Result Error Detection on Heterogeneous and Volatile Resources Via Intermediate Checkpointing

Derrick Kondo[1], Filipe Araújo[2], Patricio Domingues[3], Luis Moura Silva[2]

[1]Laboratoire de Recherche en Informatique / INRIA Futurs, France
[2]University of Coimbra, Portugal
[3]School of Technology and Management, Polytechnic Institute of Leiria, Portugal

Background

- Desktop grids
  - Use free resources in Internet and Intranet environments
  - Orders of magnitude more powerful and cheaper than dedicated computing environments
    - Enabled computational science otherwise infeasible
    - About 50 applications use over 1 PetaFLOPS from millions of Internet resources

SETI@HOME

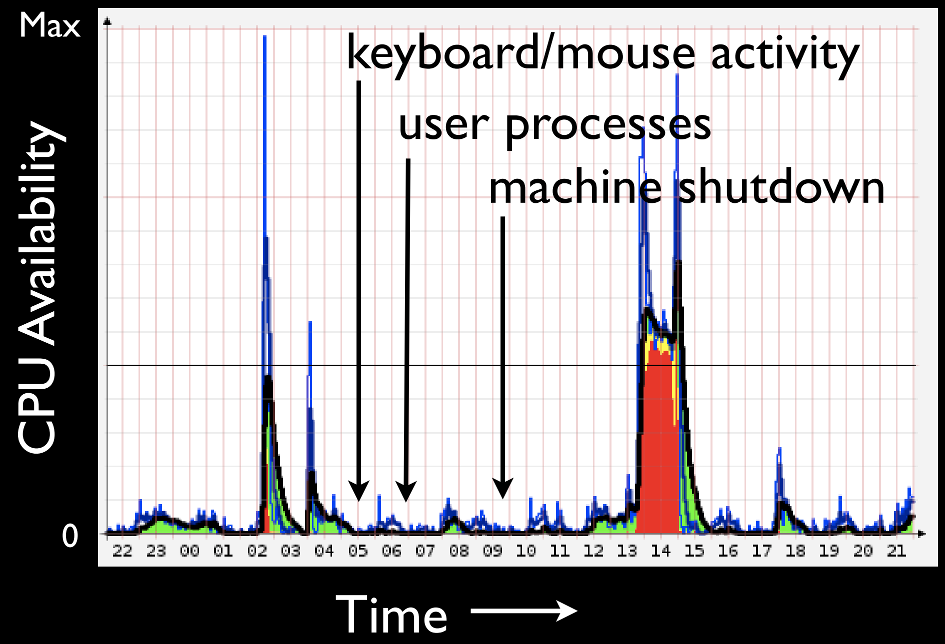
PROGRAMME DÉCRYPTHON

climateprediction.net

- - École Polytechnique (PROTEINS@home), France Telecom, CERN (EGEE), Sony, Microsoft, and others

