An Exact MCMC Accelerator Under Custom Precision

Shuanglong Liu

Imperial College London

FPT 2015, Queenstown

09 Dec 2015

Markov chain Monte Carlo (MCMC)

- MCMC is a general purpose technique for **sampling** from complex probabilistic models;

- In high dimensional space, **sampling** is a key step for
  (a) **modelling** (simulation, synthesis, verification)
  (b) **learning** (estimating parameters)
  (c) **estimation** (Monte Carlo integration, importance sampling)

- MCMC has been considered to be one of the top ten most important algorithms ever.

Example: Monte Carlo Integration

- In scientific computing, one often needs to compute the integral in very high dimensional space.

$$I(f) = \int f(x)p(x)dx$$

- Many functions, equations, and distributions cannot be integrated analytically. For example:

$$p(x) = e^{x^2}$$

- If We can draw samples from p(x)

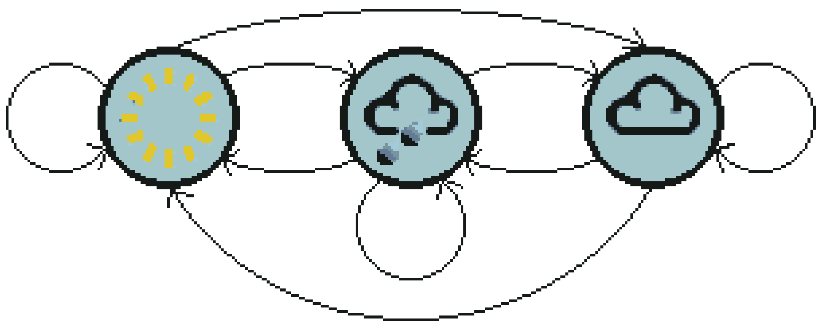
$$x_1, x_2, x_3, \ldots, x_N \sim p(x)$$

- We can easily estimate the integral from

$$I(f) = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

Markov chain Monte Carlo (MCMC)

- MCMC outputs a sequence of samples that are slightly dependent from distribution, by constructing a discrete time Markov chain;



MCMC Algorithm

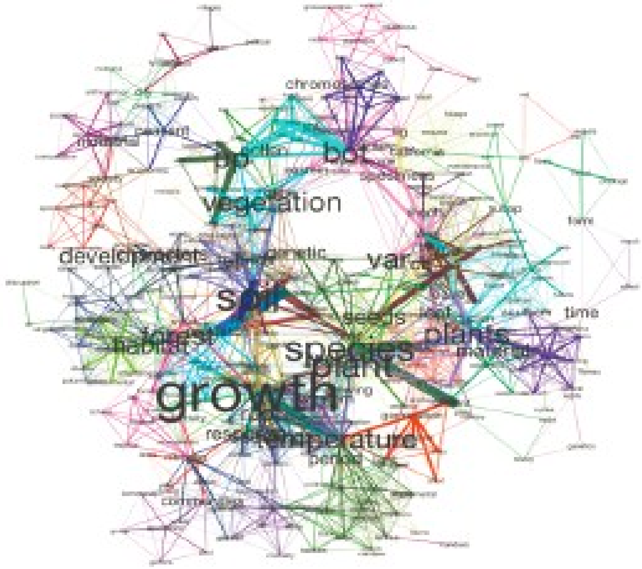
Input: initial setting θ_0, number of samples N_s;
Output: parameter samples $\theta_i, i = 1, ..., N_s$;

for $i = 1$ **to** N_s **do**
    Propose $\theta' \sim \theta_{i-1}$+Normal(0, $s^2 I_D$); // a random walk proposal with step size s.
    Compute $a = \boxed{\dfrac{p(\theta' \mid \{x_n\}_{n=1}^N)}{p(\theta_{i-1} \mid \{x_n\}_{n=1}^N)};}$
    $u \sim$ Uniform(0,1);
    if $u \leq a$ **then**
        $\theta_i = \theta'$;
    else
        $\theta_i = \theta_{i-1}$;
    end if
end for

