Probabilistic models for analysis and optimization of grid experiments

Tristan Glatard[1]
Johan Montagnat[2]
Sorina Camarasu-Pop[1]

[1]University of Lyon, CREATIS-LRMN
[2]University of Nice Sophia-Antipolis, CNRS, I3S

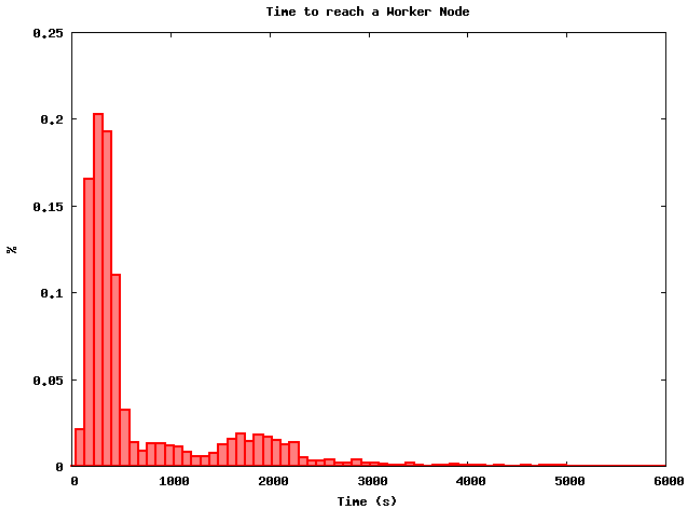
4th EGEE User-Forum, "From grid monitoring to analysis" session

Analysis/optimization issues on EGEE

- From an application point of view
  - Setting of submission parameters (timeout, job requirements, granularity, etc)
  - Performance interpretation
  - Error diagnosis, characterization and handling
  - Quality of service
- From a computer-science point of view
  - Comparative studies
  - Global consequences of a local behavior

Assumptions

- Grid seen as a black-box introducing a random latency on jobs
- Latency is a random variable capturing
  - Submission, scheduling, queuing delays
  - Load variations
  - Resource heterogeneity
- Latency distribution is known from monitoring



Outline

- Optimization of timeout value
- Optimization of task granularity
- Analysis of pilot-job experiments
- Analysis of workflow experiments