Challenges

- Volatility
- Heterogeneity
- Anonymity
- Scalability



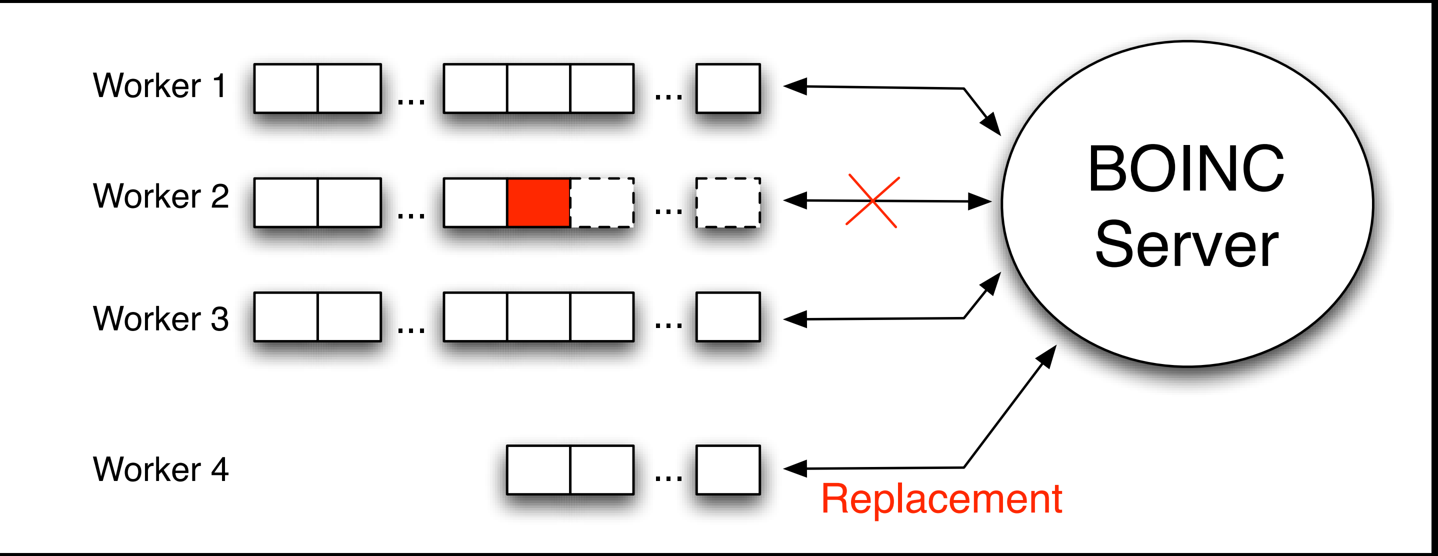
Implications for Result Correctness

- Volatility
  - Crash during I/O can cause unexpected state
- Heterogeneity
  - Discrepancies in computation on different CPU architectures
- Anonymity
  - Cannot trust the user or resource
- Scalability
  - Errors on inevitable for computation running thousands of hosts

Approach

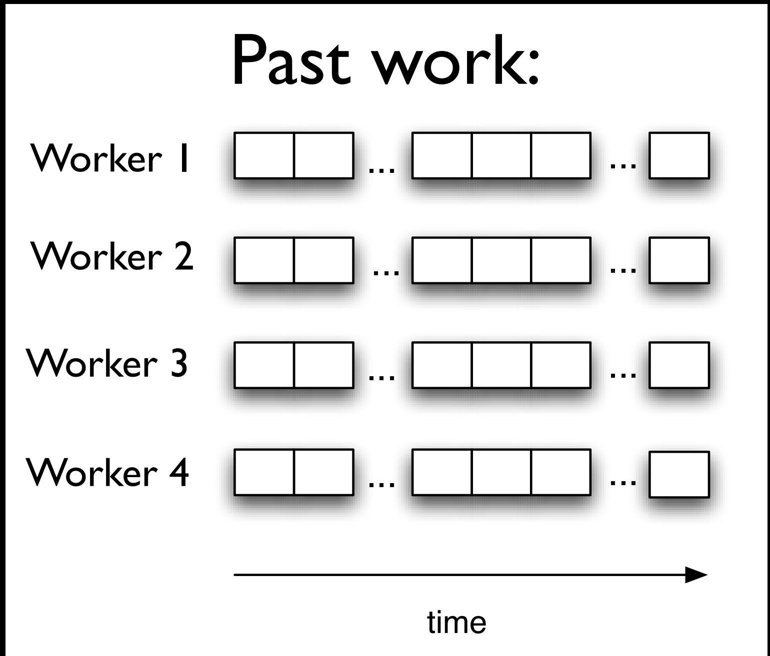
- Replication
  - Problematic especially for long-running computations
    - `climateprediction.net`, `climatechange`, `seasonalattribution` have workunits whose execution spans months
  - Compare message digest of each application-level checkpoint
    - Benefit is earlier detection of errors

