



Enhancing the Analysis of Large Multimedia Applications Execution Traces with FrameMiner

C. Kamdem, L. Fopa, N. Ibrahim, A. Termier, M.-C. Rousset, T. Washio

christiane.kamdem-kengne@imag.fr,leon-constantin.fopa@imag.fr

December 10th, 2012 PTDM@ICDM'12

Context

- Embedded systems
 - × MPSoc: System on Chips with multiple processors



Context

- Embedded systems
 - × MPSoc: System on Chips with multiple processors



- × Video decoding
- × Challenging issue using MPSoc



Context



- Embedded systems
 - × MPSoc: System on Chips with multiple processors

• Multimedia applications

- \times Video decoding
- × Challenging issue using MPSoc
- Traditional debugging techniques are not optimal

Context



- Embedded systems
 - × MPSoc: System on Chips with multiple processors
- Multimedia applications
 - \times Video decoding
 - × Challenging issue using MPSoc

Context



• Huge amount of traces

 \implies Example: 7GB of traces for less than 5mn of video decoding



2 Problem statement

FrameMiner Process

4 Evaluation



State of the art - Traces Analysis

✓ Visualization [Stein,2003],[Vite,2010]



State of the art - Traces Analysis

✓ Visualization [Stein,2003],[Vite,2010]

Information overload



Figure: Paje, An interactive visualization tool



State of the art - Traces Analysis

✓ Visualization [Stein,2003],[Vite,2010]

Information overload



Figure: Paje, An interactive visualization tool

 \implies Reduce size of execution traces

State of the art - Traces Analysis

- ✓ Visualization [Stein,2003],[Vite,2010]
 - Information overload
 - \implies Reduce size of execution traces
- ✓ Reduction [Dugerdil,2007],[chan et al.,2003]

State of the art - Traces Analysis

- ✓ Visualization [Stein,2003],[Vite,2010]
 - Information overload
 - \implies Reduce size of execution traces
- ✓ Reduction [Dugerdil,2007],[chan et al.,2003]
 - Not always representative of the entire trace

State of the art - Traces Analysis

✓ Visualization [Stein,2003],[Vite,2010]

- Information overload
 - \implies Reduce size of execution traces
- ✓ Reduction [Dugerdil,2007],[chan et al.,2003]
 - Not always representative of the entire trace
 - \implies Abstract execution traces

Our goal: Abstraction

| 9.5054 9.5073 9.5081 9.5083 9.5084 9.5096 | GetFrame ExitGet CS produc Interrupt Period ExitI Interrupt Soft |
|--|---|
| 9.5102 | ExitS |
| 9.5127 | ExitII |
| 9.5154 | CheckData |
| 9.5260 | FillJob |
| 9.5715 | CS VGA |
| 9.5845 | CS produc |
| 9.5974 | GetFrame |
| 9.6012 | ExitGet |
| 9.6125 | Interrupt Hand |
| 9.6155 | Exitl |
| 9.6234 | ExitIT |
| 9.6315 | Interrupt Soft |
| 9.6405 | Interrupt Period |
| 9.6483 | Exitl |
| 9.6514 | Interrupt Soft |
| 9.6622 | ExitS |
| 9.6715 | ExitIT |
| 9.6811 | CS produc |
| 9.6898 | CheckData |
| 9.6932 | FillJob |
| 9.6987 | CS VGA |
| 9.7001 | Interrupt Hand |

Our goal: Abstraction

| | Init |
|--|---|
| 9.5081 9.5083 9.5084 9.5096 9.5102 9.5127 9.5124 | CS produc Interrupt Period Exitl Interrupt Soft ExitS ExitIT Check Data |
| 9.5260 9.5715 9.5845 9.5974 | FillJob CS VGA CS produc GetFrame |
| 9.6012 9.6125 9.6155 9.6234 9.6315 | ExitGet Interrupt Hand ExitI ExitIT Interrupt Soft |
| 9.6405 9.6483 9.6514 9.6622 9.6715 | Interrupt Period Exitl Interrupt Soft ExitS ExitIT |
| 9.6811 9.6898 9.6932 9.6987 9.7001 | CS produc CheckData FillJob CS VGA Interrupt Hand |

Our goal: Abstraction

| | Init |
|---|--|
| | Interruption |
| 9.5154 9.5260 9.5715 9.5845 9.5974 9.6012 9.6125 9.6125 9.6234 9.6315 9.6483 9.6514 9.66483 9.6622 9.6715 9.6848 9.6838 9.6838 9.6832 9.6937 | CheckData Fillubo CS produc GetFrame ExitGet Interrupt Hand ExitI Interrupt Soft Interrupt Soft Interrupt Soft ExitI CS produc CS produc CS produc CS produc CS VGA Interrupt Hand |