Optimization of timeout value

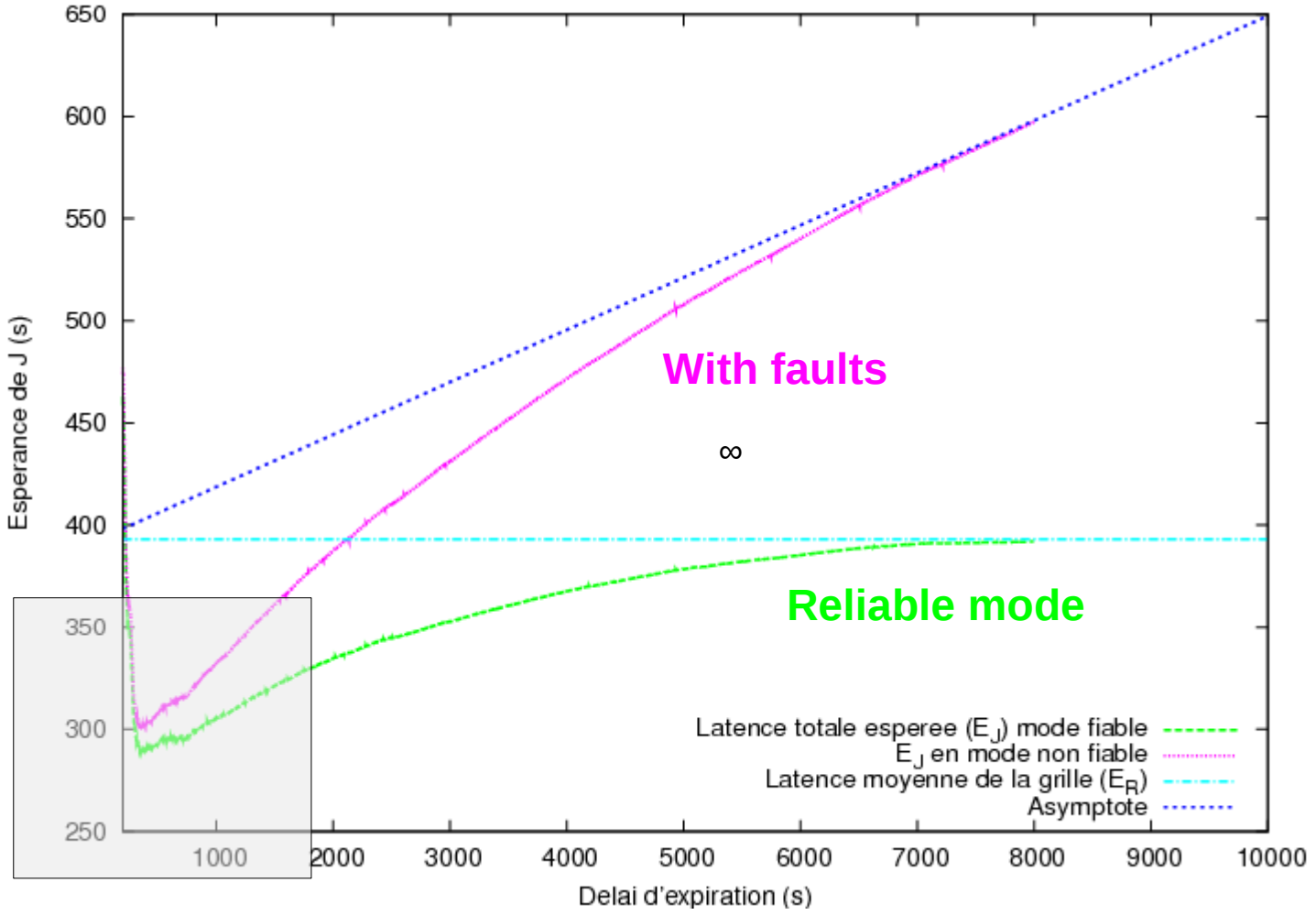
- Timeout value
  - Too short: overkilling
  - Too long: useless
- Hypotheses
  - Timeout => cancel + resubmit
  - Negligible cost for cancelation/resubmission (independent submissions)
- Expectation minimization
  - R: latency
  - t_∞: timeout value
  - ρ: fault ratio

$$E_J(t_\infty) = \frac{1}{F_R(t_\infty)} \int_0^{t_\infty} u f_R(u) du + \frac{t_\infty}{(1-\rho)F_R(t_\infty)} - t_\infty$$

