24th International Parallel and Distributed Processing Symposium – 2010

Investigating the robustness of adaptive dynamic loop scheduling on heterogeneous computing systems

Srishti Srivastava[a],
Ioana Banicescu[a,b], Florina Ciorba[c]

April 23, 2010

[a] Department of Computer Science and Engineering
[b] Center for Computational Science
[c] Center for Advanced Vehicular Systems

NSF Grant #0934393, NSF Grant NSF DBI-0923469, NSF Grant NSF EPS 0556308, DOE Grant #008860-013

Mississippi State UNIVERSITY

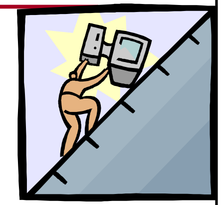
JAMES WORTH BAGLEY COLLEGE OF ENGINEERING MISSISSIPPI STATE UNIVERSITY

CAVS

Outline

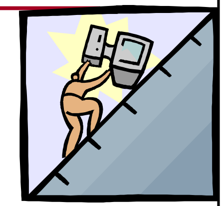
- ➔ **Motivation & approach**
- Contribution
- Background & related work
- Robustness metrics design
- Usefulness & cost analysis
- Conclusions & future work

Motivation



- New classes of applications: **(very) large & complex, computationally intensive, irregular behavior, contain large loops**
- High performance computing systems: unstable environments, complex to manage, computing resources vary in **type**, **quantity** and **availability** ➔ **multiple sources and types of uncertainty**

Motivation

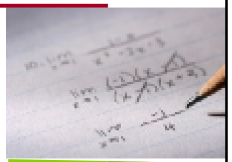


- Performance **is more than just execution time**:
  - Scalability
    - Execution time, efficiency and other metrics
  - Numerical efficiency
    - Accuracy, stability

Motivation

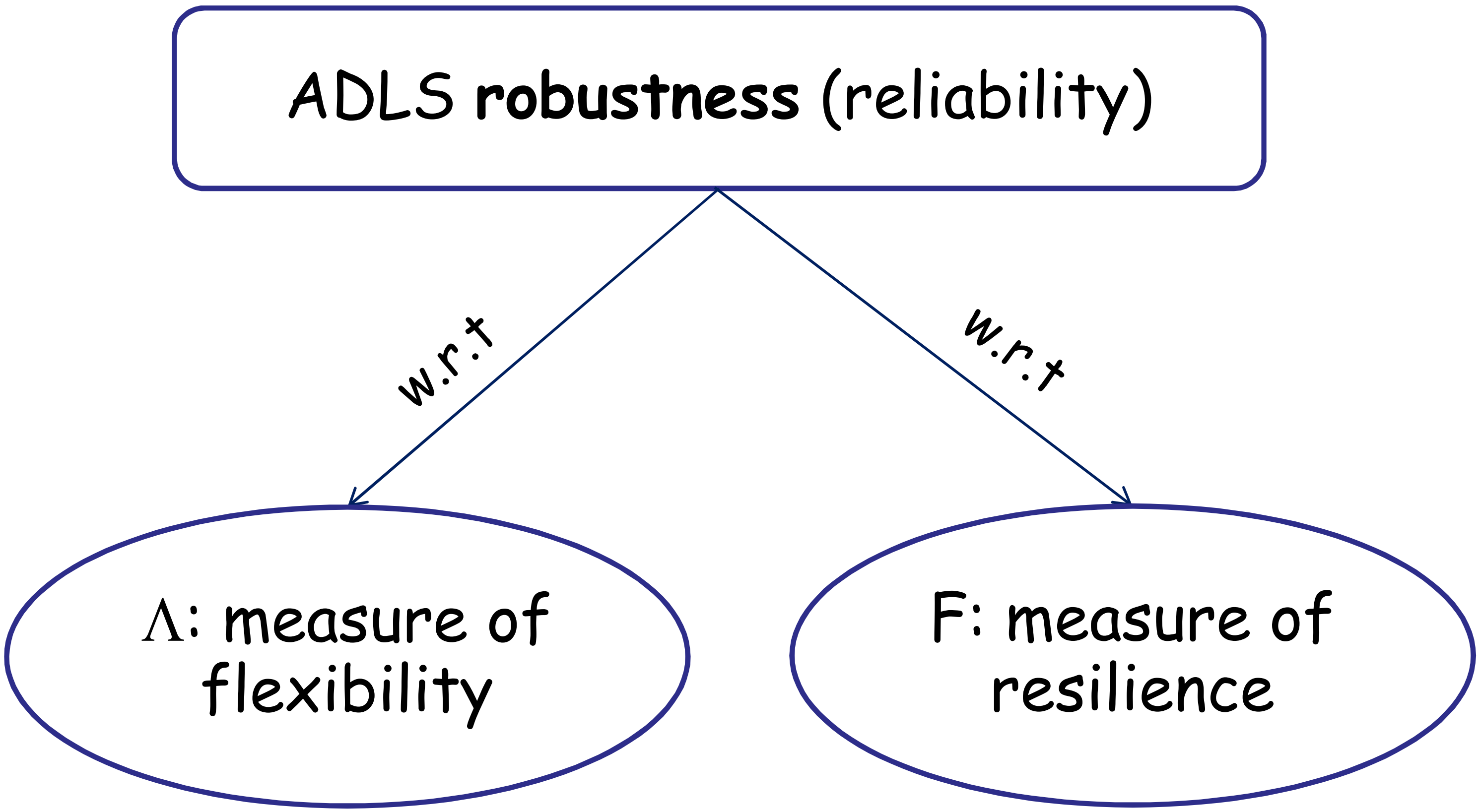
- Heterogeneity in computing resources evolves from variations in:
  - Number → failures (**fault tolerance** issues)
  - Load → availability (**load balancing** issues)