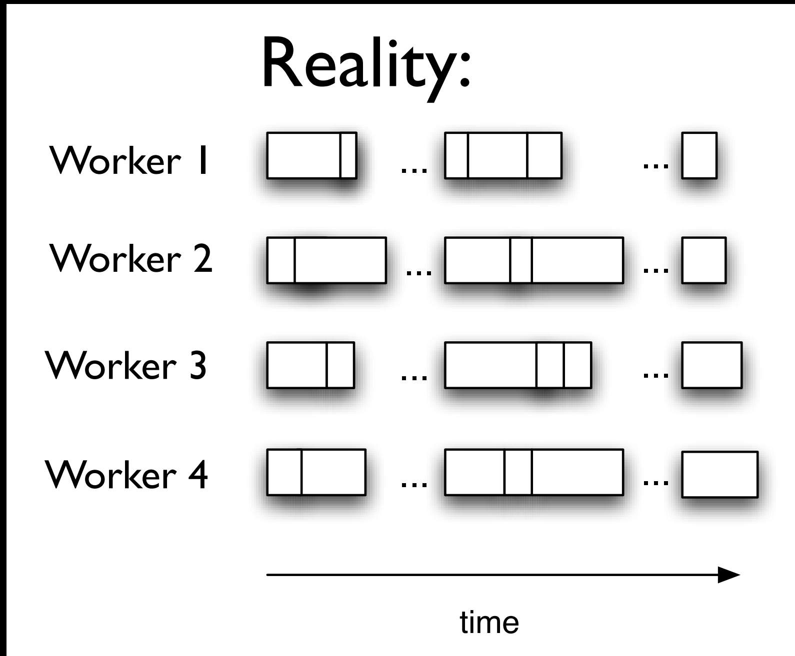
Illustration



Our Related Work

- Compare digests of intermediate checkpoints [Araujo2006]
    - Analysis assumed *uniform* segment completion times





Assumptions

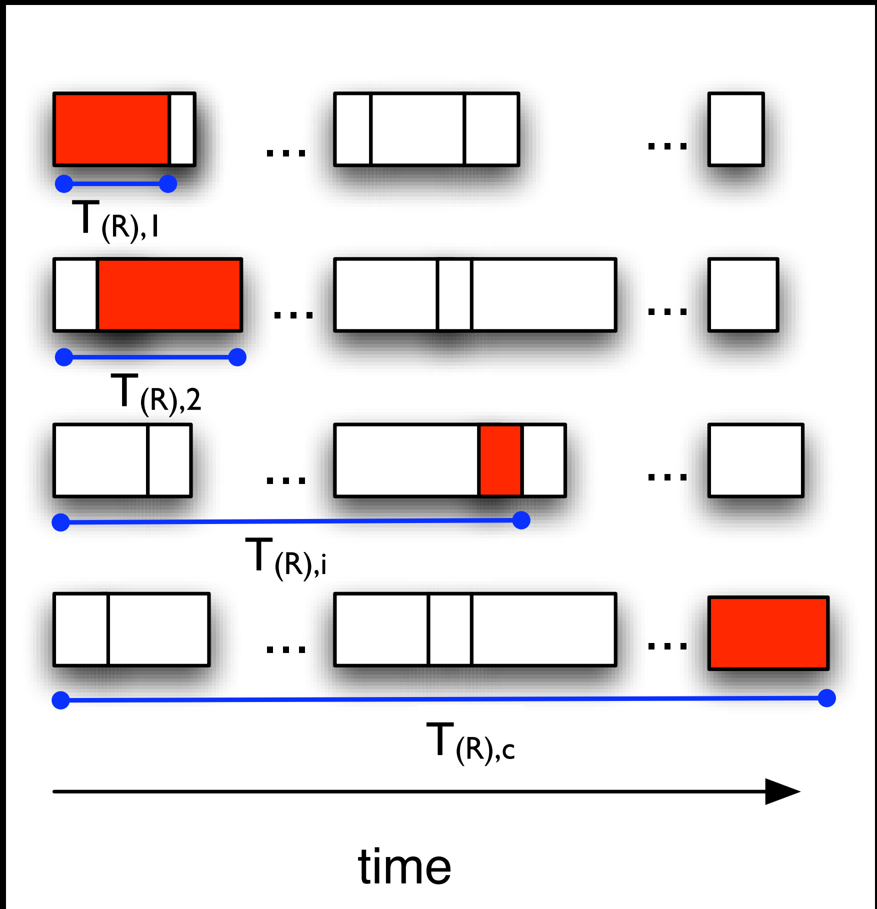
- Wait for digest of corresponding segment from all workers before doing comparison
- Assume time between consecutive segment checkpoints follows an exponential distribution with parameter λ
- If errors occur in multiple segments, the errors would be different
- (Loosen these assumptions later in simulation)