Motivation

- **Big Data**
- **Complex Models**

MCMC needs **20 days** to sample

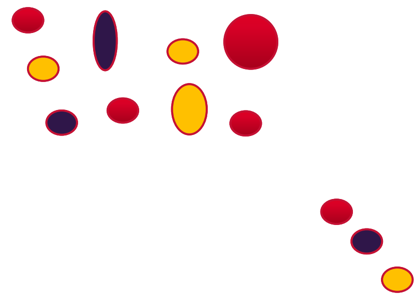


Complex / “intractable” likelihoods in high dimensionalities

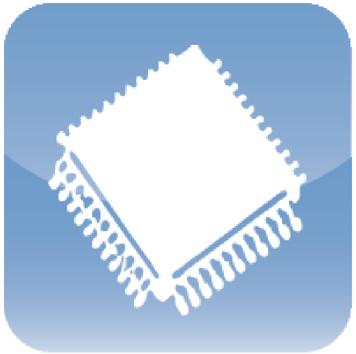
Imperial College London

Motivation

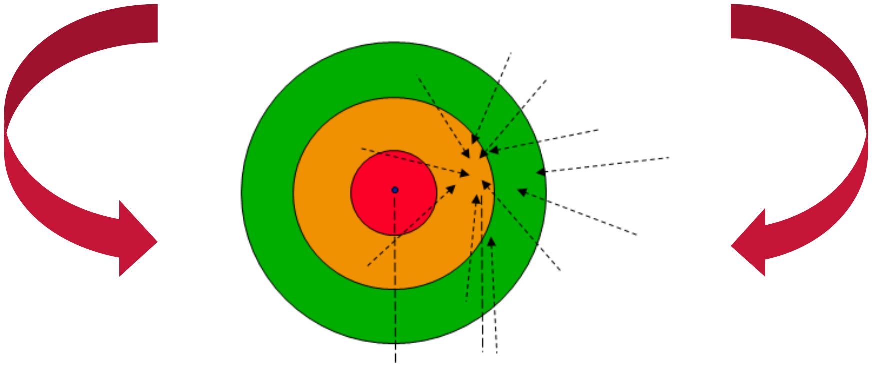
- Previous solutions to big data MCMC applications:



Data Sub-sampling

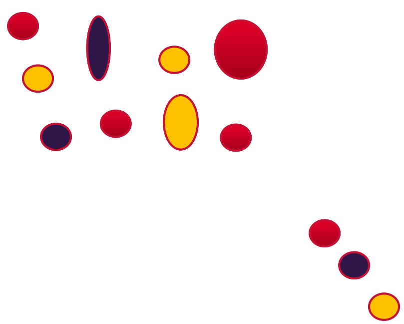


Custom Precision

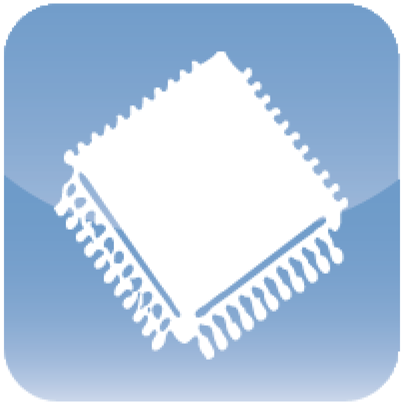


Biased Sampler

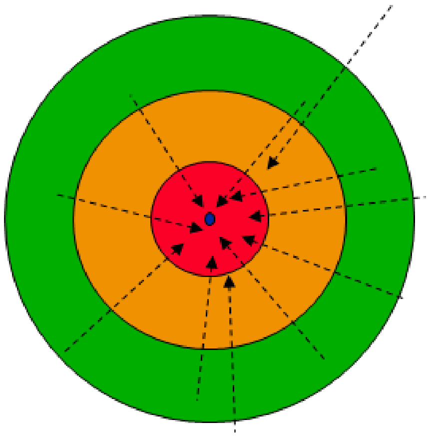
An Exact MCMC: FIREFLY MC



Data Sub-sampling



Custom Precision



Unbiased Sampler

Contribution

- A mixed precision MCMC accelerator with unbiased samples, taking into account the unique custom precision capabilities of FPGAs;
- A novel architecture which maps the algorithm to an FPGA;
- Evaluation using two case studies with varying complexity, achieving 4.21x and 4.76x speedups over double-precision designs;