Challenges

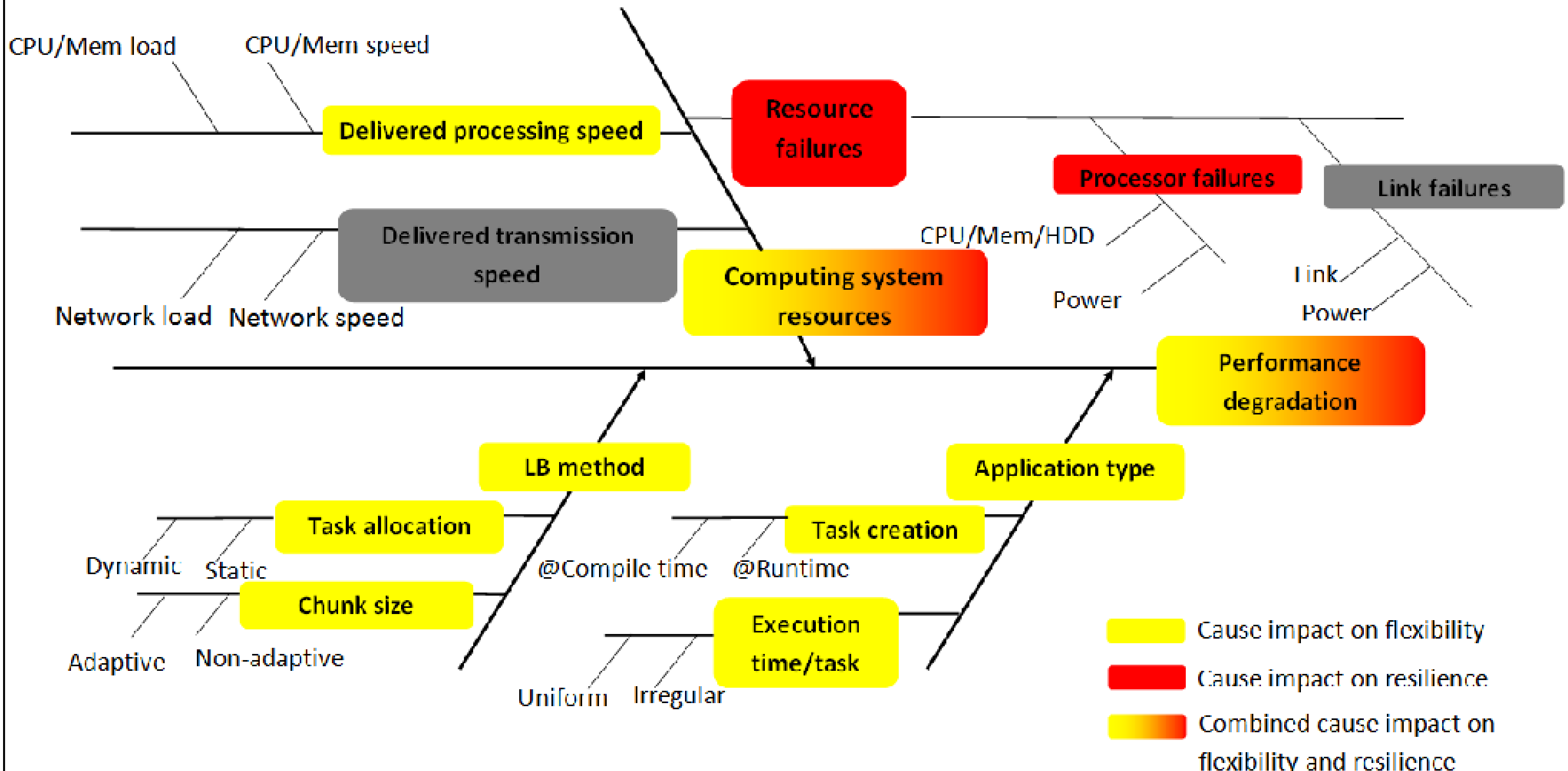


- User goals: **unchanged**: optimal results with minimum cost/effort
- Tradeoff between numerical efficiency and scalability
- Tradeoff between scheduling overhead and load imbalance