Definitions

Parameter	Definition
W	Benefit in time of intermediate checkpointing relative to the state-of-the-art method
$T_{k,j}$	Time from start of workunit to the time of checkpointing segment j on worker k
R	Number of workers on which a checkpointed task is replicated
c	Number of segments or equivalently checkpoints per task
p, v	p is the probability of getting an error within a segment on any host. $v = 1 - p$
X	Random variable distributed geometrically with parameters p and v representing the number of task segments before an error occurs

Intuition of Theoretical Analysis

Benefit Probability



$$(T_{(R),c} - T_{(R),1}) \times Pr(X = 1)$$

$$(T_{(R),c} - T_{(R),2}) \times Pr(X = 2)$$

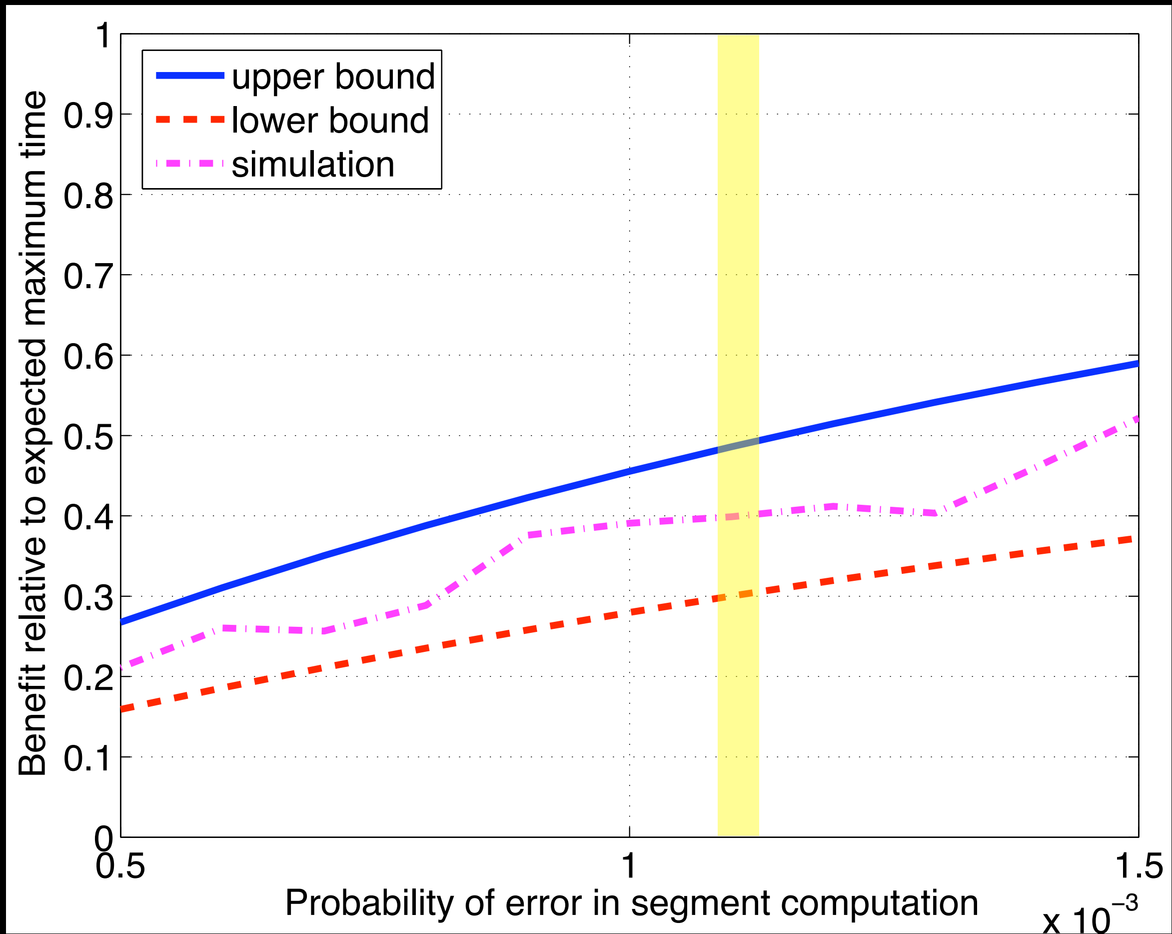
$$(T_{(R),c} - T_{(R),i}) \times Pr(X = i)$$