Introduction: HOW IT WORKS

Assuming we have:

1. **Target Distribution:**

$$p(\theta \mid \{x_n\}_{n=1}^{N}) \propto p(\theta)\prod_{n=1}^{N} p(x_n \mid \theta)$$

2. **Likelihood function:**

$$L_n(\theta) = p(x_n \mid \theta) \qquad L(\theta) = \prod_{n=1}^{N} L_n(\theta)$$

Compute all N likelihoods at every iteration is a bottleneck!

3. **Assume each term can be bounded by a lower bound:**

$$0 \le B_n(\theta) \le L_n(\theta)$$

4. **For each one, we introduce an auxiliary binary variable zn ∈ {0,1}:**

$$z_n \sim Bernoulli\{1 - B_n(\theta) / L_n(\theta)\}$$

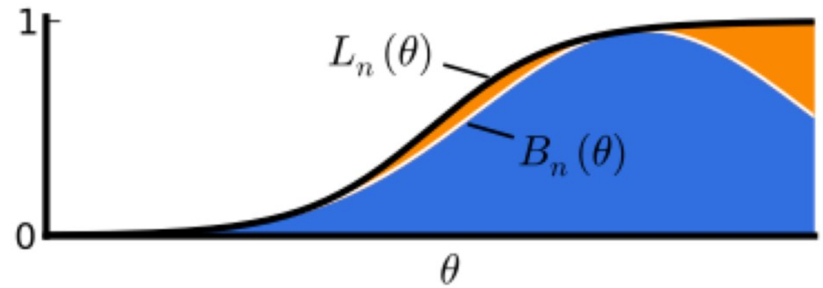
5. **Augment the posterior with these N vars:**

$$p(\theta, \{z_n\}_{n=1}^{N} \mid \{x_n\}_{n=1}^{N}) \propto p(\theta)\prod_{n=1}^{N} p(x_n \mid \theta) p(z_n \mid x_n, \theta)$$

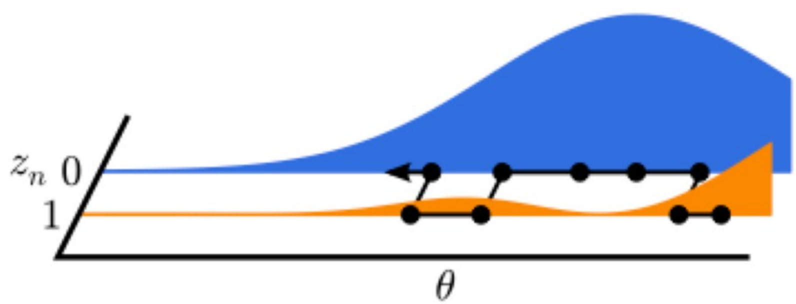
Introduction: HOW IT WORKS

We simulate the Markov chain on the z_n space:

$$L(\theta) = \prod_{i}^{z_i=1} L_i(\theta) - B_i(\theta) \prod_{j}^{z_j=0} B_j(\theta)$$



- z_n=0 => no likelihood computed
- z_n=1 => likelihoods computed



Imperial College London

Introduction: HOW IT COMBINED WITH CUSTOM PRECISION

- we propose to implement these likelihood terms under custom precision approximations as their lower bound functions, in order to get a tight bound;

- To guarantee a lower bound, we use the tool Gappa++ to get the errors between two precision values, then subtracting the error from custom precision value:

$LD_n(\theta) \sim p(x_n \mid \theta)$: *double precision likelihood*

$LC_n(\theta) \sim p(x_n \mid \theta)$: *custom precision likelihood*

ε : *max absolute difference of the two precision values*

$$B_n(\theta) = LC_n(\theta) - \varepsilon$$

Imperial College London

Firefly Algorithm

1. Choose a starting value $\theta(0)$;
2. At iteration *t*, propose a candidate $\theta*$ from a jumping distribution;
3. For each data point *n*:

    if z_n=1 then

        likelihood computation: $L(\theta*)$ *= $LDn(\theta*)$ - $LCn(\theta*)$;

        z_n update: z_n~Bernoulli(1-LCn/LDn);

    if z_n=0 then

        likelihood computation: $L(\theta*)$ *= $LCn(\theta*)$;

        partial z_n update:

            if (n%Fraction == 0) z_n~Bernoulli(1-LCn/LDn);

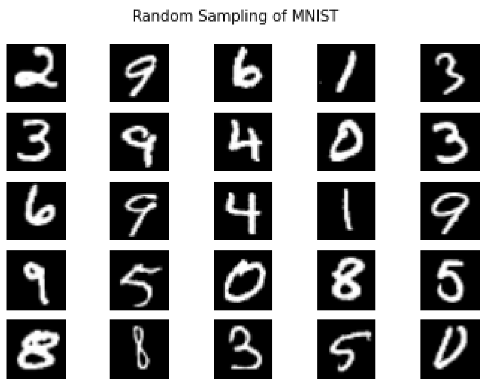
            else z_n=0; //keep unchanged

4. Compute accept ratio a= $L(\theta*)/L(\theta(t-1))$;
5. Accept $\theta*$ as $\theta(t)$ with probability min(a,1);
6. Repeat steps 2-5 M times to get M draws.