Optimal $t_\infty < \infty$ <=> F_R is heavy-tailed

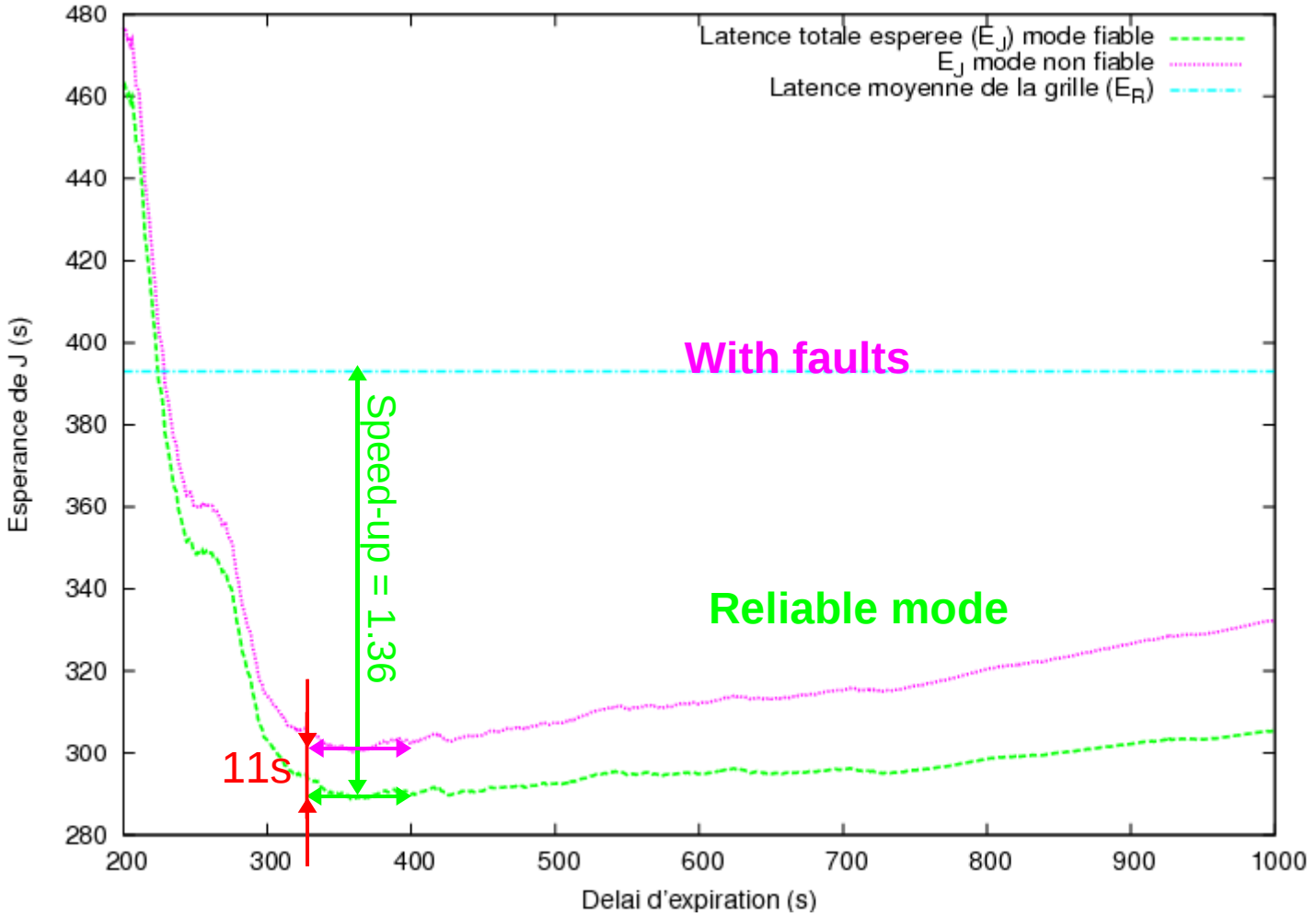
Timeout optimization

- Optimization on a latency distribution measured on EGEE:



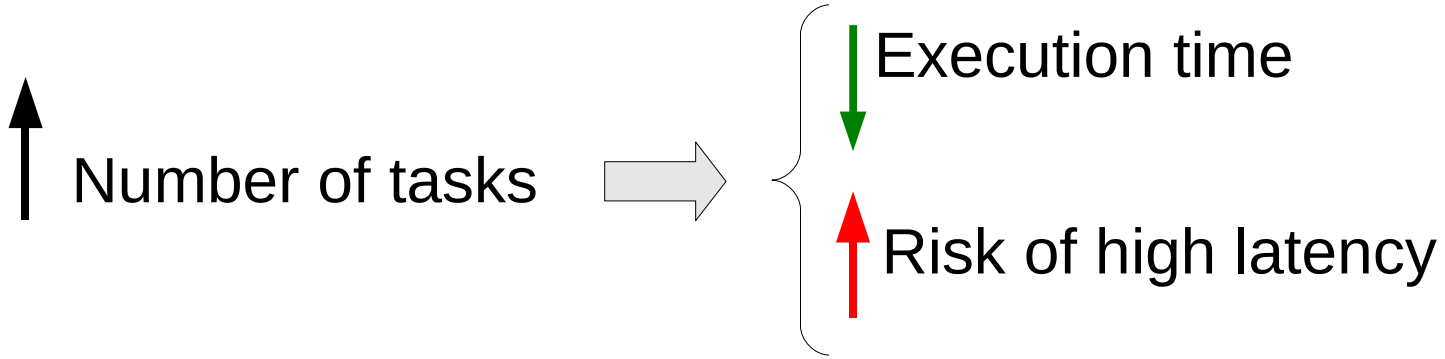
Timeout optimization

- Optimization on a latency distribution measured on EGEE:



Optimization of task granularity

- Granularity: number of tasks to submit for a work



- Total elapsed time for a work (duration w, granularity n):

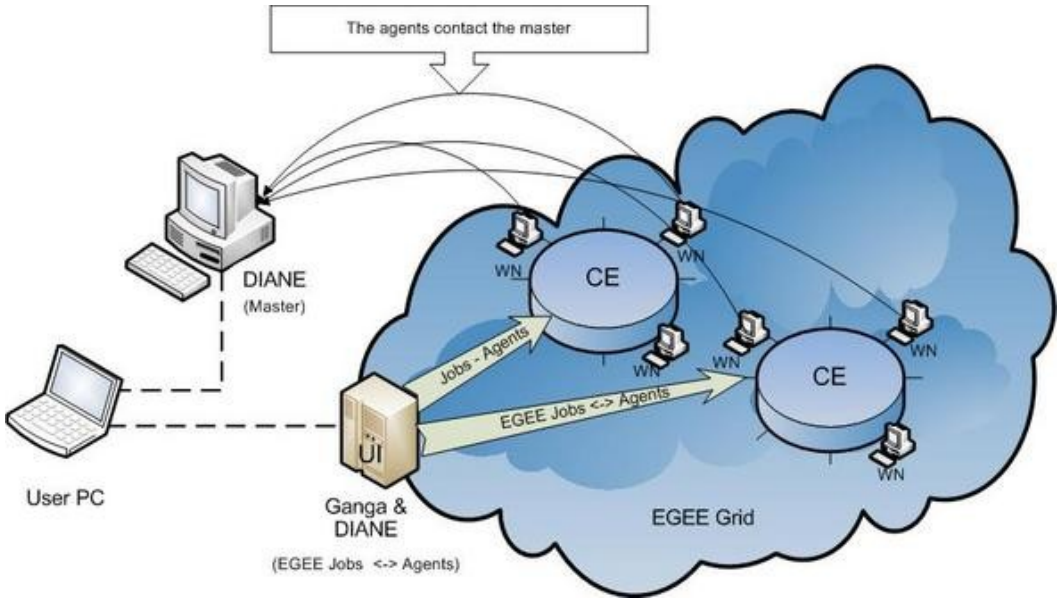
$$H = \max_{i=1..n} (R_i + w/n)$$

- Expectation minimization:

$$E_H(n) = \int_{\mathbb{R}} n.t.f_R(t).F_R(t)^{n-1} dt + \frac{W}{n}$$

Pilot-job experiments

- Pilot-job “pull” model



- Improves
  - Fault-tolerance
  - Responsiveness
  - Load balancing

Analysis of pilot-job experiments

- Number N of available pilots
  - pdf of N

$$f_N(k,t) = \binom{n}{k} F_L(t)^k (1 - F_L(t))^{(n-k)}$$

(n: total number of submitted pilots)

