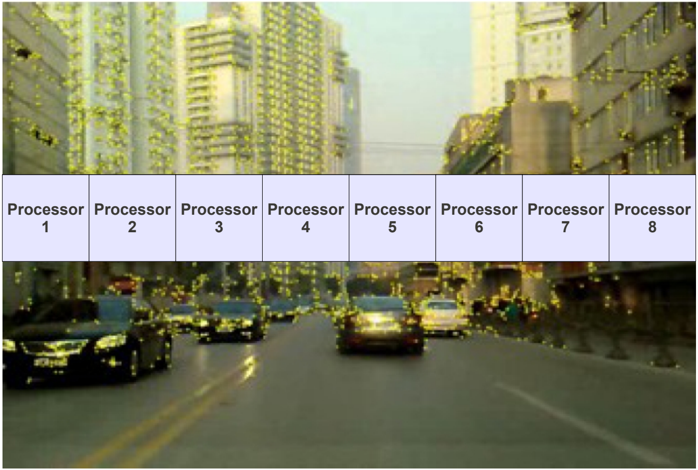


Video Processing on P2012

Bands Treatment

Video Processing on P2012

Bands Treatment



Video Processing on P2012

Bands Treatment

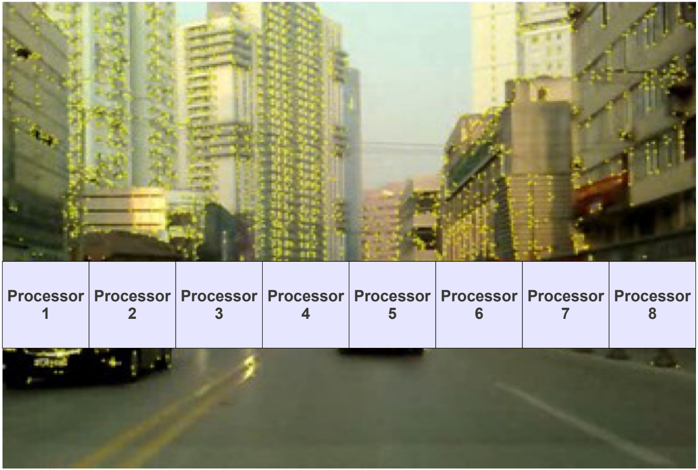
Video Processing on P2012

Bands Treatment

READ Band 4

Video Processing on P2012

Bands Treatment



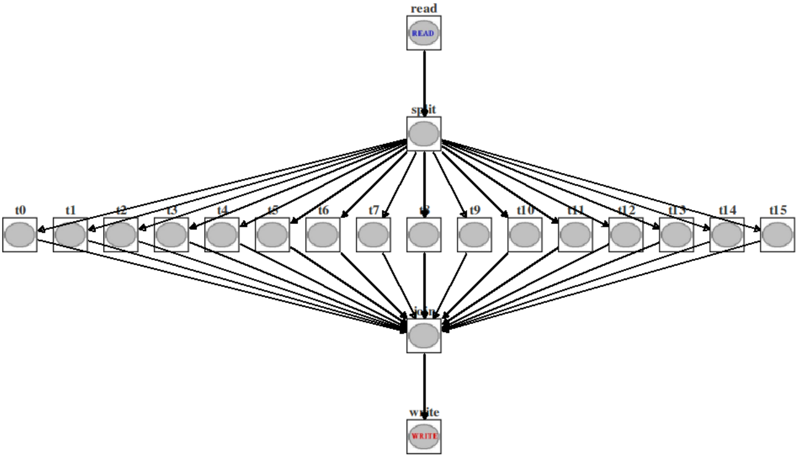
Video Processing on P2012

Bands Treatment

WRITE Band 4

Video Processing on P2012

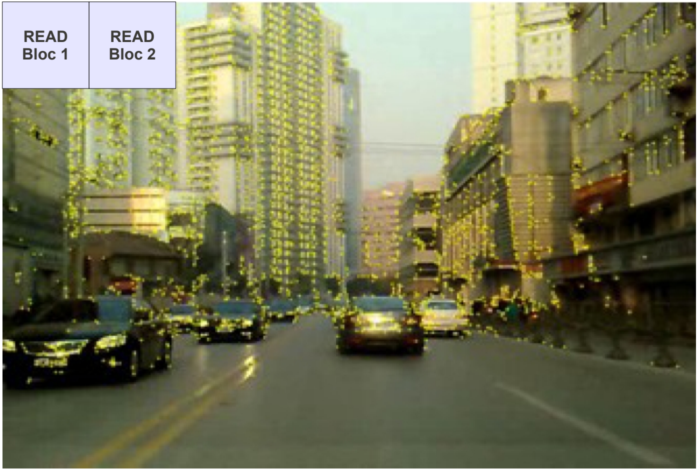
Bands Treatment Model



Blocs Treatment

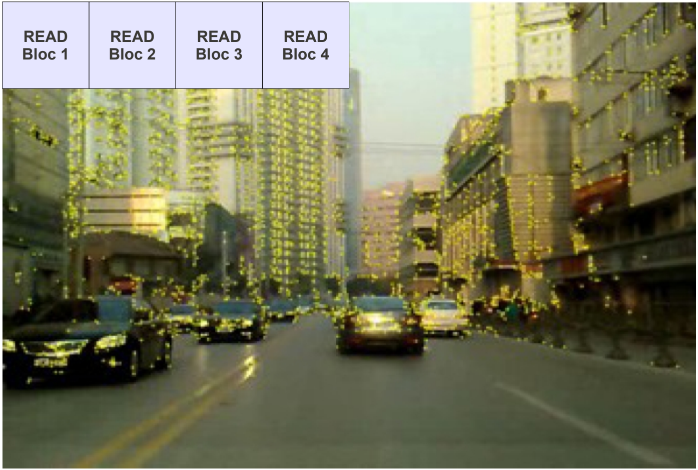
Blocs Treatment

Blocs Treatment



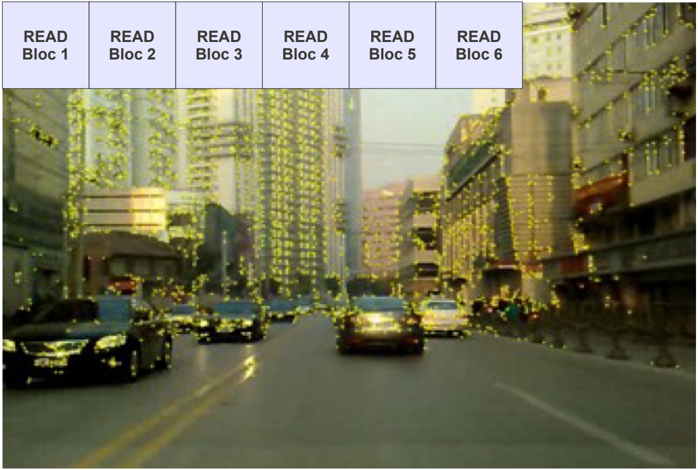
Blocs Treatment

Blocs Treatment



Blocs Treatment

Blocs Treatment



Blocs Treatment

Blocs Treatment

READ Bloc 1	READ Bloc 2	READ Bloc 3	READ Bloc 4	READ Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

Blocs Treatment

COMPUTE Bloc 1	READ Bloc 2	READ Bloc 3	READ Bloc 4	READ Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

Blocs Treatment

COMPUTE Bloc 1	COMPUTE Bloc 2	READ Bloc 3	READ Bloc 4	READ Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

Blocs Treatment

COMPUTE Bloc 1	COMPUTE Bloc 2	COMPUTE Bloc 3	READ Bloc 4	READ Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

Blocs Treatment

COMPUTE Bloc 1	COMPUTE Bloc 2	COMPUTE Bloc 3	COMPUTE Bloc 4	READ Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