Our goal: Abstraction



Our goal: Abstraction

 Abstract series of low level events: blocks

 \implies blocks are automatically discovered



Our goal: Abstraction

 Abstract series of low level events: blocks

 \implies blocks are automatically discovered

• A frame is identified by two events: *start* and *end* events

| | | | _ |
|-----------------------|--------|------------------|---|
| | 9.5054 | GetFrame | T |
| | 9.5073 | ExitGet | |
| | 9,5081 | CS produc | |
| <i>F</i> ₁ | 9.5083 | Interrupt Period | |
| F_{1} | 9.5084 | Exitl | |
| | 9.5096 | Interrupt Soft | |
| | 9.5102 | ExitS | |
| | 9.5127 | ExitIT | |
| | 9.5154 | CheckData | |
| | 9.5260 | FillJob | |
| | 9.5715 | CS VGA | Τ |
| | 9.5845 | CS produc | |
| 1 | 9.5974 | GetFrame | Т |
| | 9.6012 | ExitGet | |
| | 9.6125 | Interrupt Hand | |
| | 9.6155 | Exit | |
| | 9.6234 | ExitIT | |
| F | 9.6315 | Interrupt Soft | |
| F ₁ | 9.6405 | Interrupt Period | |
| | 9.6483 | Exitl | |
| | 9.6514 | Interrupt Soft | |
| | 9.6622 | ExitS | |
| | 9.6715 | ExitIT | |
| | 9.6811 | CS produc | |
| | 9.6898 | CheckData | |
| | 9.6932 | FillJob | 4 |
| | 9.6987 | CS VGA | T |
| | 9,7001 | Interrupt Hand | 1 |

Our goal: Abstraction

- Abstract series of low level events: blocks
 - \implies blocks are automatically discovered
- A frame identified by two events: *start* and *end* events



Our goal: Abstraction

- Abstract series of low level events: blocks
 - \implies blocks are automatically discovered
- A frame identified by two events: *start* and *end* events



Our problem: Rewrite each frame into a short description with a small set of blocks

Definitions

example

Let $S = \{B_1, B_2, B_3, B_4\}$ with $B_1 = \langle GetFrame, exitGet \rangle$ $B_2 = \langle InterrupPeriod, exitI, InterruptSoft, exitS, exitT \rangle$

| | 1 | 9.5054 | GetFrame |
|-----|---|--------|------------------|
| | | 9.5073 | ExitGet |
| | | 9.5081 | CS produc |
| | | 9.5083 | Interrupt Period |
| F | 1 | 9.5084 | Exitl |
| | | 9.5096 | Interrupt Soft |
| | | 9.5102 | ExitS |
| | | 9.5127 | ExitIT |
| F | | 9.5154 | CheckData |
| | + | 9.5260 | FillJob |
| | 1 | 9.5715 | CS VGA |
| | | 9.5845 | CS produc |
| 1 | 1 | 9.5974 | GetFrame |
| | | 9.6012 | ExitGet |
| | | 9.6125 | Interrupt Hand |
| | | 9.6155 | Exitl |
| | | 9.6234 | ExitIT |
| F | | 9.6315 | Interrupt Soft |
| r 2 | | 9.6405 | Interrupt Period |
| | | 9.6483 | Exitl |
| | | 9.6514 | Interrupt Soft |
| | | 9.6622 | ExitS |
| | | 9.6715 | ExitIT |
| | | 9.6811 | CS produc |
| | | 9.6898 | CheckData |
| | | 9.6932 | FillJob |
| | | 9.6987 | CS VGA |
| | | 9.7001 | Interrupt Hand |

 $\begin{array}{l} B_{3} = \langle \textit{CheckData}, \textit{FillJob} \rangle \\ B_{4} = \langle \textit{InterruptSoft}, \textit{exitS}, \textit{exitT}, \textit{CSProduc} \rangle \end{array}$



Definitions

example

Let $S = \{B_1, B_2, B_3, B_4\}$ with $B_1 = \langle GetFrame, exitGet \rangle$ $B_2 = \langle InterrupPeriod, exitI, InterruptSoft, exitS, exitT \rangle$