- How to execute large scientific applications on today's heterogeneous systems in a **flexible** and **resilient**, taken together as ***robustness***, manner?

Motivation

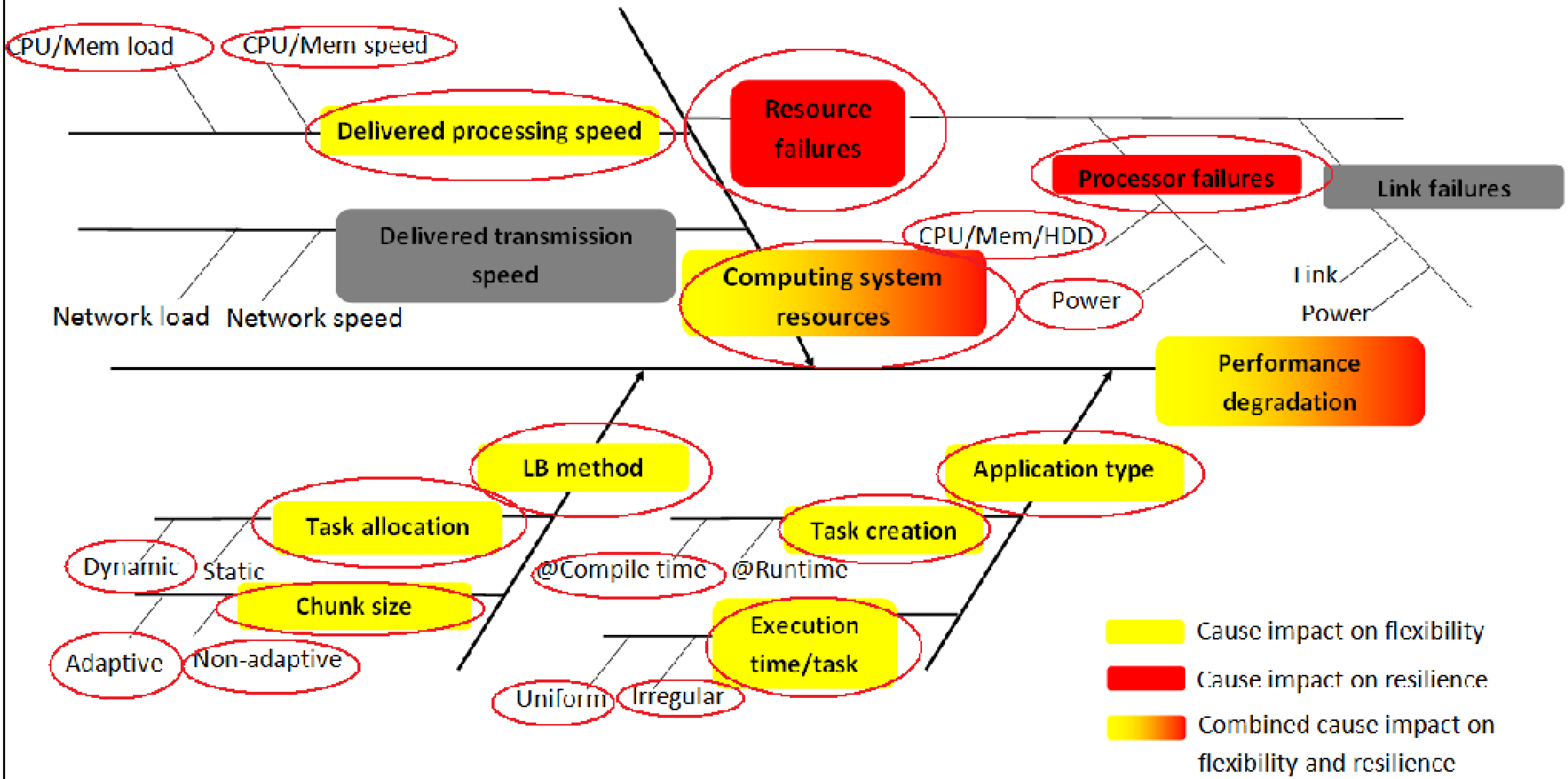

ADLS **robustness** (reliability)
w.r.t
w.r.t
Λ: measure of flexibility
F: measure of resilience