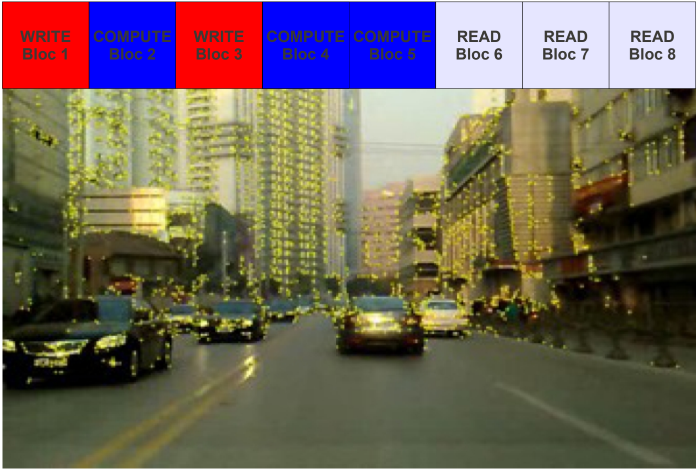
Blocs Treatment

WRITE Bloc 1	COMPUTE Bloc 2	COMPUTE Bloc 3	COMPUTE Bloc 4	READ Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

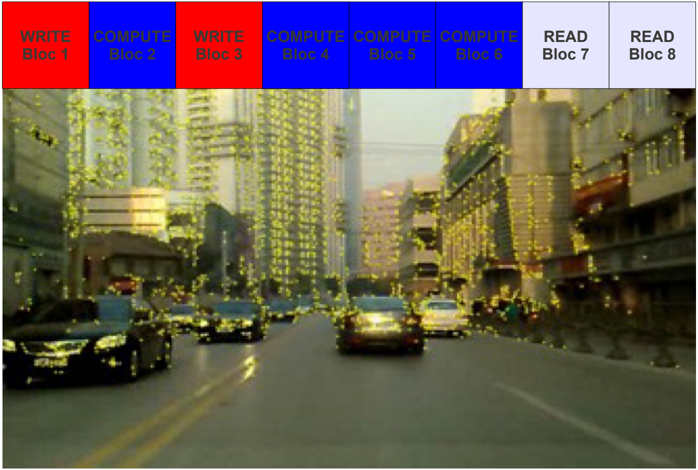
Blocs Treatment

WRITE Bloc 1	COMPUTE Bloc 2	COMPUTE Bloc 3	COMPUTE Bloc 4	COMPUTE Bloc 5	READ Bloc 6	READ Bloc 7	READ Bloc 8

Blocs Treatment

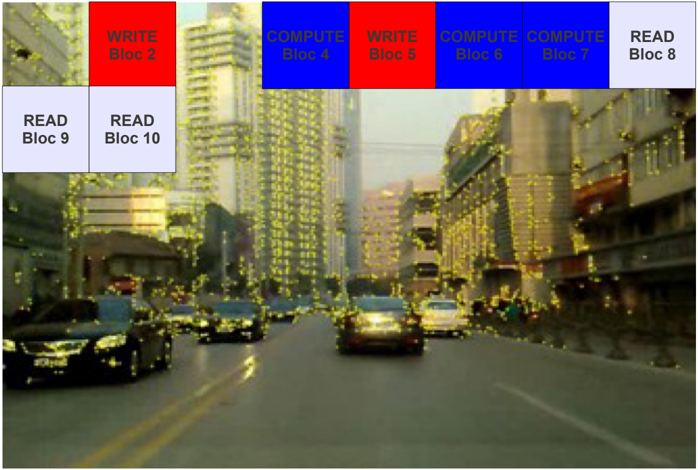


Blocs Treatment



Blocs Treatment

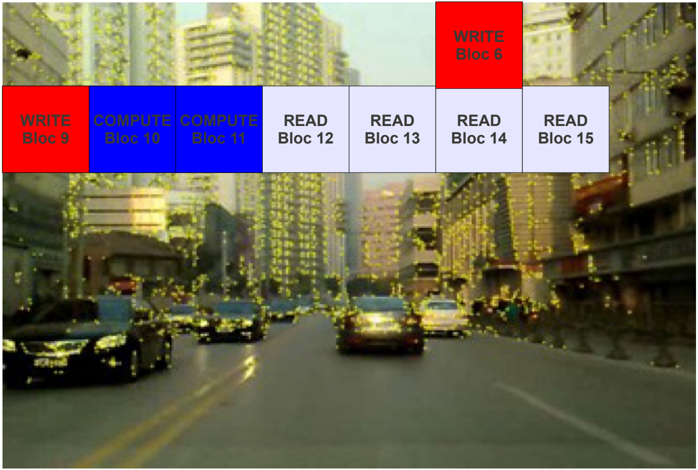
Blocs Treatment



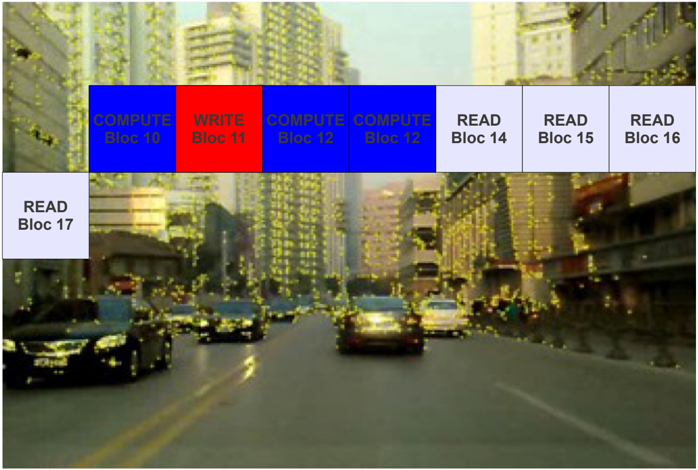
Blocs Treatment

Blocs Treatment

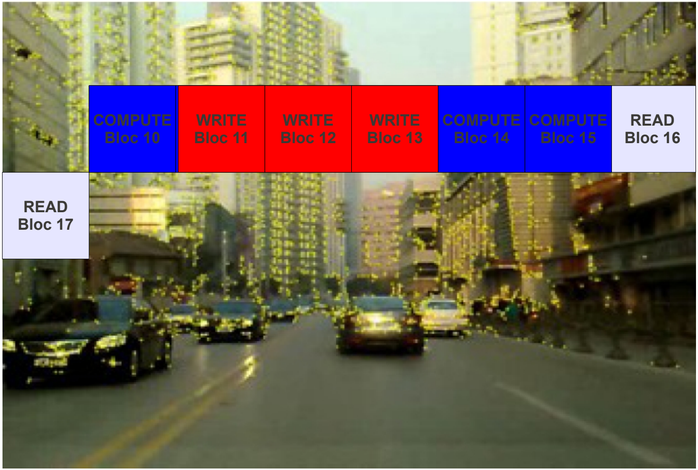
Blocs Treatment



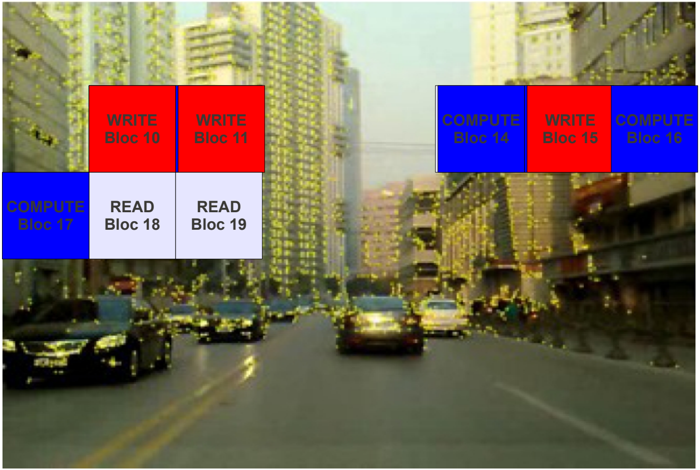
Blocs Treatment



Blocs Treatment

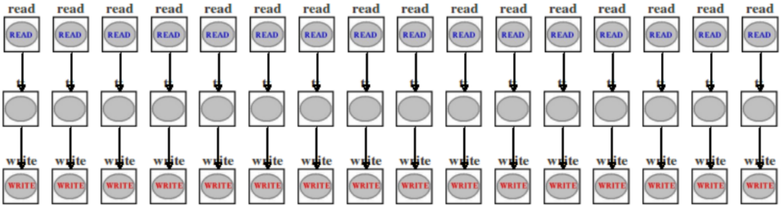


Blocs Treatment



Video Processing on P2012

Blocs Treatment Model

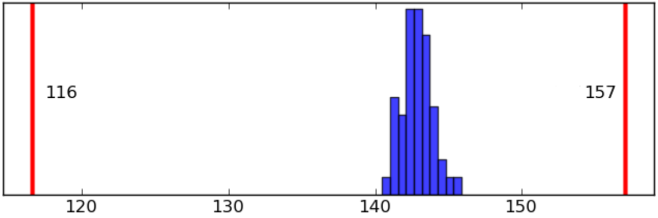


- Transfer smaller chunks of the image independently.

