Mapping Software-Defined Radio Applications onto MPSoCs

Firew Siyoum, Sander Stuijk, Marc Geilen
Department of Electrical Engineering,
Eindhoven University of Technology

Map2MPSoC 2011

June 28, 2011

Background

Radio

- Modern embedded devices support multiple wireless communication standards.
- We refer to these standards as radios.
- Smartphones, for e.g., include various radios, such as WCDMA, LTE and IEEE 802.11x.

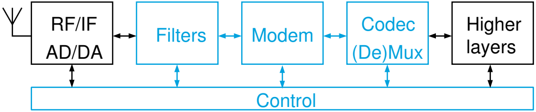
Background

Radio

- Modern embedded devices support multiple wireless communication standards.
- We refer to these standards as radios.
- Smartphones, for e.g., include various radios, such as WCDMA, LTE and IEEE 802.11x.

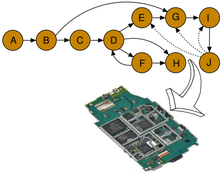
Radio Baseband Processing

- The three main stages are filters, modem and codec.
- These stages are customarily implemented as hardware blocks.



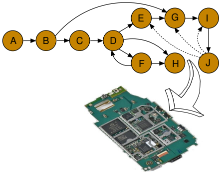
Software Defined Radio (SDR)

- Radio design is shifting from dedicated hardware blocks to software processes for better flexibility and cost efficiency.
- SDR is a radio implemented as software processes that run on a MPSoC.



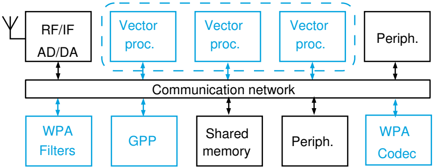
Software Defined Radio (SDR)

- Radio design is shifting from dedicated hardware blocks to software processes for better flexibility and cost efficiency.
- SDR is a radio implemented as software processes that run on a MPSoC.



MPSoC architectures for SDR

- combine homogeneous and heterogeneous multiprocessing, including GPPs, vector processors and weakly programmable accelerators.



Background

SDR design

- Requirements:
  - satisfying temporal requirements such as latency and throughput.
  - staying within limited resource budgets, such as battery power.

Background

SDR design

- Requirements:
  - satisfying temporal requirements such as latency and throughput.
  - staying within limited resource budgets, such as battery power.

- Challenges include
  - dynamism: data-dependent workload and system variability.
  - high workload: e.g. in smartphones, a digital workload of 100GOPS within 1W power budget.

Background

SDR design

- Requirements:
  - satisfying temporal requirements such as latency and throughput.
  - staying within limited resource budgets, such as battery power.
- Challenges include
  - dynamism: data-dependent workload and system variability.
  - high workload: e.g. in smartphones, a digital workload of 100GOPS within 1W power budget.
- One of our solutions is variation-aware dataflow-based design flow.

Technische Universiteit Eindhoven University of Technology

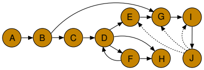
Background

Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.

Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.
    - modeling techniques for radios, storage, arbitration, etc

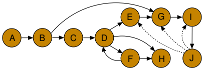


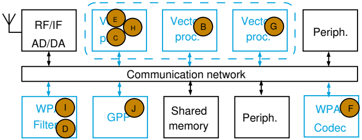
Modeling

Background

Variation-aware dataflow-based design flow

- The design flow comprises three main aspects.
  - modeling techniques for radios, storage, arbitration, etc
  - scheduling techniques for scarce resources such as power and memory



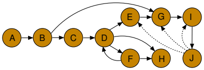


Modeling

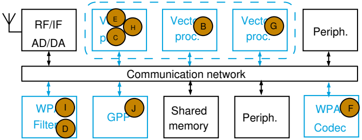
Mapping/scheduling

Variation-aware dataflow-based design flow

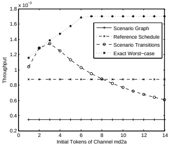
- The design flow comprises three main aspects.
  - modeling techniques for radios, storage, arbitration, etc
  - scheduling techniques for scarce resources such as power and memory
  - analysis techniques to compute buffer-sizing, latency and throughput.