Approach: cause-and-effect analysis



- Fishbone diagram illustrating the relationships between the multiple possible causes & their effects for the problem of flexible and reliable execution of scientific applications

Approach: cause-and-effect analysis



Outline

- Motivation & approach
- ➔ **Contribution**
- Background & related work
- Robustness metrics design
- Usefulness & cost analysis
- Conclusions & future work

Contribution

Contribution: **flexibility** and **resilience metrics** to quantify the **robustness** (reliability) of 3 Adaptive **DLS** (ADLS) methods against *load variations* and *resource failures*, respectively

We propose:

- Use of the FePIA procedure to design metrics to measure robustness of batched AWF, chunked AWF, and AF to schedule large loop iterations in a **flexible** and **resilient** manner on large-scale heterogeneous computing systems

Outline

- Motivation & approach
- Contribution
- ➔ **Background & related work**
- Robustness metrics design
- Usefulness & cost analysis
- Conclusion & future work

Background & related work

- **Robustness** encompasses *flexibility and resilience*
- Previous study shows it is feasible to develop a common metric for all non-adaptive and one adaptive DLS techniques

- **What is missing?**

  - A more general approach needed to address robustness for *more than one method* (DLS or ADLS) and for *more than one application*

Outline

- Motivation & approach
- Contribution
- Background & related work
- ➔ **Robustness metrics design**
- Usefulness & cost analysis
- Conclusions & future work

Robustness metrics – FePIA Design steps

S.1 Identify the set of performance features of the DLS method **(Φ)**

S.2 Identify the set of parameters perturbing the performance of the method **(Π)**

S.3 Identify and clarify the impact of the perturbation parameters on the performance features of the DLS method **($\Phi(\Pi)$)**

S.4 Identify the analysis to determine the robustness of the DLS method (**$r(\varphi,\pi)$**) – robustness radius, **$\rho(\Phi,\Pi)$** – robustness metric)

We use these steps to design the following metrics:

$\rho(\Phi_1,\Pi_1)$: Robustness against *load variations* (flexibility)

$\rho(\Phi_2,\Pi_2)$: Robustness against *resource failures* (resilience)

Robustness metrics – Notation

Notation	Definition
N	total number of tasks
N^{resch}	# of tasks that need to be *rescheduled*
a_i	i-th task, $1 \leq i \leq N$
P	total number of processors
m_j	j-th processor, $1 \leq j \leq P$
T_j	*execution* time of task a_i on m_j
T_{ij}^{W2F}	*communication* time between m_j and foreman for a_i
T_{ij}^{W2W}	*communication* time between m_j and any other worker for a_i
$\mathcal{ET}_j$	*finishing* time of all tasks computed by m_j
T_{PAR}	total parallel execution time for the N tasks
$T_{SEQ}^{fastest}$	execution time of the fastest sequential version of the application
S_p	Speed up of the parallel system defined as $\frac{T_{SEQ}^{fastest}}{T_{PAR}}$
$\lambda = [\lambda_1 \ldots \lambda_P]^T$	vector of processors *load* (= system load)
$\mathbf{F} = [\mathbf{f}_1 \ldots \mathbf{F}_P]^T$	resources status vector (active/*failed*)
$\Phi = \{\phi_1, \ldots\}$	set of *performance* features
$\Pi = \{\pi_1, \ldots\}$	set of *perturbation* parameters
τ_1, τ_2, τ_3	tolerance factors for performance features
$r_{DLS}(\,,\,)$	robustness radius
$\rho_{DLS}(\,,\,)$	robustness metric
W	normalized weight of processors in WF, AWF, AWF-B, AWF-C

Flexibility: robustness w.r.t load variations

Assumptions regarding variations that can occur in the load at run-time:

A.1 variations of individual **worker** loads are **mutually independent**

A.2 individual **worker** load variations may or may **not** occur **simultaneously**

A.3 ADLS have load variation **detection** & **monitoring** mechanisms

- These are realistic assumptions that simplify the cause-and-effect analysis for this metric.