Video Processing on P2012: Evaluation

Worst-Case vs Statistics

- Compare results from simulation and reachability analysis:

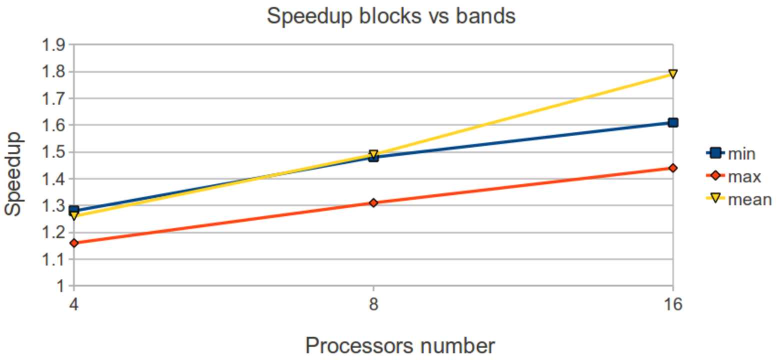


Response time of processing one image.

  - Bands treatment
  - 4 processors (600 MHz)

Video Processing on P2012: Evaluation
Bands vs blocks

- Compare results of the two variant of the application:



Video Processing on P2012: Evaluation

Bands vs blocks

- Compare results of the two variant of the application:

Fetching data with bands:

- Requires a synchronization for writing back the results.
- Latency coming from the slower computation.

Video Processing on P2012: Evaluation

Bands vs blocks

- Compare results of the two variant of the application:

Fetching data with bands:

- Requires a synchronization for writing back the results.
- Latency coming from the slower computation.

Fetching data with blocks is more flexible:

- Once a computation terminates, the transfer of the next block can start immediately.
- Communication becomes more fluid.

Video Processing on P2012: Evaluation

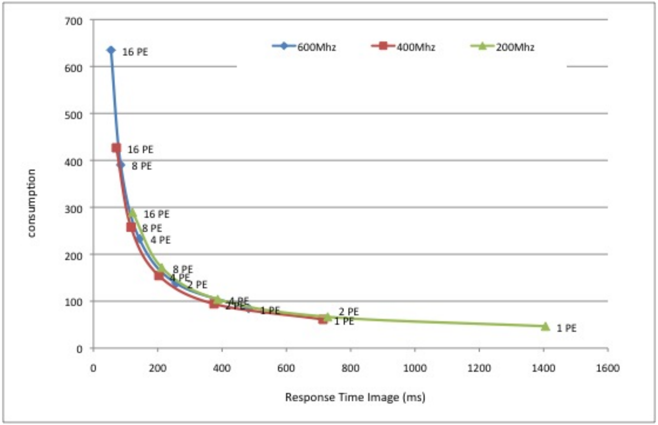
Power Consumption

We compare different configuration of the platform to get the trade-offs between response time and power consumption.

Video Processing on P2012: Evaluation

Power Consumption

We compare different configuration of the platform to get the trade-offs between response time and power consumption.



Duration Probabilistic Automata

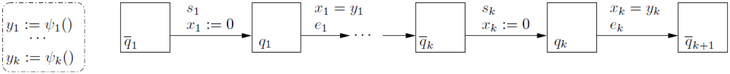
- We move to the theoretical part of the thesis:
  - Exact computation of expected performance of scheduling policies.

Duration Probabilistic Automata

- We move to the theoretical part of the thesis:
  - Exact computation of expected performance of scheduling policies.
- We use DPA to model scheduling problems such as job-shop, where task durations are uncertain.

Duration Probabilistic Automata

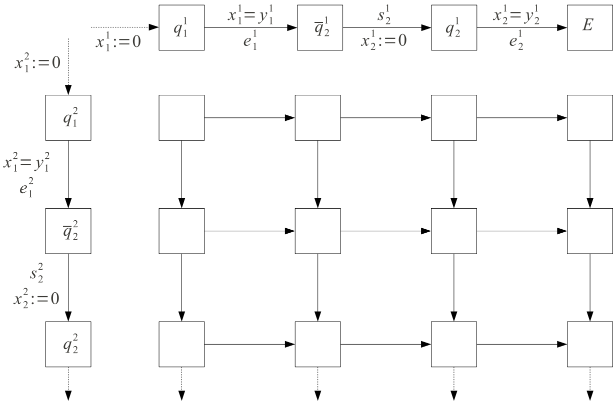
- We move to the theoretical part of the thesis:
  - Exact computation of expected performance of scheduling policies.
- We use DPA to model scheduling problems such as job-shop, where task durations are uncertain.
- Formally they are timed automata with probabilistic delay:



  - Waiting states $\overline{q_i}$, active states q_i
  - **Start** transitions s_i which resets a clock x_i to measure time elapsed in active state q_i
  - **End** transitions e_i guarded by a temporal condition:
    - interpreted probabilistically as uniform distribution

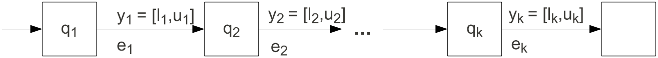
Composition

- DPA can be combined in parallel to express partially-independent processes, sometimes competing with each other:



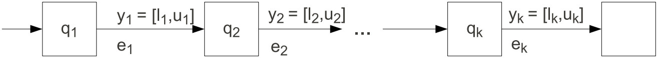
Process

- We consider only finite acyclic processes



Process

- We consider only finite acyclic processes

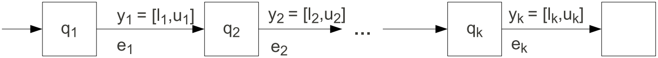


We consider two execution frameworks:

- *Independent execution*
- *Coordinated execution* :
  - conflicts on some steps
    resolved by a scheduler that guarantees mutual exclusion

Process

- We consider only finite acyclic processes



We consider two execution frameworks:

- *Independent execution*
- *Coordinated execution* :
  - conflicts on some steps
    resolved by a scheduler that guarantees mutual exclusion

Goal

- **Compute probabilities of qualitative behavior**
- **Compare the expected performance of scheduling policies**

Example Single Process

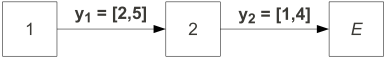
- Consider the following process:

$$\boxed{1} \xrightarrow{y_1 = [2,5]} \boxed{2} \xrightarrow{y_2 = [1,4]} \boxed{E}$$

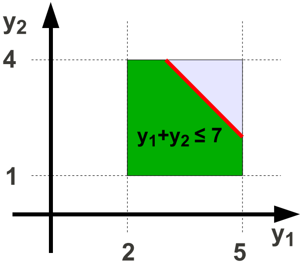
- What is the probability to terminate before 7 time units?

Example Single Process

- Consider the following process:



- What is the probability to terminate before 7 time units?

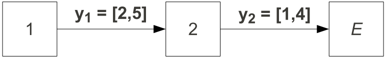


- Characterized by:
  $\varphi = y_1 + y_2 < 7$.

1. Compute the volume V_φ
   - subset of D satisfying φ.
2. Compute the volume V_D of the duration space.
3. Probability: $\frac{V_\varphi}{V_D} = \frac{7}{9} \approx 0.78$

Example Single Process

- Consider the following process:



- What is the probability to terminate before 7 time units?

