Computing the expected longest path of task graphs in the presence of silent errors

Henri Casanova, **Julien Herrmann**, Yves Robert

Nashville - May 19, 2015

Outline

Outline

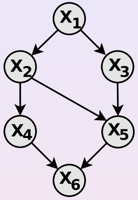
Motivation

Why do we need to compute longest paths in graphs?

- HPC applications seen as a computational workflow:
  $\hookrightarrow$ Vertices: tasks with execution time $x_i \in \mathbb{R}^+$
  $\hookrightarrow$ Edges: data dependencies

- List scheduling:
  $\hookrightarrow$ Critical Path Scheduling (based on bottom-level)
  $\hookrightarrow$ HEFT algorithm (for heterogeneous environments)

Motivation



Longest path:

$$L = X_6 + max(X_5 + max(X_2, X_3), X_4 + X_2) + X_1$$

- Deterministic weights:
  $\hookrightarrow$ Deap-first search: $O(|V| + |E|)$
- Random weights (PERT networks):
  $\hookrightarrow$ L is a random variable
  $\hookrightarrow$ Computing L's distribution: #P-complete
  $\hookrightarrow$ Computing L's expected value: #P-complete

Outline

Monte-Carlo approach

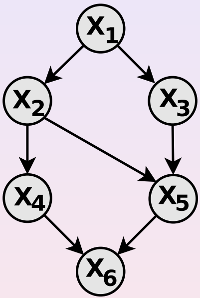
- For each task: weight is **sampled** from its probability distribution:

$$x_i \leftarrow X_i$$

- Longest path is computed:

$$L(x_1, x_2, ..., x_n)$$

- Repeat a large number of iteration



$\hookrightarrow$ gives an **empirical expected value**

Approximation by a series-parallel graph

We deal with **independent** variables!

- $X_1 + X_2$: **convolution of density functions**

$$f_{X_1+X_2}(x) = \int_t f_{X_1}(t) f_{X_2}(x-t)dt$$

- $max(X_1, X_2)$: **product of cumulative functions**

$$F_{X_1 \times X_2} = F_{X_1} \times F_{X_2}$$

$$f_{X_1 \times X_2}(x) = F_{X_1}(x) \times f_{X_2}(x) + f_{X_1}(x) \times F_{X_2}(x)$$

Exact results on series-parallel graphs.

Dodin algorithm on general graphs [Op. Research, 1985]

Approximation by a series-parallel graph.

Approximation with normality assumption

Clark's formula on two **dependent** normal laws:

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2) \qquad X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

Approximation of the Sum and Max by a normal law.

- Sum of corollated normal laws: $X_3 = X_1 + X_2$
  - Expected value: $\mu_3 = \mu_1 + \mu_2$
  - Variance: $\sigma_3^2 = \sigma_1^2 + 2.\sigma_1^2.\sigma_2^2.\rho_{X_1,X_2} + \sigma_2^2$
  - Correlation coefficient: closed formula
- Max of corollated normal laws: $X_3 = X_1 \times X_2$
  - (Complicated) closed formulas

Normal approximation on general graphs [Op. Research, 1983]

Consider that every random variable is normally distributed.

Outline

Silent errors

- Silent Data Corruptions
  $\hookrightarrow$ major challenges for Exascale
  $\hookrightarrow$ cosmic radiations
  $\hookrightarrow$ packaging pollution
  $\hookrightarrow$ Dynamic Voltage Frequency Scaling

- Verification at the end of task
  $\hookrightarrow$ checksums (linear algebra kernels)

- Errors are independent and exponentially distributed
  $\hookrightarrow$ Mean Time Between Failure: $1/\lambda$

First order approximation

- Probability that an error occurs during the first execution of task i:

$$1 - e^{-\lambda a_i} = \lambda a_i + O(\lambda^2)$$

- Probability that an error occurs during the first **and** the second execution of task i:

$$(1 - e^{-\lambda a_i})^2 = O(\lambda^2)$$

- λ is small $\Rightarrow$ **first order approximation**

Model with first order approximation

For every task i :

$$X_i = \begin{cases} a_i & \text{with probability} \quad 1 - \lambda a_i \\ 2.a_i & \text{with probability} \quad \lambda a_i \end{cases}$$

Model with first order approximation

For every task i :

$$X_i = \begin{cases} a_i & \text{with probability} \quad 1 - \lambda a_i \\ 2.a_i & \text{with probability} \quad \lambda a_i \end{cases}$$

Theorem

Computing the expected longest path of a **probabilistic 2-state DAG** is a #P-complete problem.