| | 1 | 9.5054 | GetFrame | |
|-----|----------------|--------|------------------|-------|
| | | 9.5073 | ExitGet | |
| | | 9.5081 | CS produc | |
| | | 9.5083 | Interrupt Period | |
| F | F ₁ | 9.5084 | Exitl | |
| | | 9.5096 | Interrupt Soft | |
| | | 9.5102 | ExitS | |
| | | 9.5127 | ExitIT | |
| | | 9.5154 | CheckData | |
| | + | 9.5260 | FillJob | |
| | - | 9.5715 | CS VGA | - |
| | | 9.5845 | CS produc | |
| - 6 | Ť | 9.5974 | GetFrame | |
| | | 9.6012 | ExitGet | |
| | | 9.6125 | Interrupt Hand | |
| | | 9.6155 | Exitl | |
| | | 9.6234 | ExitIT | |
| F | | 9.6315 | Interrupt Soft | |
| r 2 | | 9.6405 | Interrupt Period | 1 |
| | | 9.6483 | Exitl | |
| | | 9.6514 | Interrupt Soft | B_2 |
| | | 9.6622 | ExitS | |
| | | 0.6715 | ExitIT | |
| | | 9.6811 | CS produc | |
| | | 9.6898 | CheckData | |
| | * | 9.6932 | FillJob | |
| | | 9.6987 | CS VGA | |
| | | 9.7001 | Interrupt Hand | |
| | | | | |

 $\begin{array}{l} B_{3} = \langle \textit{CheckData}, \textit{FillJob} \rangle \\ B_{4} = \langle \textit{InterruptSoft}, \textit{exitS}, \textit{exitT}, \textit{CSProduc} \rangle \end{array}$



Definitions

example

Let $S = \{B_1, B_2, B_3, B_4\}$ with $B_1 = \langle GetFrame, exitGet \rangle$ $B_2 = \langle InterrupPeriod, exitI, InterruptSoft, exitS, exitT \rangle$

| | Ť | 9.5054 | GetFrame |
|---------|----|--------|----------------------|
| | | 9.5073 | ExitGet |
| | | 9.5081 | CS produc |
| | | 9.5083 | Interrupt Period |
| F_{1} | | 9.5084 | Exitl |
| | 1 | 9.5096 | Interrupt Soft |
| | | 9.5102 | ExitS |
| | | 9.5127 | ExitIT |
| | | 9.5154 | CheckData |
| | Ť. | 9.5260 | FillJob |
| | | 9.5715 | CS VGA |
| | | 9.5845 | CS produc |
| | Ť | 9.5974 | GetFrame] P |
| | | 9.6012 | ExitGet |
| | | 9.6125 | Interrupt Hand |
| | | 9.6155 | Exitl |
| | | 9.6234 | ExitIT |
| F | | 9.6315 | Interrupt Soft |
| 1 2 | | 9.6405 | Interrupt Period |
| | | 9.6483 | Exitl |
| | | 9.6514 | Interrupt Soft |
| | | 9.6622 | ExitS B ₄ |
| | | 9.6715 | ExitIT |
| | | 9.6811 | CS produc |
| | | 9.6898 | CheckData 1 B |
| - | ¥ | 9.6932 | FillJob J 23 |
| | | 9.6987 | CS VGA |
| | | 9.7001 | Interrupt Hand |

 $\begin{array}{l} B_{3} = \langle \textit{CheckData}, \textit{FillJob} \rangle \\ B_{4} = \langle \textit{InterruptSoft}, \textit{exitS}, \textit{exitT}, \textit{CSProduc} \rangle \end{array}$



Definitions

example

Let $S = \{B_1, B_2, B_3, B_4\}$ with $B_1 = \langle GetFrame, exitGet \rangle$ $B_2 = \langle InterrupPeriod, exitI, InterruptSoft, exitS, exitT \rangle$



 $\begin{array}{l} B_{3} = \langle \textit{CheckData}, \textit{FillJob} \rangle \\ B_{4} = \langle \textit{InterruptSoft}, \textit{exitS}, \textit{exitT}, \textit{CSProduc} \rangle \end{array}$

Definitions local coverage Coverage A coverage of F = {F₁, F₂} using S is C = {L₁, L₂} with L₁ = (B₁, B₃) a local coverage F₁ The covering degree is 0.5 coverage rank k-golden set

Definitions

example

Let $S = \{B_1, B_2, B_3, B_4\}$ with $B_1 = \langle GetFrame, exitGet \rangle$ $B_2 = \langle InterrupPeriod, exitI, InterruptSoft, exitS, exitT \rangle$



 $\begin{array}{l} B_{3} = \langle \textit{CheckData}, \textit{FillJob} \rangle \\ B_{4} = \langle \textit{InterruptSoft}, \textit{exitS}, \textit{exitT}, \textit{CSProduc} \rangle \end{array}$