$$(T_{(R),c} - T_{(R),c}) \times Pr(X = c)$$

$$E[W] = E[T_{(R),c} - T_{(R),i}]$$

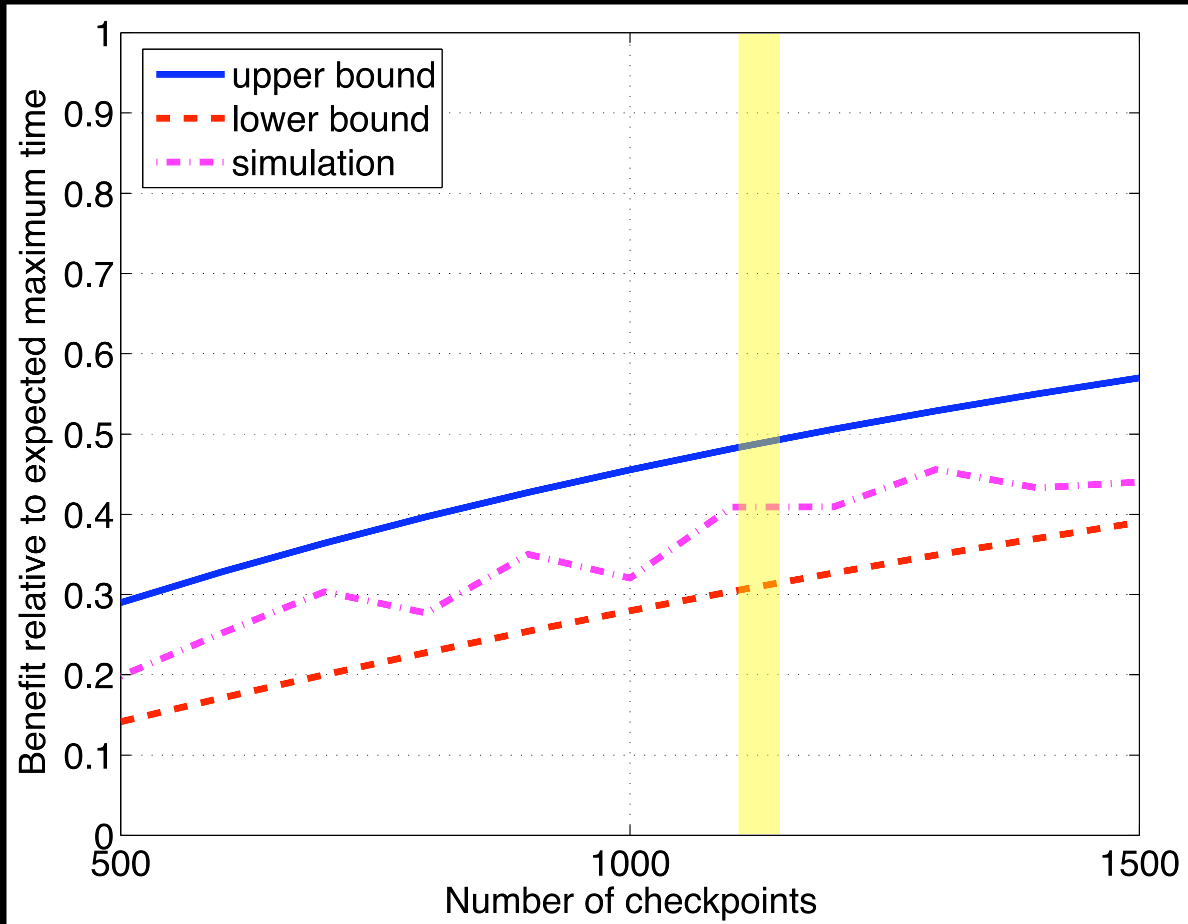
where $1 \leq i \leq c$

Benefits Varying Probability of Error



c=1000, variable p

Benefits Varying Checkpoint Frequency



variable c, p = 0.001

Related Work

- Error detection [Sarmenta01]
  - Majority voting
  - Spot-checking
  - Credibility-based methods

Summary & Future Work

- Gave theoretical upper and lower bounds on expected benefit for case where segment completion times are variable
- Ran simulations loosening assumptions and verified theoretical bounds
- Observed gains relative to the state-of-the-art often between 30-50%
- Use DHT's to compare checkpoint digests in a scalable way