Imperial College London

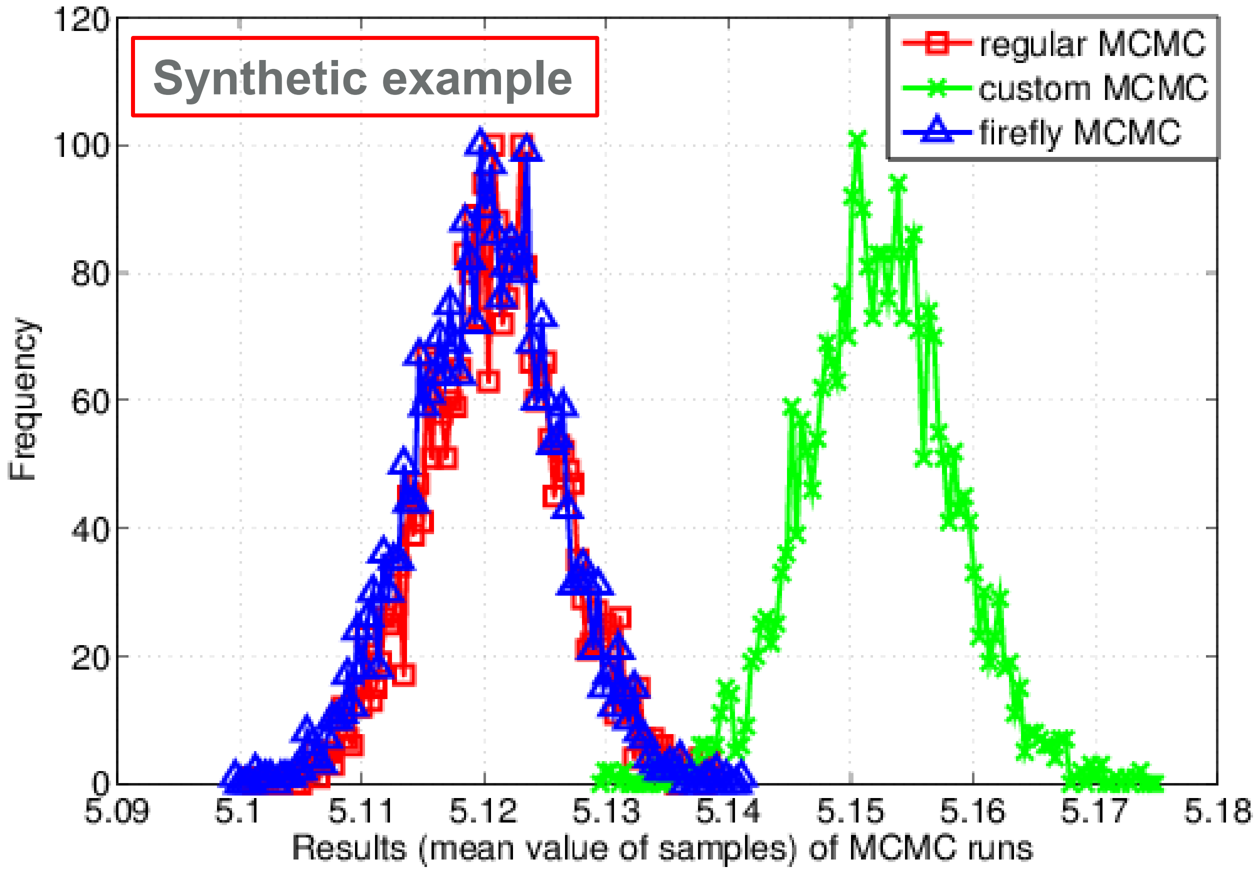
Case Studies

- Example: Logistic Regression
  - a two-class classification problem;
- Synthetic data set
  - 3-dimension of the parameters;
  - 3*500-dimension of the data set;
- MNIST Classification
  - to classify handwritten digits in the large MNIST database;
  - 13-dimension of the parameters
  - 2000-dimension of the data points