  - Expectation / stdev

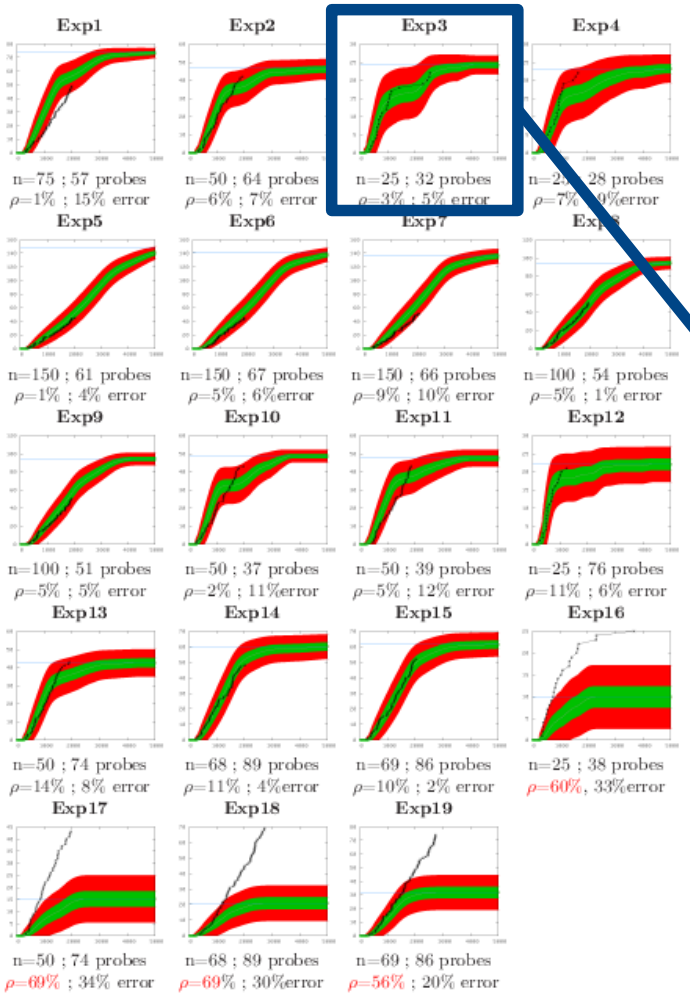
$$E_N(t) = nF_L(t) \qquad \sigma_N(t)^2 = n(1 - F_L(t))F_L(t)$$

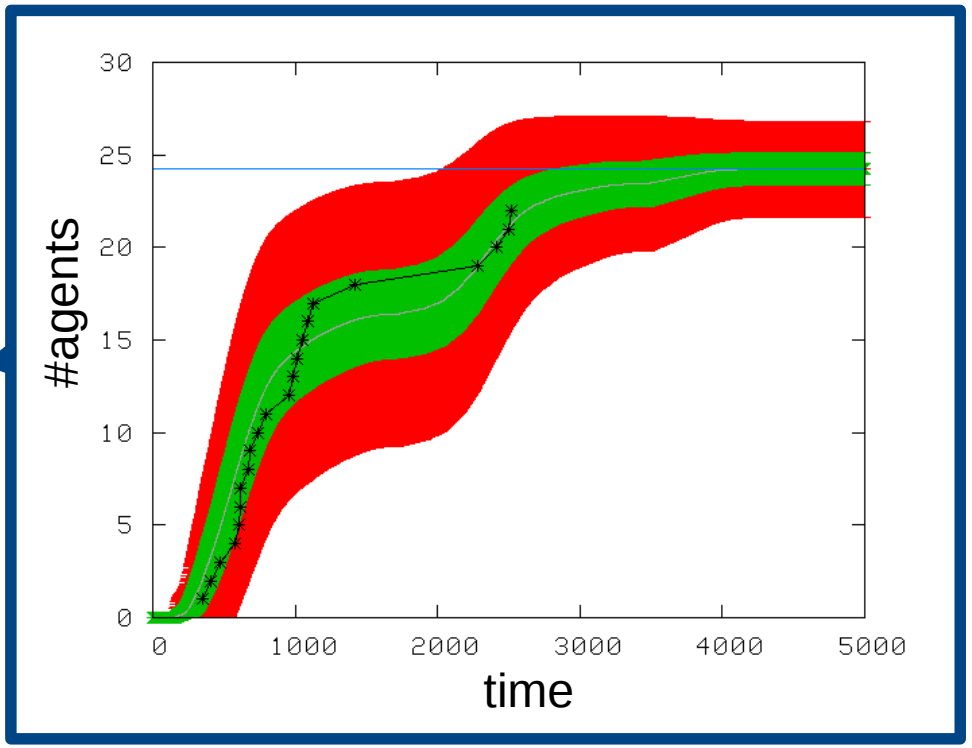
- Makespan of the experiment

$$\int_a^{E_M(n)} F_L(u)du = \frac{w_0}{n}$$

(w_0: total work time)