Modeling



Mapping/scheduling



Analysis

Outline

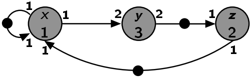
What Synchronous Dataflow Scenarios are
and how to analyse them

How to model radios using scenarios
dynamism in Long Term Evolution (LTE)
and how scenarios capture dynamism in LTE

Synchronous dataflow (SDF)

SDF

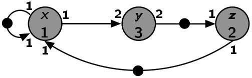
- A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.



Synchronous dataflow (SDF)

SDF

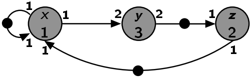
- A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.



- Firing of an actor = starting execution

SDF

- A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.

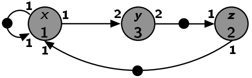


- Firing of an actor = starting execution
- Repetition vector: the number of firings of each actor that brings the graph back to its original state. e.g for the above SDF, $\bar{R} = [2, 1, 2]$

Synchronous dataflow (SDF)

SDF

- A SDF is a directed graph that can model concurrent tasks.
- consists of actors that communicate tokens through FIFO channels.
- Actors have fixed port rates and execution durations.
- Channels may have fixed number of initial tokens.



- Firing of an actor = starting execution
- Repetition vector: the number of firings of each actor that brings the graph back to its original state. e.g for the above SDF, $\bar{R} = [2, 1, 2]$

Iteration

- is a set of actor firings, as specified by the repetition vector.
- an iteration is marked by the production times of initial tokens, that is recoreded in a time-stamp vector γ.

Synchronous Dataflow Scenarios

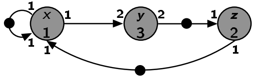
Scenario-aware Dataflow (SADF)

- A scenario - a single mode of operation of an application.

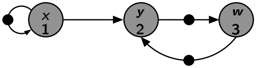
Synchronous Dataflow Scenarios

Scenario-aware Dataflow (SADF)

- A scenario - a single mode of operation of an application.
- For a scenario, behavior is mostly invariable and modeled with SDF.



scenario 1

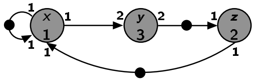


scenario 2

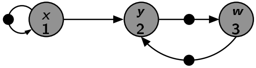
Synchronous Dataflow Scenarios

Scenario-aware Dataflow (SADF)

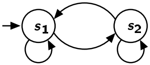
- A scenario - a single mode of operation of an application.
- For a scenario, behavior is mostly invariable and modeled with SDF.
- Possible transitions can be modeled by a finite-state machine (FSM).



scenario 1



scenario 2

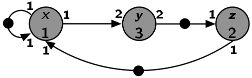


FSM

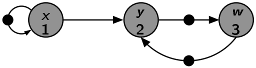
Synchronous Dataflow Scenarios

Scenario-aware Dataflow (SADF)

- A scenario - a single mode of operation of an application.
- For a scenario, behavior is mostly invariable and modeled with SDF.
- Possible transitions can be modeled by a finite-state machine (FSM).



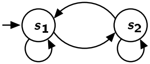
scenario 1



scenario 2

FSM-SADF

- is a tuple $\mathbf{F} = (S, \mathbf{f})$, consisting of a set of scenarios S and a FSM $\mathbf{f}$.

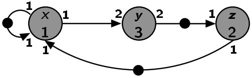


FSM

Analysing FSM-SADF - single scenario

- Every SDF has a matrix **M**, whose dimension is $|\gamma| \times |\gamma|$.

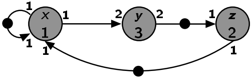
- Every SDF has a matrix **M**, whose dimension is $|\gamma| \times |\gamma|$.