We work on time-stamp space:

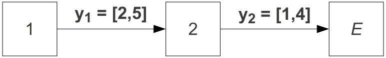
- constraints are of restricted form (difference constraints only)

Time-stamp space:

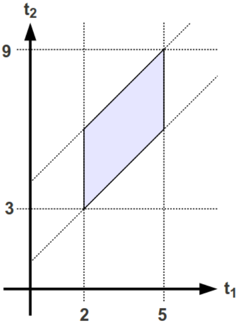
- t_j is the *absolute time* of e_j:
  - $t_j = y_1 + y_2 + \cdots + y_j$
  - $y_j = t_j - t_{j-1}$

Example Single Process

- Consider the following process:



- What is the probability to terminate before 7 time units?



The time-stamp space is a zone Z_C defined by:

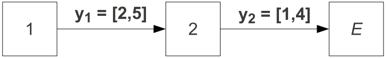
$$2 \leq t_1 \leq 5$$

$$1 \leq t_2 - t_1 \leq 4$$

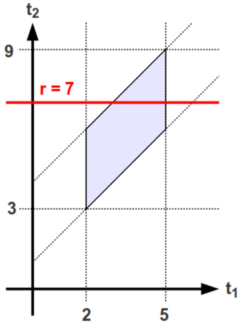
$$3 \leq t_2 \leq 9$$

Example Single Process

- Consider the following process:



- What is the probability to terminate before 7 time units?

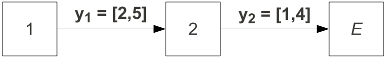


- The termination time corresponds to t_2:
  - time-stamp of the last step.
- The behavior corresponding to a termination time ≤ 7 is defined by:

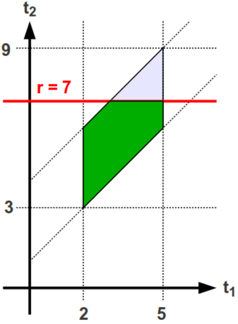
$$Z_r = Z_C \wedge t_2 \leq 7$$

Example Single Process

- Consider the following process:



- What is the probability to terminate before 7 time units?

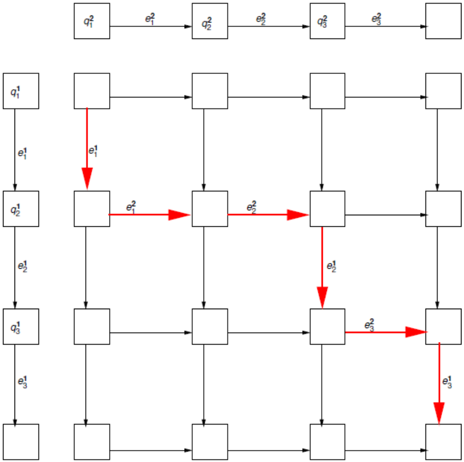


Probability:

$$\frac{|Z_r|}{|Z_C|} = \frac{7}{9} \approx 0.78$$

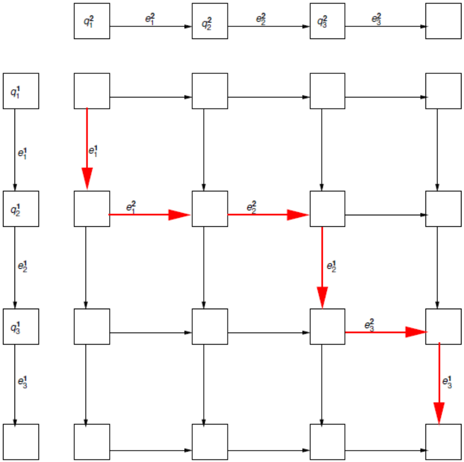
Processes in parallel

A global behavior: $w = e_1^1 \; e_1^2 \; e_2^2 \; e_2^1 \; e_3^2 \; e_3^1$



Processes in parallel

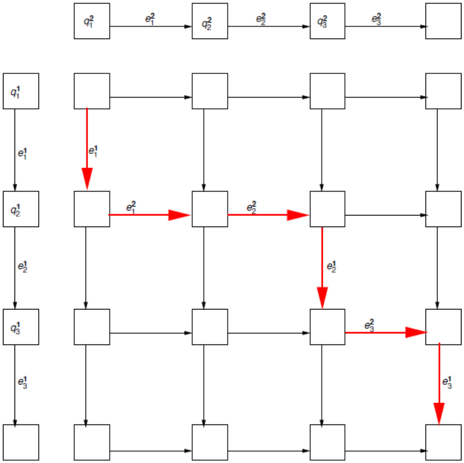
A global behavior: $w = e^1_1 \; e^2_1 \; e^2_2 \; e^1_2 \; e^2_3 \; e^1_3$



- What is the probability of this path ?

Processes in parallel

A global behavior: $w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$



- What is the probability of this path ?
- What is the most probable path ?

Probability of Qualitative behavior

To compute the probability of a qualitative behavior w, i.e. the probability that events occur in a particular order:

Probability of Qualitative behavior

To compute the probability of a qualitative behavior w, i.e. the probability that events occur in a particular order:

1. characterize the subset $Z_w \subseteq \mathcal{C}$ consisting of all instances of time-stamps that yield this behavior.

Probability of Qualitative behavior

To compute the probability of a qualitative behavior w, i.e. the probability that events occur in a particular order:

1. characterize the subset $Z_w \subseteq \mathcal{C}$ consisting of all instances of time-stamps that yield this behavior.
2. Compute the volume of Z_w divided by the volume of $\mathcal{C}$

Probability of Qualitative behavior

To compute the probability of a qualitative behavior w, i.e. the probability that events occur in a particular order:

1. characterize the subset $Z_w \subseteq \mathcal{C}$ consisting of all instances of time-stamps that yield this behavior.
2. Compute the volume of Z_w divided by the volume of $\mathcal{C}$

- The whole time-stamp space is defined by:

$$\varphi_{\mathcal{C}} : \bigwedge_{i \in N} \bigwedge_{j \in K} a_j^i \leq t_j^i - t_{j-1}^i \leq b_j^i$$

Probability of Qualitative behavior

To compute the probability of a qualitative behavior w, i.e. the probability that events occur in a particular order:

1. characterize the subset $Z_w \subseteq \mathcal{C}$ consisting of all instances of time-stamps that yield this behavior.
2. Compute the volume of Z_w divided by the volume of $\mathcal{C}$

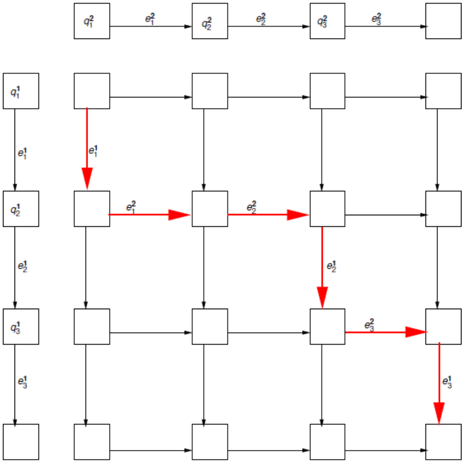
- The whole time-stamp space is defined by:

$$\varphi_{\mathcal{C}} : \bigwedge_{i \in N} \bigwedge_{j \in K} a_j^{\mathbf{i}} \leq t_j^{\mathbf{i}} - t_{j-1}^{\mathbf{i}} \leq b_j^{\mathbf{i}}$$

- The zone Z_w is characterized by adding constraints that specify the particular order of event in w.