Number of available pilots





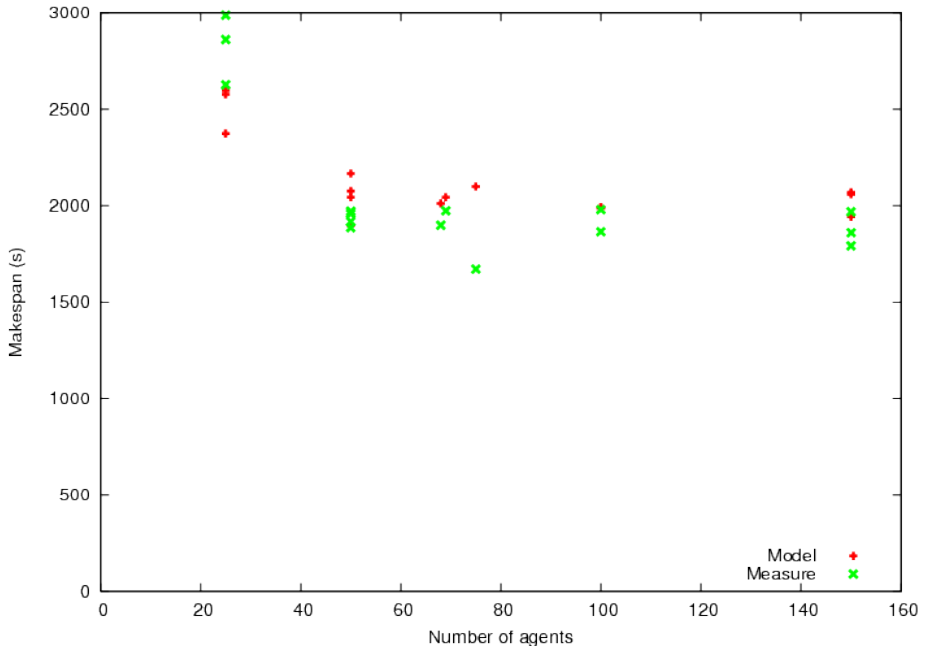
- 7% mean error between model and measures (Exp1 to 15)

Makespan analysis

- Makespan of the experiment

$$\int_a^{E_M(n)} F_L(u)du = \frac{w_0}{n}$$

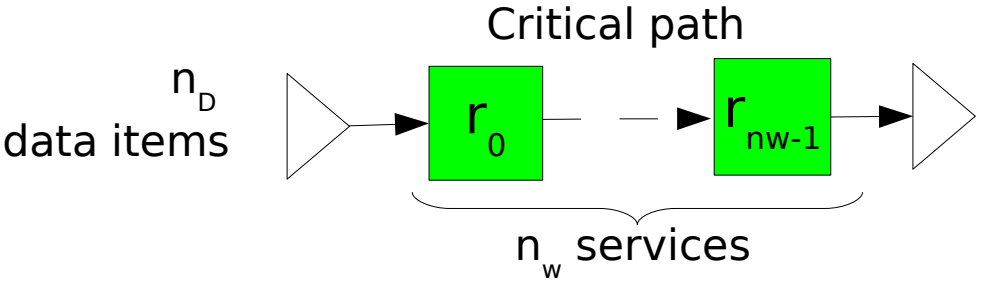
- Model VS measures
  - Mean error: 5'27"



- Estimation of w_0 by least-square minimization
  - Estimated: 11h ; Measured: 11h13'