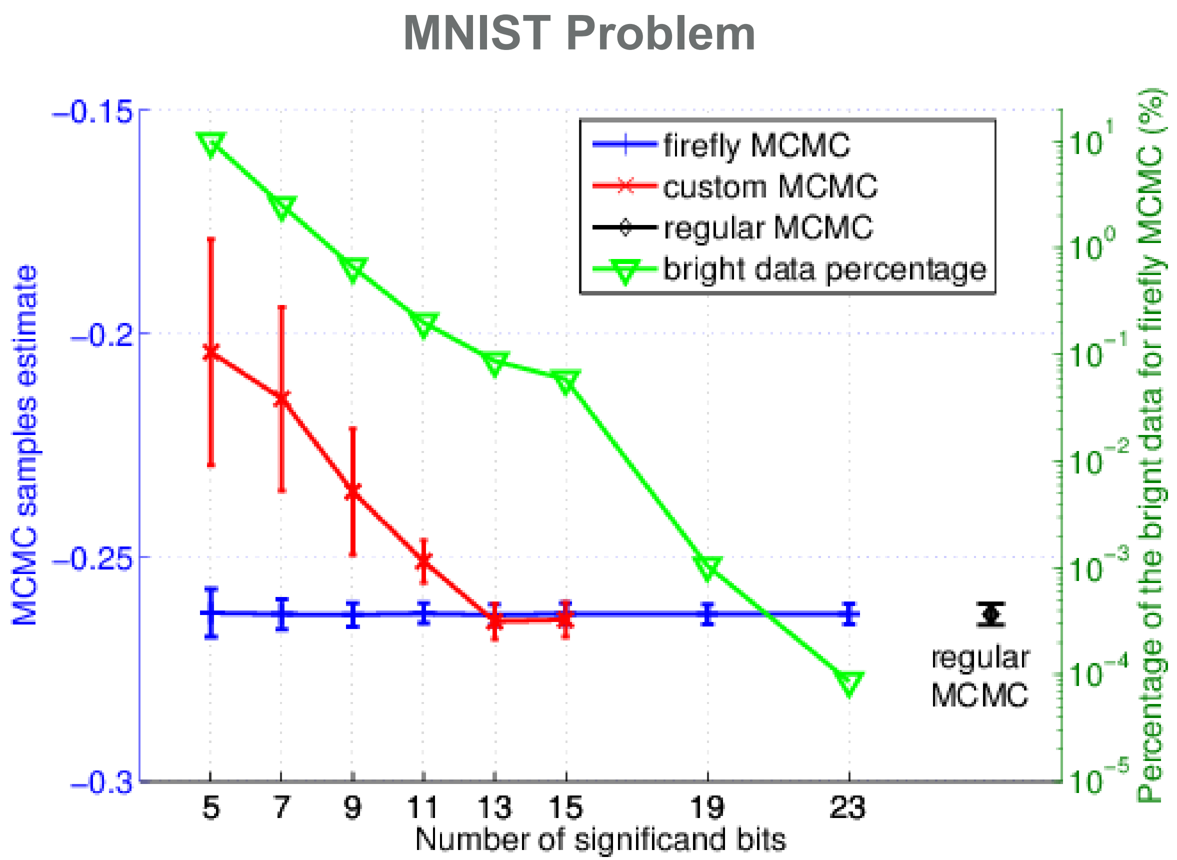
Imperial College London

Results: MCMC Samples

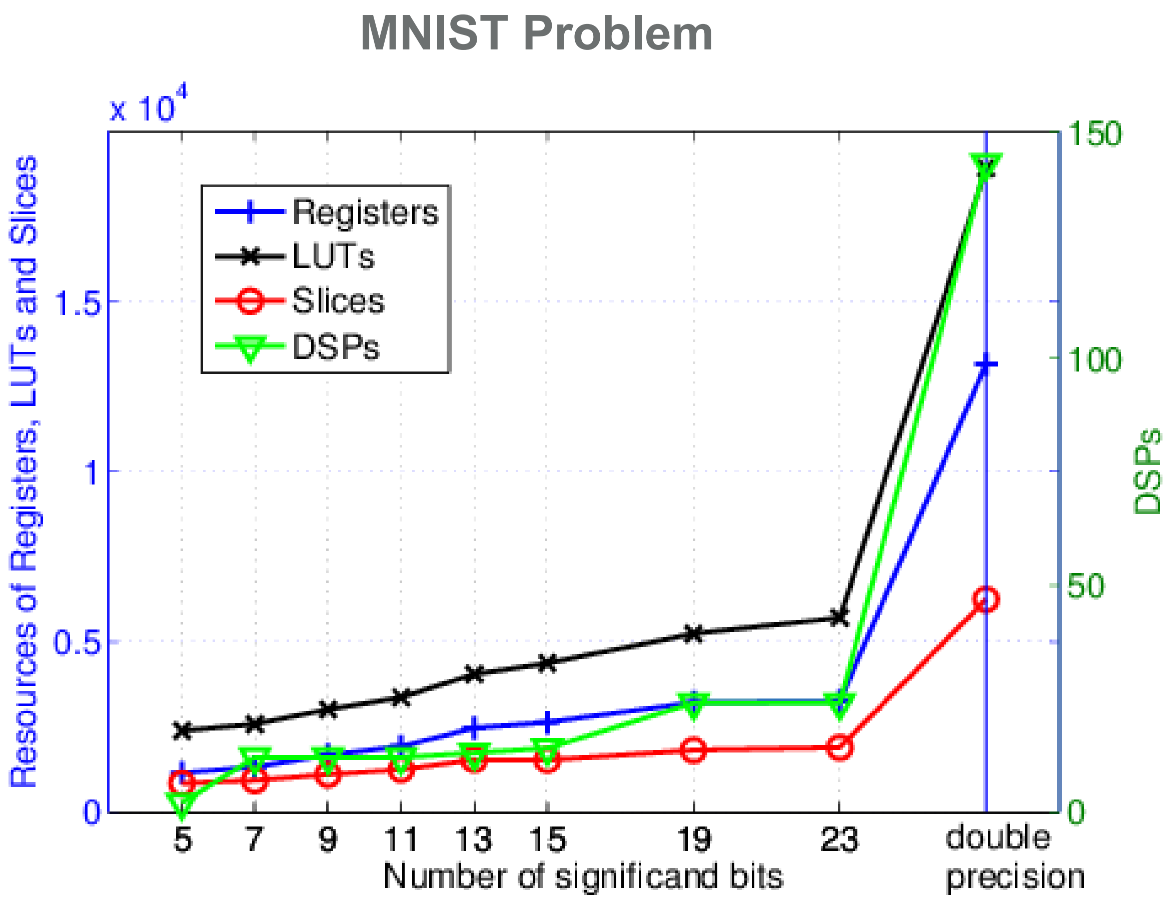


Imperial College London

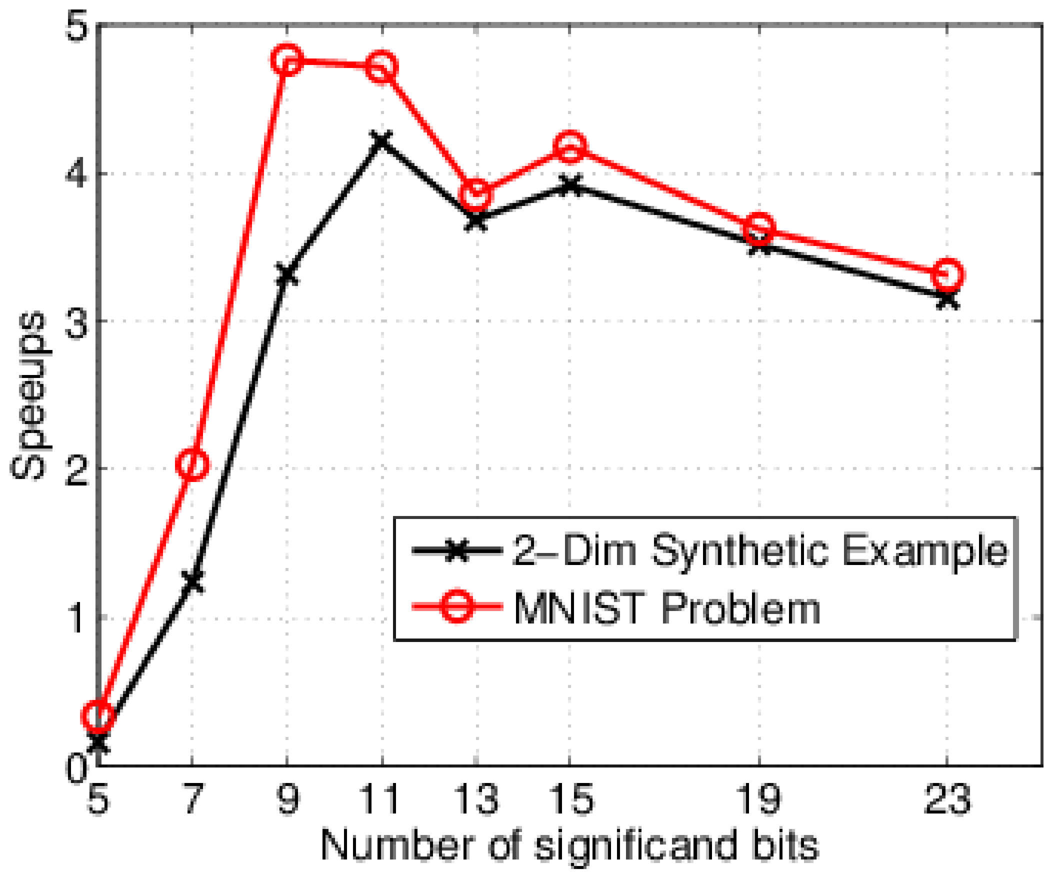
Results: MCMC Samples



Results: Resources



Results: Sampling Efficiency Speedups



Conclusions

- Firefly MC Algorithm
  - ✓ mixed precision design;
  - ✓ unbiased samples guaranteed;

- 4x-5x speedups over regular MCMC design;

- Custom precision values used as lower bound
  - ✓ not application specific;
  - ✓ a very tight bound;

Thanks

QUESTIONS?