Example

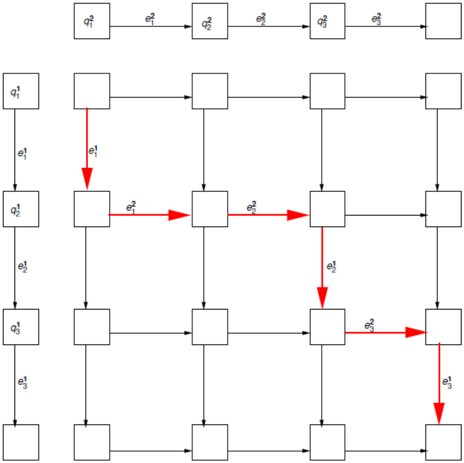
$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



- Characterize $\varphi_{\mathcal{C}}$

Example

$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



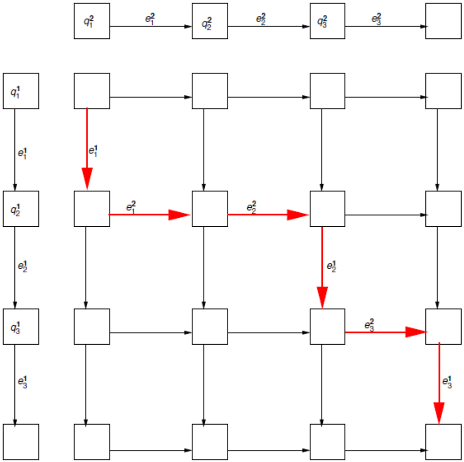
- Characterize $\varphi_{\mathcal{C}}$

P^1
$a_1^1 \leq t_1^1 \leq b_1^1$
$a_2^1 \leq t_2^1 - t_1^1 \leq b_2^1$
$a_3^1 \leq t_3^1 - t_2^1 \leq b_3^1$

P^2
$a_1^2 \leq t_1^2 \leq b_1^2$
$a_2^2 \leq t_2^2 - t_1^2 \leq b_2^2$
$a_3^2 \leq t_3^2 - t_2^2 \leq b_3^2$

Example

$$w = e_1^1 \; e_1^2 \; e_2^2 \; e_2^1 \; e_3^2 \; e_3^1$$

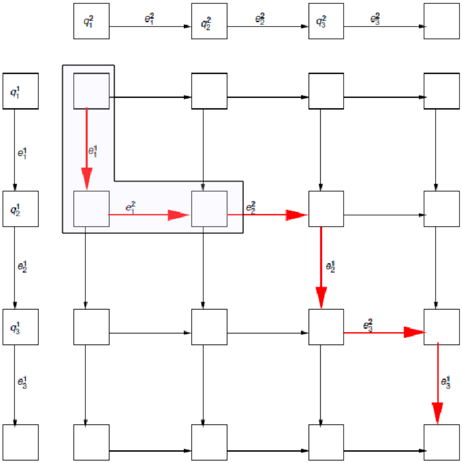


- Characterize $\varphi_{\mathcal{C}}$
- Add constraints according to order induced by w:

$$\varphi_w : \varphi_{\mathcal{C}}$$

Example

$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



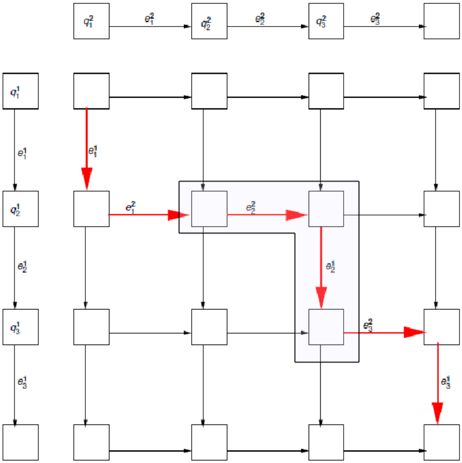
- Characterize $\varphi_{\mathcal{C}}$
- Add constraints according to order induced by w:

$e_1^1\ e_1^2$

$$\varphi_w : \varphi_{\mathcal{C}} \wedge (t_1^1 < t_1^2)$$

Example

$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



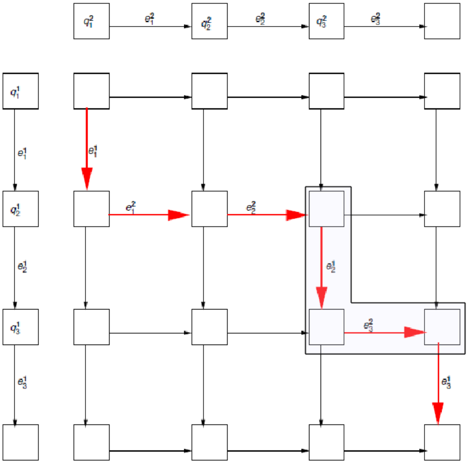
- Characterize $\varphi_{\mathcal{C}}$
- Add constraints according to order induced by w:

$$e_1^1\ e_1^2\ e_2^2\ e_2^1$$

$$\varphi_w : \varphi_{\mathcal{C}} \wedge (t_1^1 < t_1^2) \wedge (t_2^2 < t_2^1)$$

Example

$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



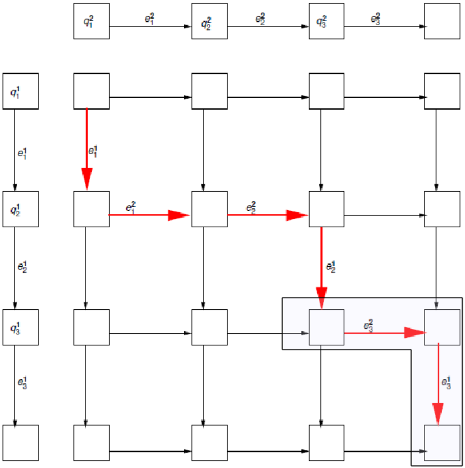
- Characterize $\varphi_{\mathcal{C}}$
- Add constraints according to order induced by w:

$$e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2$$

$$\varphi_w : \varphi_{\mathcal{C}} \wedge (t_1^1 < t_1^2) \wedge (t_2^2 < t_2^1) \wedge (t_2^1 < t_3^2)$$

Example

$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



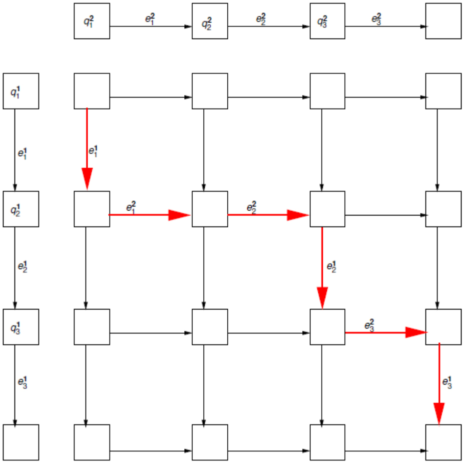
- Characterize $\varphi_{\mathcal{C}}$
- Add constraints according to order induced by w:

$$e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$

$$\varphi_w : \varphi_{\mathcal{C}} \wedge (t_1^1 < t_1^2) \wedge (t_2^2 < t_2^1)$$
$$\wedge (t_2^1 < t_3^2) \wedge (t_3^2 < t_3^1)$$

Example

$$w = e_1^1\ e_1^2\ e_2^2\ e_2^1\ e_3^2\ e_3^1$$



- Characterize $\varphi_{\mathcal{C}}$
- Add constraints according to order induced by w:

The volume of Z_w is then computed by integration:

$$|Z_w| = \int_{Z_w} 1$$

Computability

Theorem

The probability of a qualitative behavior in a system of acyclic stochastic sequential processes with uniform probabilistic durations is computable.

- The global termination time of a behavior is $\Theta = \max\{t_k^{\mathbf{1}}, \ldots, t_k^{\mathbf{n}}\}$
- In any behavior, if process $P^{\mathbf{i}}$ is the last to terminate then the total termination time is $t_k^{\mathbf{i}}$.
- The expected termination time is:

$$\mathbb{E}(\Theta) = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{n} \sum_{w=w' e_k^{\mathbf{i}}} \int_{Z_w} t_k^{\mathbf{i}}.$$

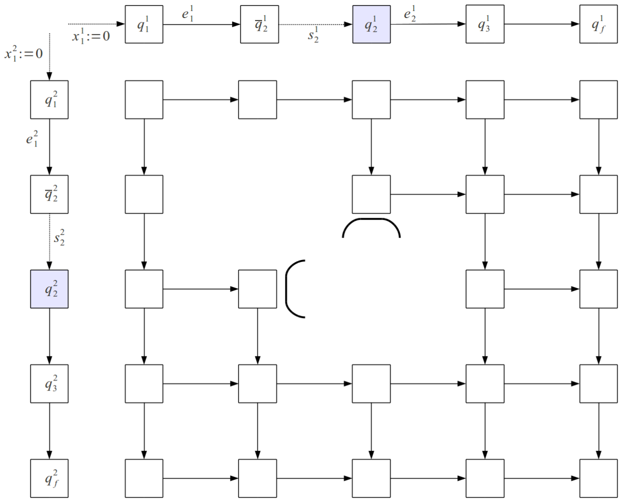
Conflicts and Scheduler

- We assume that steps of different processes can be in *conflict* as they require the same bounded resource
- A scheduler should decide to whom to give the resource first based on some policy