Flexibility: robustness w.r.t load variations

S.1 Performance features set: $\Phi_1 = \{\varphi_1\} = \{\epsilon T_j\}$

S.2 Perturbation parameters set: $\Pi_1 = \{\pi_1\} = \{\lambda_j\}$

S.3 Impact of Π_1 on Φ_1 : for all {tasks i | i executed on m_j in presence of λ_j}

$$\epsilon T_j\,(\lambda_j) = \Sigma_{i,j}^{N,P}\,(T_{ij}(\lambda_j) + T_{ij}^{w2f}(\lambda_j) + T_{ij}^{w2w}(\lambda_j))$$

S.4 Analyze the robustness radius and define the robustness metric:

$\{\, \lambda_j \in \langle\, \lambda_j', \lambda_j'' \,\rangle \mid (\epsilon T_j\,(\lambda_j) = \tau_1 \cdot \epsilon \tau_j^{orig}) \wedge (1 \le j \le P)\,\}$, λ_j^{orig} – initial load on m_j

Flexibility: robustness w.r.t load variations

τ_1 - acceptable tolerance on ϵT_j or T_{PAR} impact caused by variations in λ_j or Λ

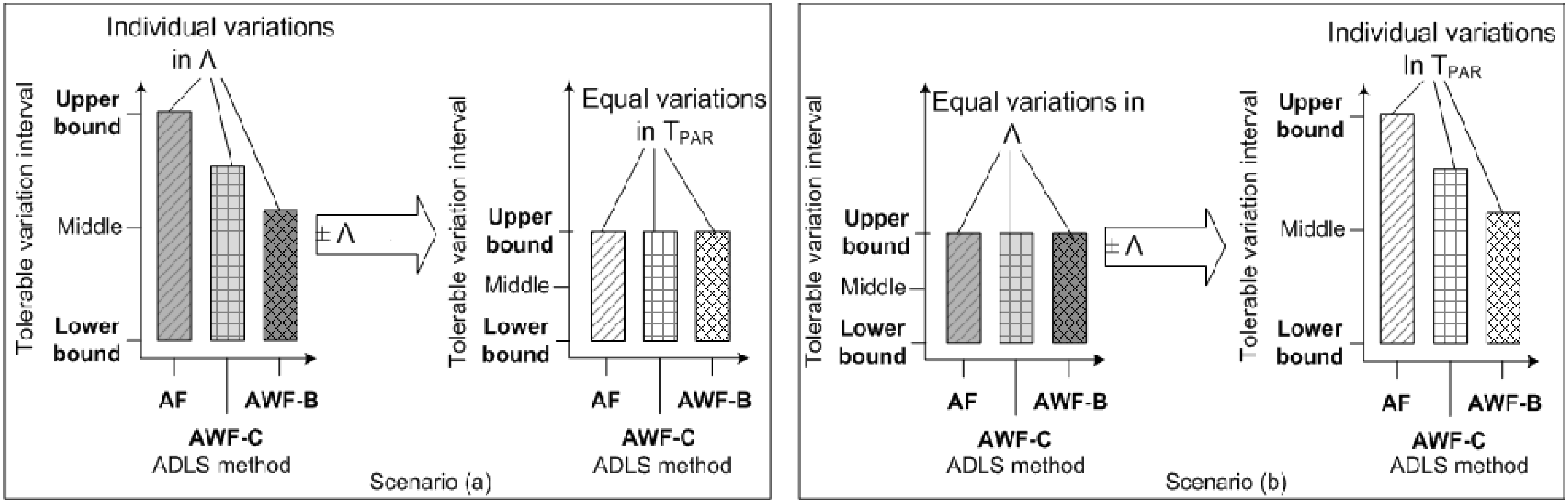
Robustness radii $r_{ADLS}(\varphi_i,\pi_i) = r_{ADLS}(\epsilon T_j,\lambda_j) = \max\|\lambda_j - \lambda_j^{orig}\|_1$, s.t. $\epsilon T_j(\lambda_j) = \tau_1 \cdot \epsilon T_j^{orig}$

Individual **robustness value** $\rho_{ADLS}(\Phi_1,\Pi_1) = \mathbf{max}(r(\epsilon T_j,\lambda_j))$

$\forall \varphi_i \in \Phi_1$ and $\pi_i \in \Pi_1$

Robustness metric $\rho_{metric}(T_{PAR},\Lambda) = \min(\rho_{ADLS}(\epsilon T_j,\lambda_j))$

Possible scenarios to determine the flexibility of ADLS techniques



Scenario (a)

Choose the ADLS method that has the *lowest impact on ADLS* performance AND can handle the *largest variation in* Λ

Scenario (b)

Choose the ADLS method that has the *lowest impact on ADLS* performance for a *fixed variation in* Λ

Resilience: robustness w.r.t resource failures

Assumptions regarding failures that can occur in the system and their handling mechanisms:

A.4 only resources associated with **worker** processors fail

A.5 resource failures occur **simultaneously**

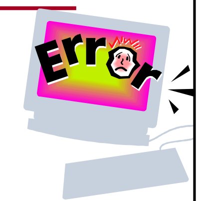
A.6 resource failures are **mutually independent**

A.7 resource failures are **permanent**

A.8 ADLS have **fault discovery** & **fault-recovery** mechanisms

- These are realistic assumptions that simplify the cause-and-effect analysis for this metric.