Definitions

- local coverage
- 2 coverage
- coverage rank coverRank(S_1, \mathcal{F}) = Max(0.25, 0.5, 0.33) with $S_1 = \{B_1, B_3, B_4\}$

k-golden set

Definitions

example

Let $S = \{B_1, B_2, B_3, B_4\}$ with $B_1 = \langle GetFrame, exitGet \rangle$ $B_2 = \langle InterrupPeriod, exitI, InterruptSoft, exitS, exitT \rangle$

| _ | | | |
|---------|--------|-------------------|-----------------------|
| | 9.5054 | GetFrame | R |
| | 9.5073 | ExitGet | ^D 1 |
| | 9.5081 | CS produc | - I |
| | 9,5083 | Interrupt Period | 1 |
| F_{1} | 9,5084 | Exitl | |
| | 9,5096 | Interrupt Soft | B_2 |
| | 9.5102 | ExitS | |
| | 9.5127 | ExitIT | |
| | 9,5154 | CheckData | i. |
| | 9 5260 | Fill lob | <i>B</i> ₃ |
| - | 9.5/15 | CSVGA | - |
| | 9.5845 | CS produc | |
| - | 9 5974 | GetFrame | 1 |
| | 9,6012 | ExitGet | B_1 |
| | 9.6125 | Interrupt Hand | - 1 |
| | 9 6155 | Fxitl | |
| | 9 6234 | ExitIT | |
| 1005 | 9 6315 | Interrupt Soft | |
| F_2 | 9 6405 | Interrupt Period | 1 |
| | 9 6483 | Fxitl | |
| | 9 6514 | Interrunt Soft | B. |
| | 9 6622 | FxitS | - 2 |
| | 9 6715 | ExitIT | |
| | 9 6811 | CS produc | - I |
| | 9 6898 | CheckData | 1 |
| | 9 6932 | Fillloh | B_3 |
| - | 9 6987 | CS VGA | |
| | 9 7001 | Interrupt Hand | |
| | 0001 | interrupt Fidilia | |

 $\begin{array}{l} B_{3} = \langle \textit{CheckData}, \textit{FillJob} \rangle \\ B_{4} = \langle \textit{InterruptSoft}, \textit{exitS}, \textit{exitT}, \textit{CSProduc} \rangle \end{array}$

local coverage coverage coverage rank

• k-golden set $S_2 = \{B_1, B_2, B_3\}$ is the 3-goldenset with coverRank $(S_2, \mathcal{F}) = 0.64$

Problem statement submodular function maximization problem

Find a *k-golden* set of blocks that provides the best coverage

- ✓ Well known NP-Hard problem: $max{f(S) : |S| ≤ k}$ [Nemhauser et al.,1978][Kulik,2009]
- $\checkmark\,$ Studied in particular for resource allocation
- $\checkmark\,$ Proved that the lower bound of any local optimum f , found by a greedy algorithm $\geq\,\,63\%$

Problem statement submodular function maximization problem

Find a *k-golden* set of blocks that provides the best coverage

- ✓ Well known NP-Hard problem: $max{f(S) : |S| ≤ k}$ [Nemhauser et al.,1978][Kulik,2009]
- $\checkmark\,$ Studied in particular for resource allocation
- $\checkmark\,$ Proved that the lower bound of any local optimum f , found by a greedy algorithm $\geq\,\,63\%$
 - set of blocks obtained by applying profSpan [Zou et al.,2010]

Problem statement submodular function maximization problem

Find a *k-golden* set of blocks that provides the best coverage

- ✓ Well known NP-Hard problem: $max{f(S) : |S| ≤ k}$ [Nemhauser et al.,1978][Kulik,2009]
- $\checkmark\,$ Studied in particular for resource allocation
- $\checkmark\,$ Proved that the lower bound of any local optimum f , found by a greedy algorithm $\geq\,\,63\%$
 - set of blocks obtained by applying profSpan [Zou et al.,2010]
 - **f** is the function of Coverage Rank

FrameMiner Process





Experimentation settings

- Real trace of embedded MP4 video decoding
 - 240 frames and 123575 events
- ◊ Intel Xeon X4760 processor: 2.66 GHZ and 64 GB of RAM
- ◊ Exhaustive approach: baselineFrame
- ◊ Profspan: support threshold of 20%

Comparative assessment

 90% of the optimal solution with an execution three orders of magnitude faster.



Subjective assessment







Reduction

• 70% of reduction in the information to handle

Reduction

• 70% of reduction in the information to handle

→ The programmer can now focus on blocks, instead of individual events



(a)

(b)

Conclusion

• An original approach to abstract an execution trace

 \implies A great simplification of trace exploration

• An efficient greedy algorithm that automatically find interesting blocks

\implies A significant reduction of the execution trace volume

Further directions

- Automatic labelling
- Ø Multicore traces
- Beyond Coverage rank function

Questions

Thank you for your kind attention