$\mathbf{M} = \begin{bmatrix} 2 & 3 & 2 \\ 5 & 6 & 5 \\ 7 & 8 & 7 \end{bmatrix}$, i_{xy}=distance b/n tokens y & x tokens

Analysing FSM-SADF - single scenario

- Every SDF has a matrix $\mathbf{M}$, whose dimension is $|\gamma| \times |\gamma|$.

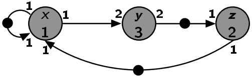


$\mathbf{M} = \begin{bmatrix} 2 & 3 & 2 \\ 5 & 6 & 5 \\ 7 & 8 & 7 \end{bmatrix}$, i_{xy}=distance b/n tokens y & x tokens

- Any two consecutive iterations are related by a matrix multiplication in $(max, +)$ algebra, i.e. $\gamma_{k+1} = \mathbf{M} \cdot \gamma_k$.

Analysing FSM-SADF - single scenario

- Every SDF has a matrix $\mathbf{M}$, whose dimension is $|\gamma| \times |\gamma|$.



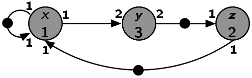
$$\mathbf{M} = \begin{bmatrix} 2 & 3 & 2 \\ 5 & 6 & 5 \\ 7 & 8 & 7 \end{bmatrix}, \; i_{xy}\text{=distance b/n tokens } y \text{ \& } x \text{ tokens}$$

- Any two consecutive iterations are related by a matrix multiplication in $(max, +)$ algebra, i.e. $\gamma_{k+1} = \mathbf{M} \cdot \gamma_k$.
- for the above graph, γ for the first 3 iterations are

$$\gamma_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \gamma_1 = \begin{bmatrix} 3 \\ 6 \\ 8 \end{bmatrix}, \gamma_2 = \begin{bmatrix} 10 \\ 13 \\ 15 \end{bmatrix}, \gamma_3 = \begin{bmatrix} 17 \\ 20 \\ 22 \end{bmatrix}$$

Analysing FSM-SADF - single scenario

- Every SDF has a matrix $\mathbf{M}$, whose dimension is $|\gamma| \times |\gamma|$.



$\mathbf{M} = \begin{bmatrix} 2 & 3 & 2 \\ 5 & 6 & 5 \\ 7 & 8 & 7 \end{bmatrix}$, i_{xy}=distance b/n tokens y & x tokens

- Any two consecutive iterations are related by a matrix multiplication in $(max, +)$ algebra, i.e. $\gamma_{k+1} = \mathbf{M} \cdot \gamma_k$.
- for the above graph, γ for the first 3 iterations are

$$\gamma_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \gamma_1 = \begin{bmatrix} 3 \\ 6 \\ 8 \end{bmatrix}, \gamma_2 = \begin{bmatrix} 10 \\ 13 \\ 15 \end{bmatrix}, \gamma_3 = \begin{bmatrix} 17 \\ 20 \\ 22 \end{bmatrix}$$

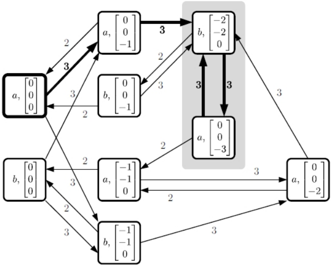
- Period = 7 time-units & hence, throughput = $\frac{1}{7}$ iters. per time-unit.

Technische Universiteit Eindhoven University of Technology

Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.

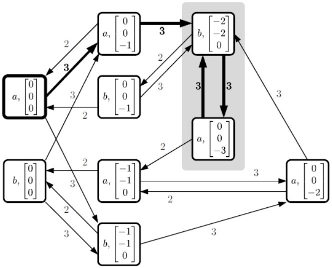


* M.Geilen and S. Stuijk, CODES+ISSS, 2010

Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.



Conservative methods†

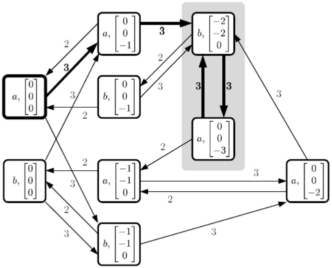
- linear upper-bounds to γ_k

* M.Geilen and S. Stuijk, CODES+ISSS, 2010
†M.Geilen, ACM tran. on Emb. Comp. Sys. 2010

Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.