Resilience: robustness w.r.t resource failures



S.1 Performance features set: $\Phi_2 = \{\varphi'_1, \varphi'_2\} = \{N^{resch}, T_{PAR}\}$

S.2 Perturbation parameters set $\Pi_2 = \{\pi'_1\} = \{\mathbf{F}\}$

S.3 Impact of Π_2 on Φ_2 :

$\varphi'_1 = f_{11}(\pi'_1)$: $N^{resch} = f_{11}(\mathbf{F})$, where $N^{resch}(\mathbf{F}) = N_p^{resch}(\mathbf{F}) + N_{lb}^{resch}(\mathbf{F}) \wedge$

$N^{resch}(\mathbf{F}) = f'(\text{ADLS method of choice})$

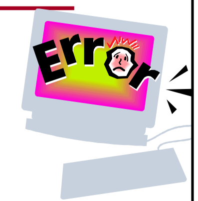
$\varphi'_2 = f_{21}(\pi'_1)$: $T_{PAR} = f_{21}(\mathbf{F})$, where $T_{PAR}(\mathbf{F}) = f''(\text{ADLS method \& recovery mech.})$

S.4 Analyze the robustness radius/radii and define the robustness metric:

$$\{\, \mathbf{F} \mid (N^{resch}(\mathbf{F}) \leq \tau_2 \cdot N) \wedge (\exists\, \mathbf{F}' \text{ s.t. } N^{resch}(\mathbf{F}') > \tau_2 \cdot N) \,\}$$

$$\{\, \mathbf{F} \mid (T_{PAR}(\mathbf{F}) \leq \tau_3 \cdot T_{PAR}(\mathbf{F}^{orig})) \wedge (\exists\, \mathbf{F}' \text{ s.t. } T_{PAR}(\mathbf{F}') > \tau_3 \cdot T_{PAR}(\mathbf{F}'^{orig}))\}$$

Resilience: robustness w.r.t resource failures (cont.)



Robustness radii (tolerance intervals):

$$r_{ADLS}(\varphi'_1,\pi'_1)=r_{ADLS}(N^{resch},\mathbf{F}) = \max||\mathbf{F}-\mathbf{F}^{orig}||_1, \text{ such that}$$
$$(N^{resch}(\mathbf{F})\le\tau_2\cdot N) \wedge (\exists\, \mathbf{F'} \text{ s.t. } N^{resch}(\mathbf{F'}) > \tau_2\cdot N)$$

$$r_{ADLS}(\varphi'_2,\pi'_1)=r_{ADLS}(T_{PAR},\mathbf{F}) = \max||\mathbf{F}-\mathbf{F}^{orig}||_1, \text{ such that}$$
$$(T_{PAR}(\mathbf{F})\le\tau_3\cdot T_{PAR}(\mathbf{F}^{orig})) \wedge (\exists\, \mathbf{F'} \text{ s.t. } T_{PAR}(\mathbf{F'}) > \tau_3\cdot T_{PAR}(\mathbf{F'}^{orig}))$$

τ_2 – acceptable tolerance on N^{resch} impact caused by # of failures in **F**.

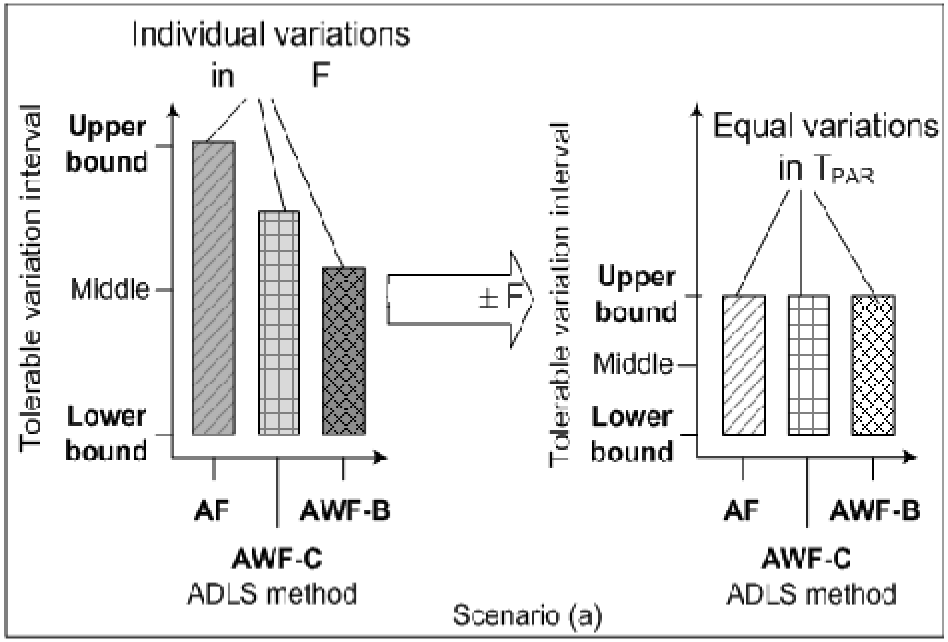
τ_3 – acceptable tolerance on T_{PAR} impact caused by # of failures in **F**.