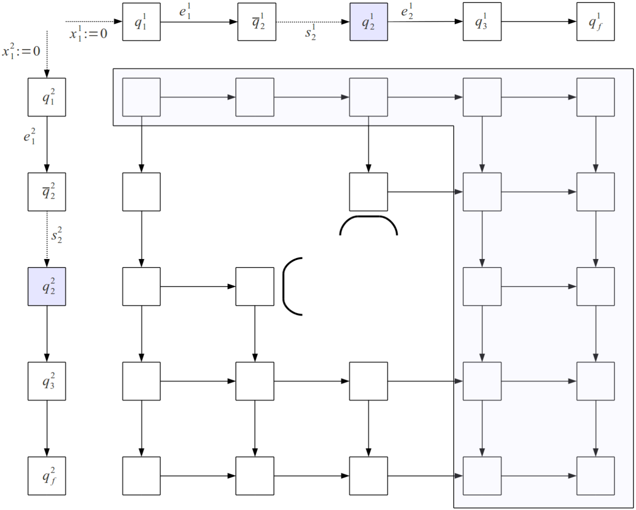
Qualitative evaluation of a scheduler

- compute the expected termination time over all runs
- corresponds to the evaluation of a finite number of different qualitative behaviors
- dependent on the choices taken to solve conflicts

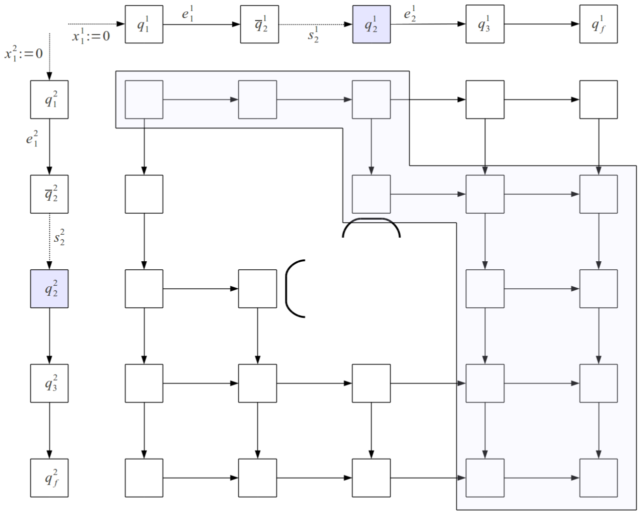
FIFO scheduler



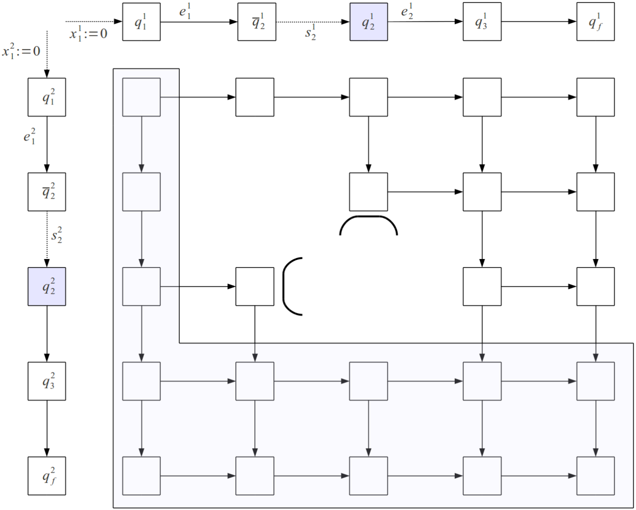
FIFO scheduler



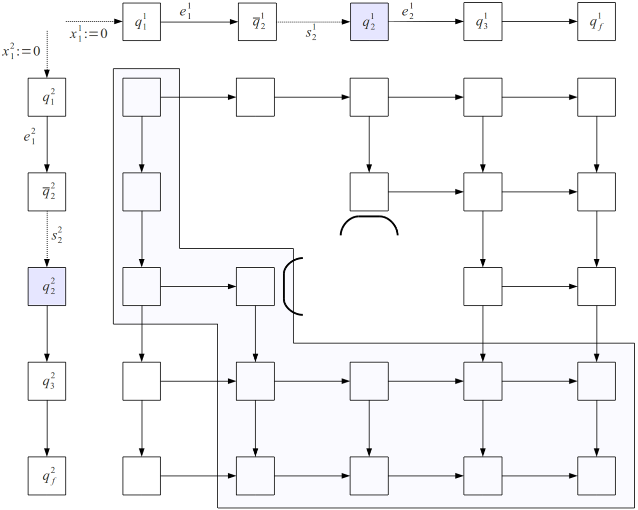
FIFO scheduler



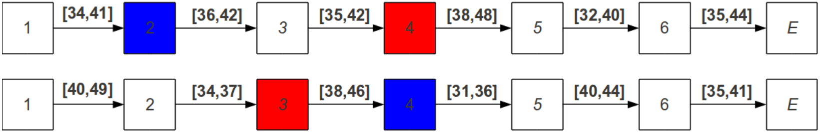
FIFO scheduler



FIFO scheduler

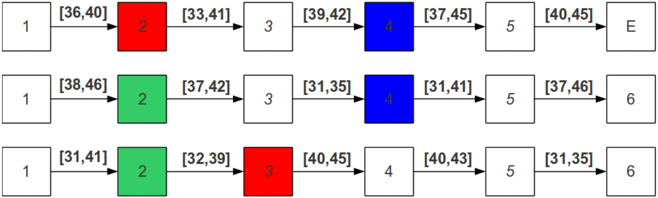


FIFO Scheduler: 2 processes with 2 conflicts



Case	Probability	Expected Time
1	0.76160	242.13086
2	0.15962	237.73817
3	0.07084	238.54240
4	0.00794	233.20148
Global	1	**241.105**

FIFO Scheduler: 3 processes with 3 conflicts



Case	Probability	Expected Time
1	0.00260	234.50898
2	0.00436	233.28736
3	0.75366	234.54334
4	0.00000	227.77534
5	0.18376	231.99327
6	0.00633	228.96075
7	0.02428	233.50729
8	0.02501	232.10406
Global	1	**233.948**

Implementation

A prototype tool:

- Computes the zone for each utilization scenario, using the DBM library of IF to simplify and check emptiness
- Performs integration over the non-empty zones to compute probabilities and expected termination time
- Integration uses high-precision arithmetic (GMP library) to avoid rounding errors
- A heuristic to determine the order of variable elimination integration based on a fast estimation of their ranges
- can solve (in <3 minutes) problems with $(n,k) = (1,63),(2,12),(4,6),(5,4)$ with two or three conflicts.

Synthesis

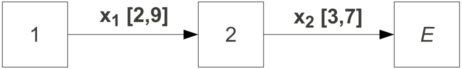
Problem

Synthesis of optimal schedulers that minimize the expected termination time.

Solution

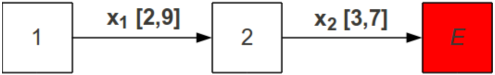
- We develop techniques for value iteration:
  - Exact computation of a value function (expected time to terminate)
  - Optimal action for any point in the extended state space (including clock value).
- We define a *stochastic time-to-go* function
- We show how this *density function* can be computed backwards from the final state.