Conservative methods†

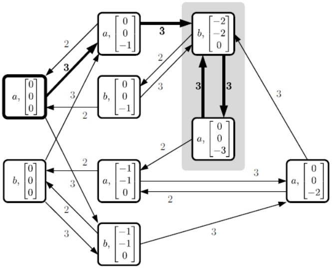
- linear upper-bounds to γ_k
- find a vector $\bar{\tau}$ and a scalar number λ such that $\gamma_k \preceq \bar{\tau} + k \cdot \lambda$

* M.Geilen and S. Stuijk, CODES+ISSS, 2010
†M.Geilen, ACM tran. on Emb. Comp. Sys. 2010

Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.



Conservative methods†

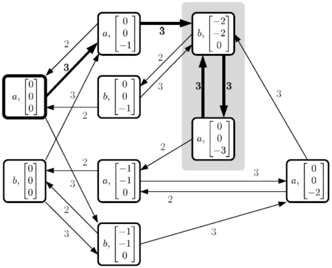
- linear upper-bounds to γ_k
- find a vector $\bar{\tau}$ and a scalar number λ such that $\gamma_k \preceq \bar{\tau} + k \cdot \lambda$
- **Scenario graph**: $\max_{s \in S} \tau_s + \lambda_s$

* M.Geilen and S. Stuijk, CODES+ISSS, 2010
†M.Geilen, ACM tran. on Emb. Comp. Sys. 2010

Analysing FSM-SADF: multiple scenarios

Exact methods*

- the inverse of maximum cycle mean (MCM) of the state space.



Conservative methods†

- linear upper-bounds to γ_k
- find a vector $\bar{\tau}$ and a scalar number λ such that $\gamma_k \preceq \bar{\tau} + k \cdot \lambda$
- **Scenario graph**: $\max_{s \in S} \tau_s + \lambda_s$
- **Reference schedule**: MCM of the FSM where the weight of each node is $\tau_s + \lambda_s$.

* M.Geilen and S. Stuijk, CODES+ISSS, 2010
†M.Geilen, ACM tran. on Emb. Comp. Sys. 2010

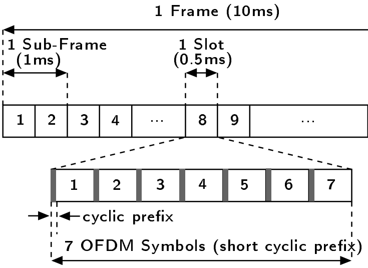
Long Term Evolution (LTE)

- LTE is a recent standard in cellular wireless communication.

Long Term Evolution (LTE)

- LTE is a recent standard in cellular wireless communication.

LTE frame structure for FDD

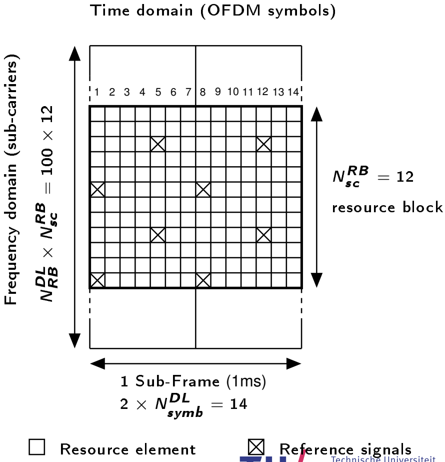


- allocation of the grid to channels varies between sub-frames.

Long Term Evolution (LTE)

- LTE is a recent standard in cellular wireless communication.