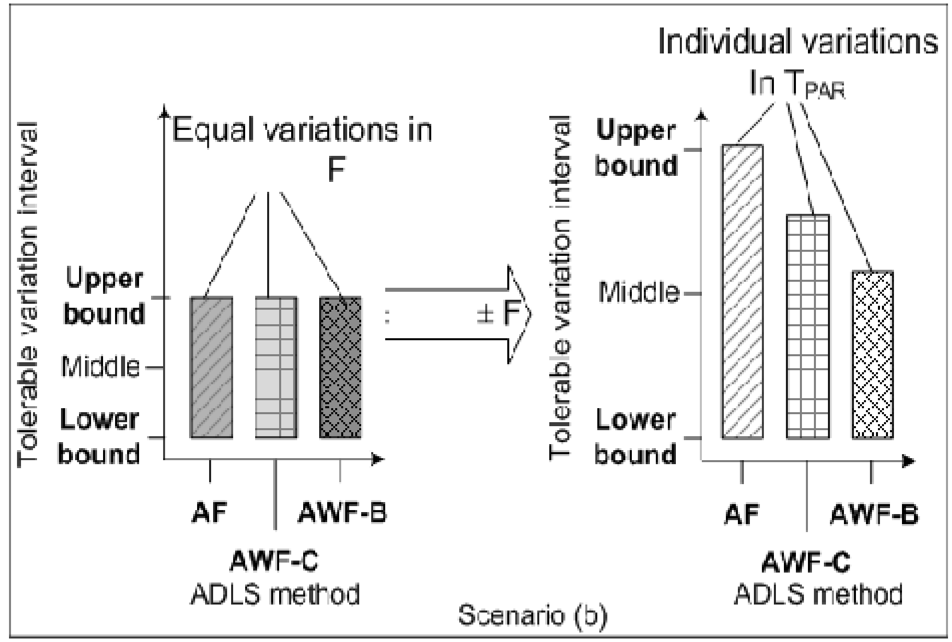
Individual **robustness value** $\rho_{ADLS}(\Phi_2,\Pi_2) = \mathbf{max}(r_{ADLS}(\varphi'_1,\pi'_1), r_{ADLS}(\varphi'_2,\pi'_1))$

Robustness metric

$$\rho_{metric}(T_{PAR},\mathbf{F}) = \min(\rho_{ADLS}(N^{resch},\mathbf{F}),\rho_{ADLS}(T_{PAR},\mathbf{F}))$$

Possible scenarios to determine the resilience of ADLS techniques





Scenario (a)

Choose the ADLS method that has the *lowest impact on ADLS* performance AND can handle the *largest variation in* **F**

Scenario (b)

Choose the ADLS method that has the *lowest impact on ADLS* performance for a *fixed variation in* **F**

Outline

- Motivation & approach
- Contribution
- Background & related work
- Robustness metrics design
- ➔ **Usefulness & cost analysis**
- Conclusions & future work

Usefulness

- The usefulness of proposed metrics is twofold:
  - the metrics can be formulated offline with application/ system/ algorithm-specific initial values and integrated into the master to guide and adapt autonomously the scheduling decisions
  - the metrics are usable in conjunction with other desired performance metrics (e.g., makespan) for differentiating among DLS that have similar performance w.r.t makespan
- Tolerance factors (i.e., τ_1, τ_2, τ_3) must accurately reflect real life conditions and realistic scenarios

Usefulness

Tolerance factor	Depends on	Best case	Worst case	Average case
τ_1	Application type	1.0	1.5	1.25
τ_2	ADLS method of choice	0%of N	50%of N	25% of N
τ_3	ADLS method of choice, # of failures, fault detection & recovery mechanism	S_p^{ideal}	1 (no S_p)	$\frac{S_p}{2}$

- A careful choice of the tolerance factors and incorporation into the adaptive DLS methods, renders the proposed metrics useful towards producing *efficient, qualitative* and *reliable* schedules for execution of large and complex scientific applications

Usefulness

- When considering a set of ADLS and DLS techniques one should select as *robustness metric* (ρ_{metric}) the one that gives the smallest robustness radius among all techniques.
- In order to determine the most robust algorithm one should select the DLS or the ADLS technique with the highest robustness value ($\rho_{ADLS}^{max} = k \cdot \rho_{metric}$) as derived from the selected robustness metric.