Single Process Example



- Backward computation of the *expected time to go*

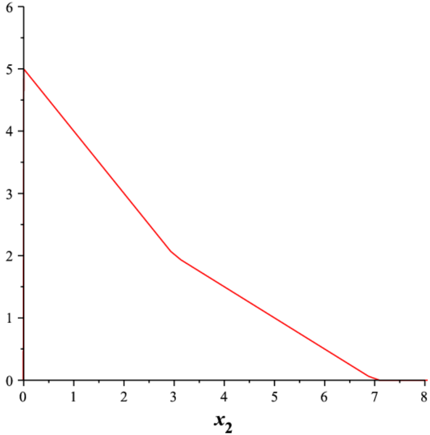
Single Process Example



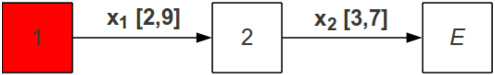
Single Process Example

Expected time-to-go

$$\begin{cases} 5 - x_2 & \text{if } \; 0 \leq x_2 \leq 3 \\ \frac{7}{2} - \frac{1}{2} \cdot x_2 & \text{if } \; 3 \leq x_2 \leq 7 \end{cases}$$

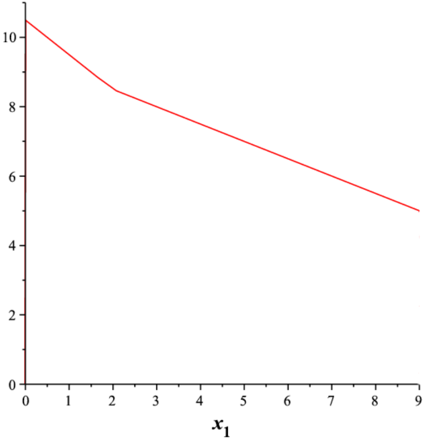


Single Process Example

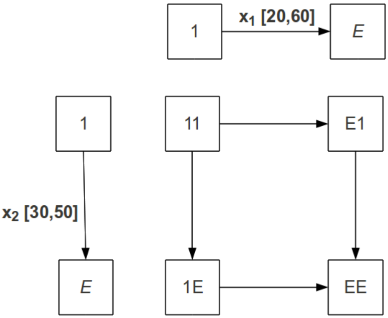


Expected time-to-go

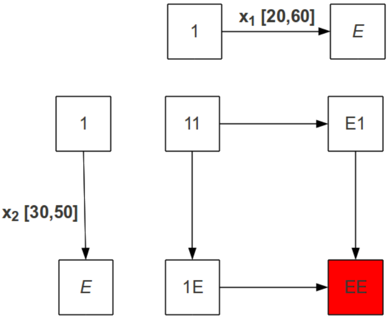
$$\begin{cases} \frac{21}{2} - x_1 & \text{if } \; 0 \leq x_1 \leq 2 \\ \frac{19}{2} - \frac{1}{2} \cdot x_1 & \text{if } \; 2 \leq x_1 \leq 9 \end{cases}$$



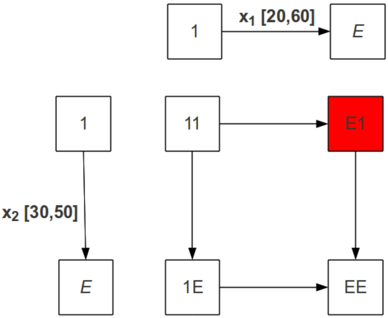
DPA Example

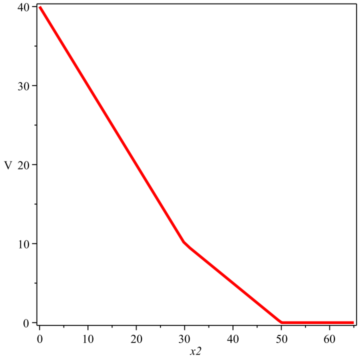


DPA Example

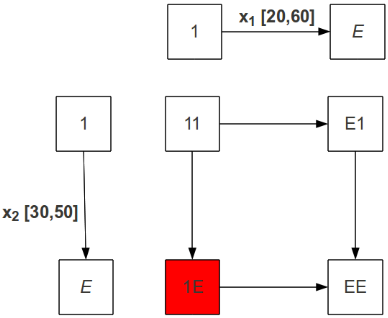


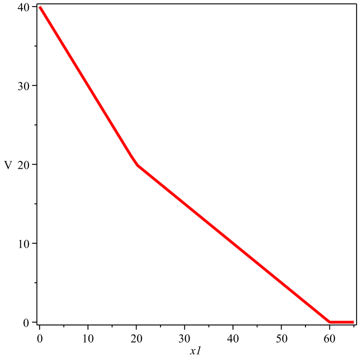
DPA Example



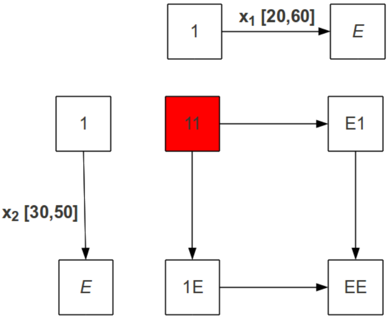


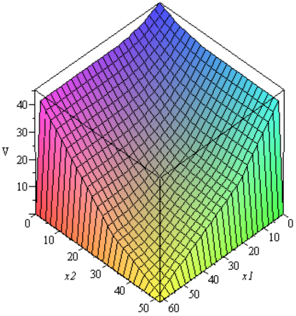
DPA Example





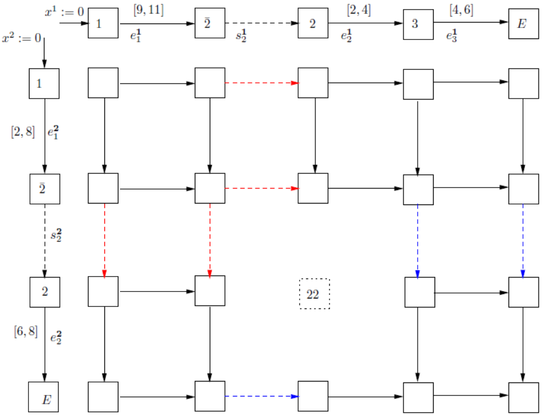
DPA Example



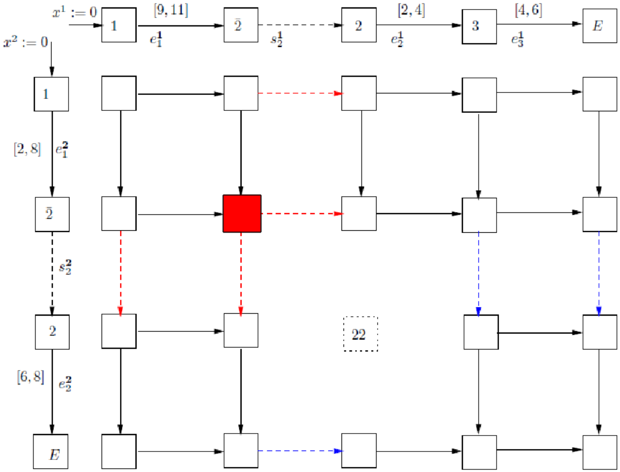


Conflicts and Schedulers

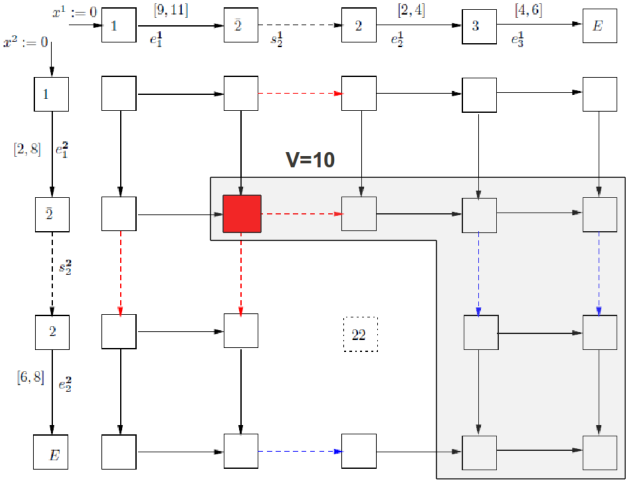
We are now interested in computing *expected time optimal scheduler*.