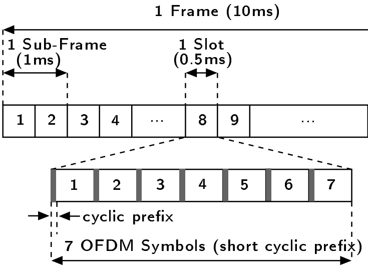
Resource grid of a sub-frame



Long Term Evolution (LTE)

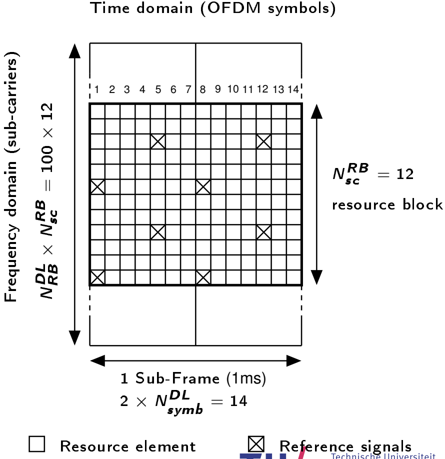
- LTE is a recent standard in cellular wireless communication.

LTE frame structure for FDD

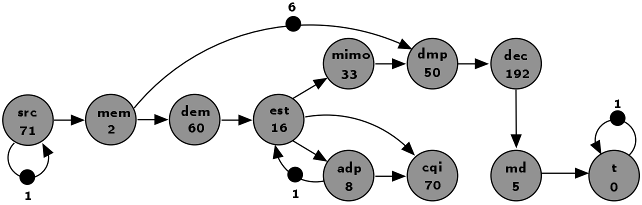


- allocation of the grid to channels varies between sub-frames.

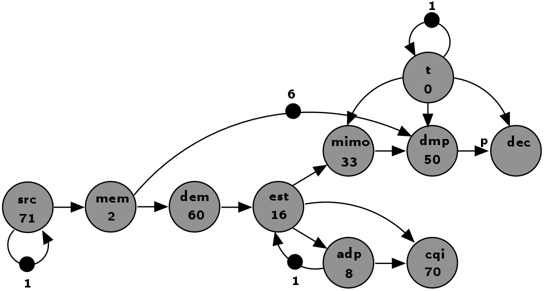
Resource grid of a sub-frame



FSM-SADF model of LTE



S_1



$S_2 : \mathcal{X}(\boldsymbol{dec}) = \mathbf{192}, \mathcal{R}(\boldsymbol{p}) = \mathbf{1}$ $S_4 : \mathcal{X}(\boldsymbol{dec}) = \mathbf{895}, \mathcal{R}(\boldsymbol{p}) = \mathbf{12}$

$S_3 : \mathcal{X}(\boldsymbol{dec}) = \mathbf{970}, \mathcal{R}(\boldsymbol{p}) = \mathbf{13}$ $S_5 : \mathcal{X}(\boldsymbol{dec}) = \mathbf{820}, \mathcal{R}(\boldsymbol{p}) = \mathbf{11}$

Analysing FSM-SADF model of LTE

Worst-case throughput computation

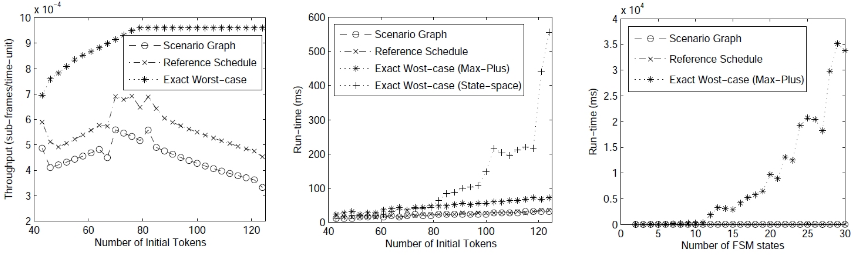
- WCT computation of the FSM-SADF model ($\times 10^{-4}$ sub-frames/time-unit)

Method	1	2	3	4	5
Name	Static SDFG	Scenario graph	Reference schedule	State-space	MaxPlus
WCT	2.6	5.2	6.6	8.9	8.9

- Scenario-based techniques improve the static SDF result by 2 to 3.4 times more.