$\mathcal{E}(G)$: expected longest path of G

- $\mathcal{E}(G)$ is a **polynomial** in λ
- $\mathcal{E}(G) = L(G) + \lambda \sum_{i \in V} a_i(L(G_i) - L(G)) + O(\lambda^2)$

  where $L(G)$: deterministic longest path in G

  $L(G_i)$: deterministic longest path in G when task i has weight $2.a_i$
- First order approximation:

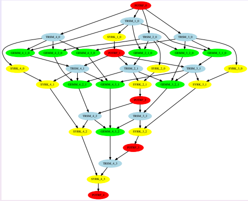
$$\mathcal{E}(G) = L(G) + \lambda \sum_{i \in V} a_i(L(G_i) - L(G))$$

$\hookrightarrow (n+1)$ Deap-first search: $O(|V|^2 + |V|.|E|)$

Outline

Experiments

- Evaluation of:
  ↪ First order approximation
  ↪ Approximation by a series-parallel graph
  ↪ Approximation with normality assumption

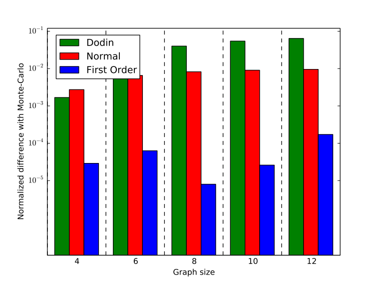


- Comparison with Monte-Carlo approach
  ↪ 300,000 iterations

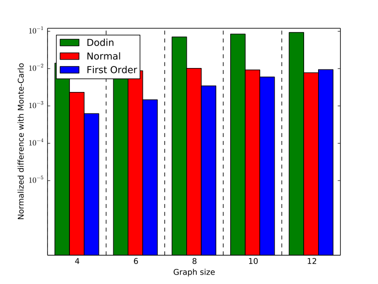
- DAGs from tiled Cholesky, LU and QR factorizations
  ↪ kernels execution times from StarPU

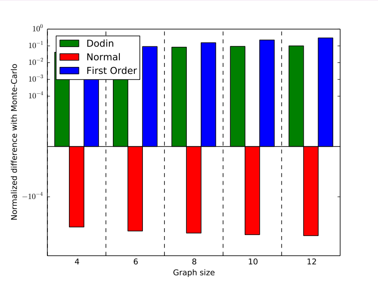
Results for QR factorization

$\lambda = 0.001$



$\lambda = 0.01$

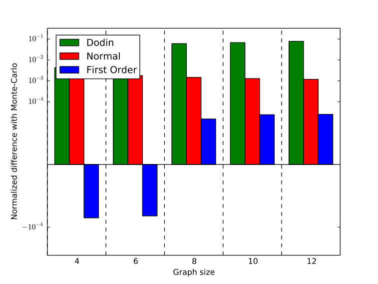




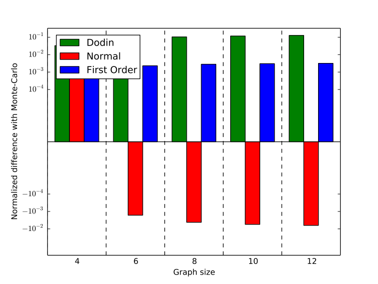
$\lambda = 0.1$

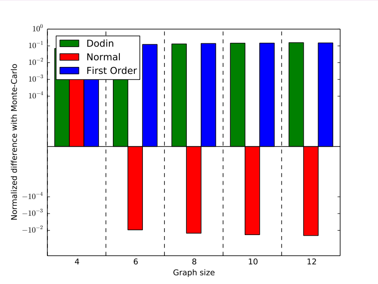
Results for LU factorization

$\lambda = 0.001$



$\lambda = 0.01$





$\lambda = 0.1$

Outline

Conclusion

First order approximation

- First order approximation of expected longest path with silent errors
- Lower complexity than existing methods
- Better results for small (but realistic) failure rate

Perspective

- Expected makespan for limited resources
- Introduction of checkpoints