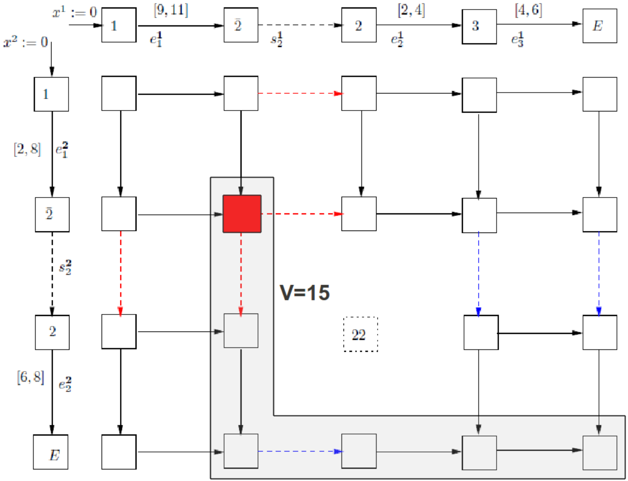
Conflicts and Schedulers



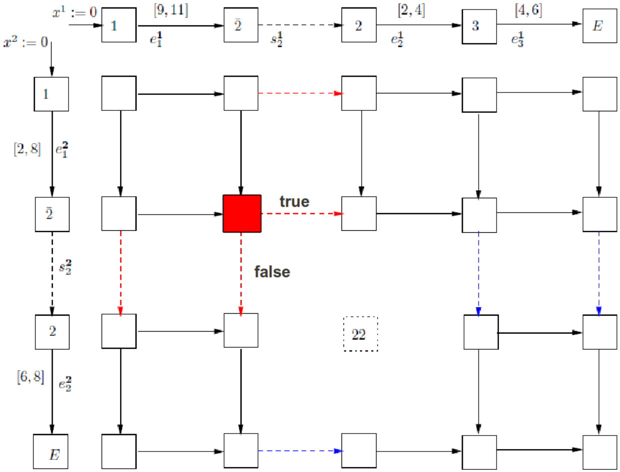
Conflicts and Schedulers



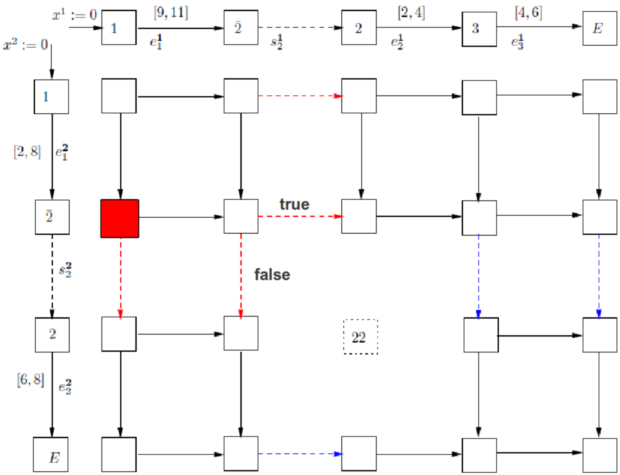
Conflicts and Schedulers



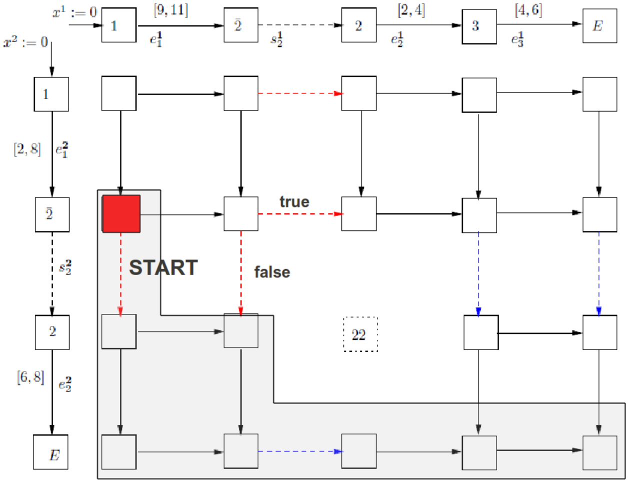
Conflicts and Schedulers



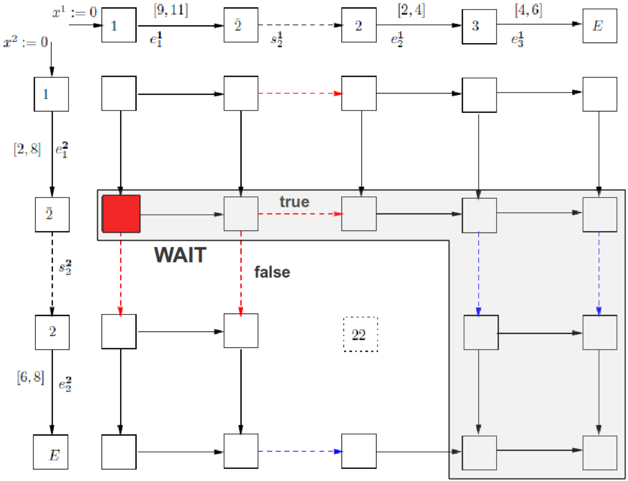
Conflicts and Schedulers



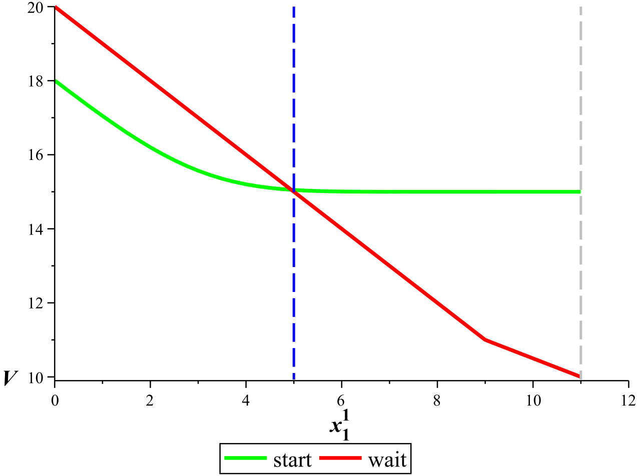
Conflicts and Schedulers



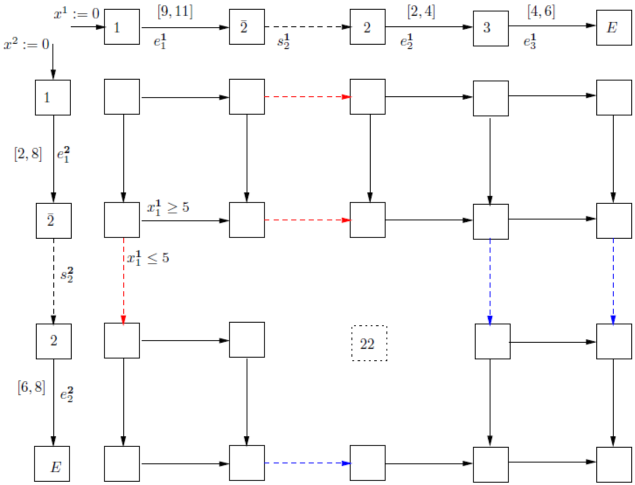
Conflicts and Schedulers



Conflicts and Schedulers



Conflicts and Schedulers



Implementation

- We show that the density belongs to a particular form which is *zone-polynomial function*.
- Implementation of a library for all needed operations (convolution, max, . . .) on zone polynomial functions.
- Implementation of the optimization process is still ongoing work.

Conclusion

- We provide a tool-supported framework for Design-Space Exploration based on abstract model
- The framework and analysis techniques have been implemented into an extensible toolset: DeSpEx
- We demonstrate its functionality on several case studies
- We provided new results concerning analysis and scheduler synthesis for continuous-time stochastic processes where task durations are distributed uniformly

Perspectives

DeSpEx:

- Enrich the modeling framework
- Partially automate the design space exploration
- Systematic way to populate the high-level models with performances numbers

DPA:

- Complete implementation of the optimization process
- Extension to richer models: other distributions, cyclic systems
- Comparison between analytical and statistical results

Perspectives

DeSpEx:

- Enrich the modeling framework
- Partially automate the design space exploration
- Systematic way to populate the high-level models with performances numbers

DPA:

- Complete implementation of the optimization process
- Extension to richer models: other distributions, cyclic systems
- Comparison between analytical and statistical results

Thank you for your attention