Trade-offs between methods

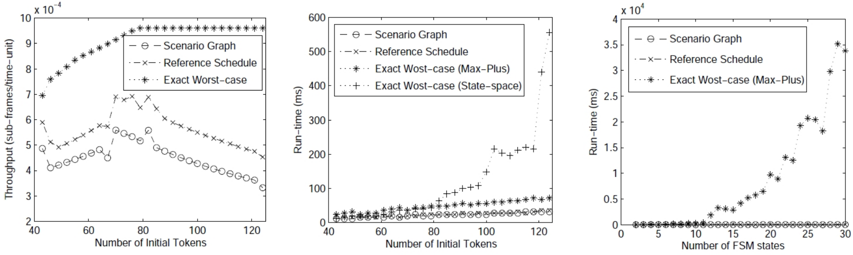
- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.



(a) Varying number of initial tokens (b) Varying number of initial tokens (c) Varying number of FSM states

Trade-offs between methods

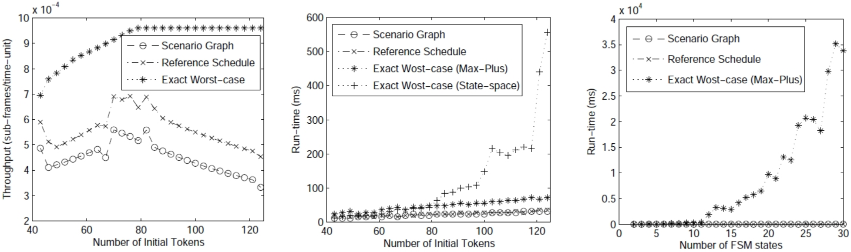
- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
- Methods 2 and 3 trade accuracy for lower run-time - useful for iterative DSE algorithms.



(a) Varying number of initial tokens (b) Varying number of initial tokens (c) Varying number of FSM states

Trade-offs between methods

- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
- Methods 2 and 3 trade accuracy for lower run-time - useful for iterative DSE algorithms.
- Methods 4 and 5 give the exact WCT, at the cost of run-time.



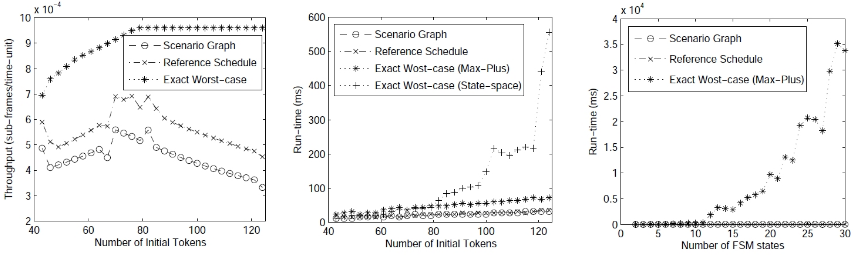
(a) Varying number of initial tokens

(b) Varying number of initial tokens

(c) Varying number of FSM states

Trade-offs between methods

- There are trade-offs in conservativeness, run-time and scalability between the different analysis techniques.
- Methods 2 and 3 trade accuracy for lower run-time - useful for iterative DSE algorithms.
- Methods 4 and 5 give the exact WCT, at the cost of run-time.
- Method 5 has a run-time in the order of tens of second.



(a) Varying number of initial tokens (b) Varying number of initial tokens (c) Varying number of FSM states

Conclusions

Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to pessimistic temporal analysis.

Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to pessimistic temporal analysis.
- Synchronous dataflow scenarios can be used to capture dynamism in SDRs.

Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to pessimistic temporal analysis.
- Synchronous dataflow scenarios can be used to capture dynamism in SDRs.
- Existing timing analysis techniques of SDF scenarios have very low run-time that scales well with increase in graph size.

Conclusions

- Static dataflow models such as SDF that abstract applications dynamism lead to pessimistic temporal analysis.
- Synchronous dataflow scenarios can be used to capture dynamism in SDRs.
- Existing timing analysis techniques of SDF scenarios have very low run-time that scales well with increase in graph size.

Thank you! Questions?