Workflow analysis

- Application described as a workflow of services

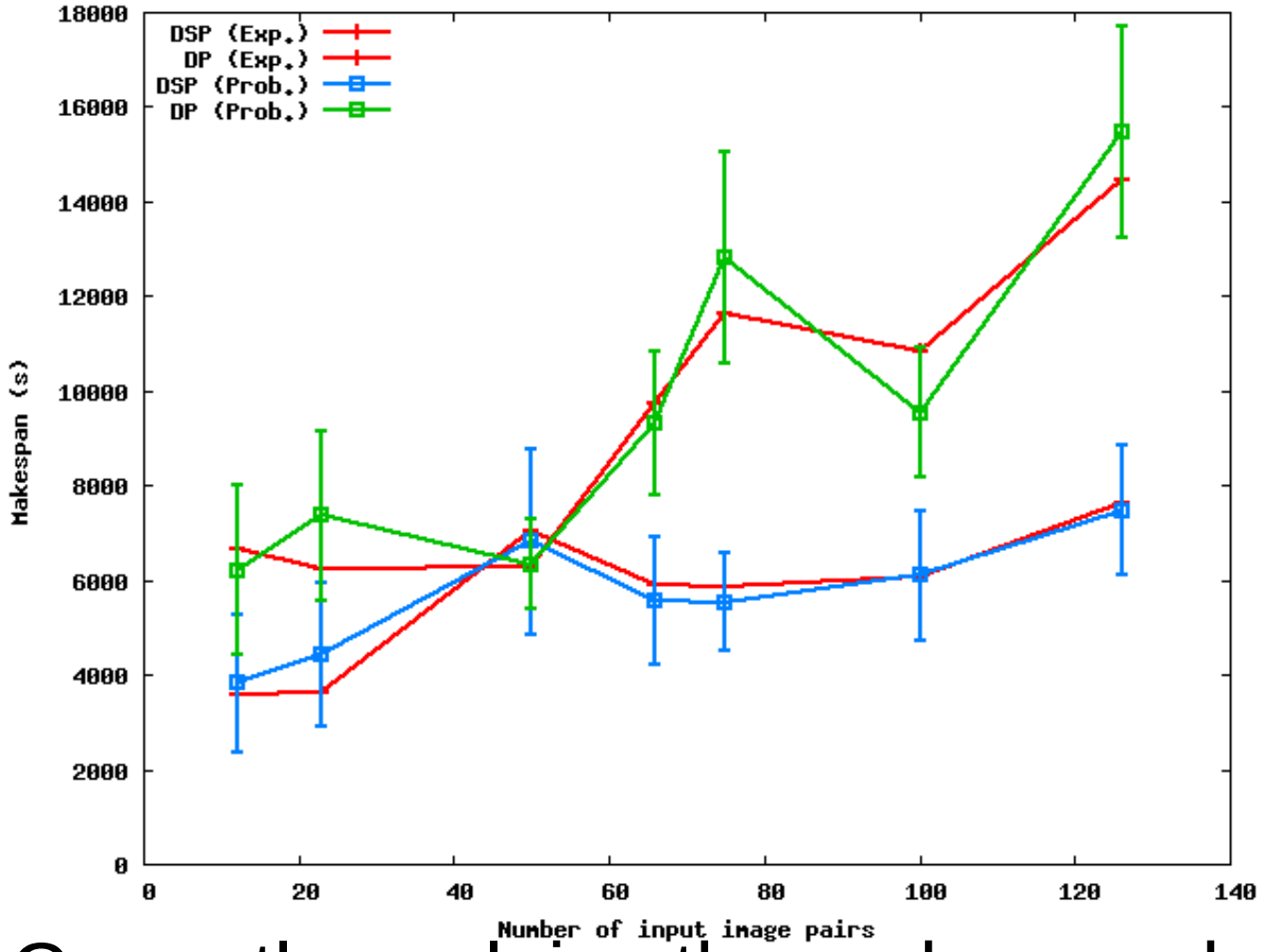


- Makespan determination

	Probabilistic
Data Parallelism only	$\sum_{i<n_W} \max_{j<n_D} \{r_i + R_{i,j}\}$
Data and Service parallelism	$\max_{j<n_D} \left\{ \sum_{i<n_W} (r_i + R_{i,j}) \right\}$

  - Ri,j supposed to be iid variables
  - Then compute expectation and standard deviation

Workflow makespan analysis



Experiment
Data and Service parallelism
Data parallelism only

(error bars cover ±3σ)

- Correctly explains the makespan behavior
  - Local makespan decrease w.r.t #inputs
  - Singular DSP > DP for 48 inputs

Conclusion

- General approach applicable to several problems
  - Optimization of timeout value
  - Optimization of task granularity
  - Analysis of pilot-job experiments
  - Analysis of workflow experiments
  - ...
- Drastic simplification of the middleware behavior
  - Still yields quite accurate results
- Not exploited in production yet
  - Need for a monitoring service
  - Application parameters have to be estimated
  - Need for a global model