Cost analysis

The computational complexity of each metric is driven by the computational complexity of calculating robustness radii for each metric

- General optimization problem for robustness:

$$\textbf{Maximize } \sum_k ||\pi_k-\pi_k^{orig}||_p,\ k>0$$

$$\textbf{subject to } \beta^{min} \le \{\Phi=f(\pi_k)\} \le \beta^{max}$$

where: $||\pi_k||_p$ is the L_p-norm of the perturbation parameter π_κ

$\langle \beta^{min}, \beta^{max} \rangle$ is the tolerance interval,

$\Phi=f(\pi_k)$ is equivalent to $\Phi(\pi_k) = \tau \cdot \Phi(\pi_k^{orig})$

τ is the tolerance factor of the performance feature

Cost analysis

- Optimization problem for **flexibility**:

$$\textbf{maximize } ||\Lambda-\Lambda^{orig}||_1 \textbf{ subject to}$$

$$T_{PAR}(\Lambda) \le \tau_1 \cdot T_{PAR}(\Lambda^{orig}) \wedge T_{PAR}(\Lambda) \le T^{fastest}{}_{SEQ}$$

- Optimization problem for **resilience**:

$$\textbf{maximize } ||F-F^{orig}||_1 \textbf{ subject to}$$

$$N^{resch}(F) \le \tau_2 \cdot N \wedge T_{PAR}(F) \le \tau_3 \cdot T_{PAR}(F^{orig}) \wedge T_{PAR}(F) \le T^{fastest}{}_{SEQ}$$

Fact: All norms $|| \Lambda ||_p$ are convex functions

Dilemma: $\Phi(\Pi)$: convex or concave function?

If $\Phi(\Lambda)$ and $\Phi(F)$ are linear or convex functions both metrics become *convex optimization problems with* inexpensive optimal solutions

If $\Phi(\Lambda)$ and $\Phi(F)$ are concave then near-optimal solutions cannot be found with optimal costs

Outline

- Motivation & approach
- Contribution
- Background & related work
- Robustness metrics design
- Usefulness & cost analysis
- ➔ **Conclusions & future work**

Conclusions & future work

Pre-∃ inherent robustness of AWF, AWF-B, AWF-C and AF against load variations

☺ The proposed metrics bring out the most adaptive and efficient DLS algorithms to state-of-the-art performance

☺ Careful choice of tolerance values results in qualitative, efficient and reliable schedules in today's large scale, high-performance, heterogeneous systems

Conclusions & future work

- Consider multiple perturbation factors to devise more realistic metrics
- Use the metrics to measure the robustness of DLS techniques lower in the hierarchy
- **Simulate** and use the proposed metrics to test the robustness of the DLS and ADLS techniques using an event-based simulator
- **Incorporate** the proposed metrics in scheduling techniques, to test their robustness when executing real world irregular scientific applications on real heterogeneous computing systems

Summary

- New complex classes of applications; uncertain computing environments; performance is more than execution time
- Tradeoffs:
  - Numerical efficiency & stability
  - Scheduling overhead & load imbalance
- Robustness metrics
  - w.r.t load variations (flexibility)
  - w.r.t resource failures (resilience)
- Usefulness: automatic selection of robust, efficient & optimal scheduling techniques

Thank you

Questions?

Selected bibliography

- Ali, S., Siegel, H. J., Maciejewski, A. A.: ***Perspectives on Robust Resource Allocation for Heterogeneous Parallel and Distributed Systems***. Chapter 4 - Handbook of Parallel Computing Models, Algorithms and Applications. MK Publishing (2008)
- Banicescu, I., Ciorba, F. M., and Carino, R. L.: **Towards the Robustness of Dynamic Loop Scheduling on Large-Scale Heterogeneous Distributed Systems**. In Proceedings of the 2009 Eighth international Symposium on Parallel and Distributed Computing (ISPDC), IEEE, 129–132 (2009)
- Gribble, S.D.: ***Robustness in Complex Systems***. 8th Workshop on Hot Topics in Operating Systems (2001)