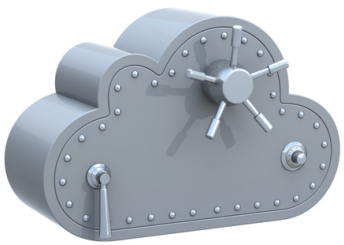


PhD thesis: Integrity issues on the cloud

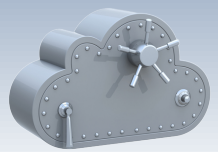
Jakub Muszyński (office E008)

22–23 November 2012

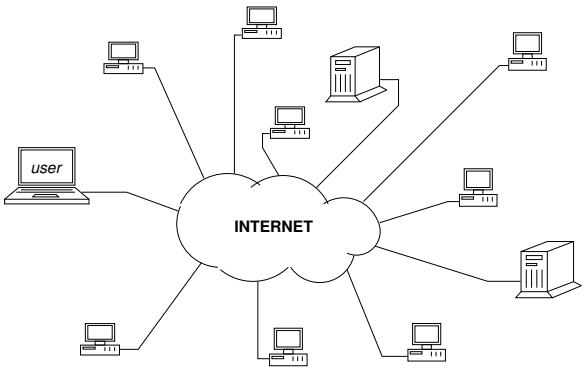
Computer Science and Communications

(CSC) Research Unit

Yes, it's **32** slides! ⇓

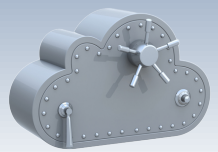


Volunteer Computing (VC) Platforms

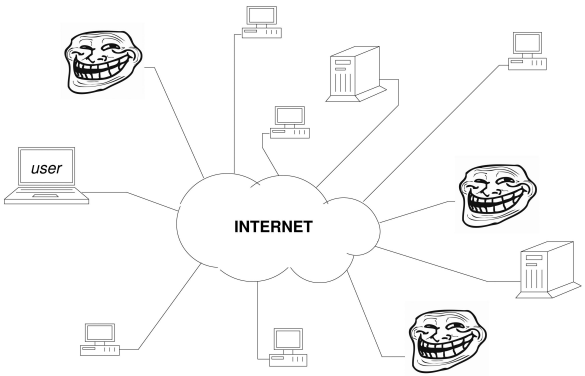


- Desktop Grids (DG) or DGVCS
- **Ex:** BOINC
  - ↪ ≈ 2 500 000 users
  - ↪ ≈ 8 000 000 machines
  - ↪ ≈ 7 000 teraFLOPS

- Steal computing cycle from idle computers (the case ≃ 75% of the time)
  - ↪ No guarantees about the security of a given machine
  - ↪ Heterogeneous platform with high volatility
  - ↪ Usually with reward system (credit points, hall of fame *etc.*)



Volunteer Computing (VC) Platforms

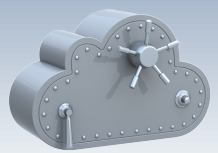


- Desktop Grids (DG) or DGVCS
- **Ex:** BOINC
  - ↪ ≈ 2 500 000 users
  - ↪ ≈ 8 000 000 machines
  - ↪ ≈ 7 000 teraFLOPS

- Steal computing cycle from idle computers (the case ≃ 75% of the time)
  - ↪ No guarantees about the security of a given machine
  - ↪ Heterogeneous platform with high volatility
  - ↪ Usually with reward system (credit points, hall of fame *etc.*)

Incentives attract cheaters!

- Seek to obtain rewards with little/no contribution to the system
  - ↪ **Cheating Faults** (*fault by value*) modelled by result falsification



Cheating Faults[1]

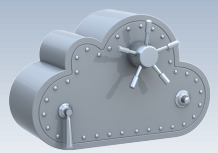
- Typically done via software tampering/man-in-the-middle attack
  - ↪ Sub-class of byzantine failures, more complex than **Crash Faults**
- Cheater knows everything
  - ↪ Every cheater wants to be maximally effective in his actions

[Lazy] Cheating Fault Modelisation

- For every task T in the system, exists its cheated version T'
  - ↪ **Prototype of T' unaffected** by cheating (easy to detect)
  - ↪ Result of T' altered **with worst impact** (delay the execution)
  - ↪ **Don't work more at T'** than at T

[1] Introduced in: S. Varrette et al. *Nature inspired Algorithm-Based Fault Tolerance on Global Computing Platforms. Application to Symbolic Regression*. 2008

UNIVERSITÉ DU LUXEMBOURG



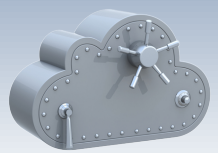
Fault Tolerance strategies

- Generic (*i.e.* algorithm-independent)
  - ↪ Checkpoint/rollback — crash faults
  - ↪ Duplication (result-checking etc.) — crash/cheating faults

Algorithmic Based Fault Tolerance (ABFT)

- Error detection/correction tailored to the algorithm performed
  - ↪ Resilience to byzantine failures producing falsified results.
- **Inherent tolerance of a limited number of faults**
  - ↪ Avoids overhead of checkpoint/restart methods
  - ↪ Mostly applied against **Crash Faults**

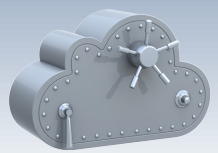
UNIVERSITÉ DU LUXEMBOURG



Done

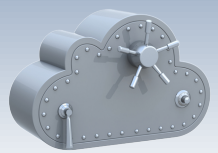
- **Convergence proofs** of dEAs despite crash-faults/cheaters published in [CAMWA12]
  - ↪ Based on Markov chains and transition kernels
- Hash function generation by means of Gene Expression Programming

[CAMWA12] J. Muszyński et al. *Convergence analysis of evolutionary algorithms in the presence of crash-faults and cheaters*. J. of Computers and Mathematics with Applications. 2012

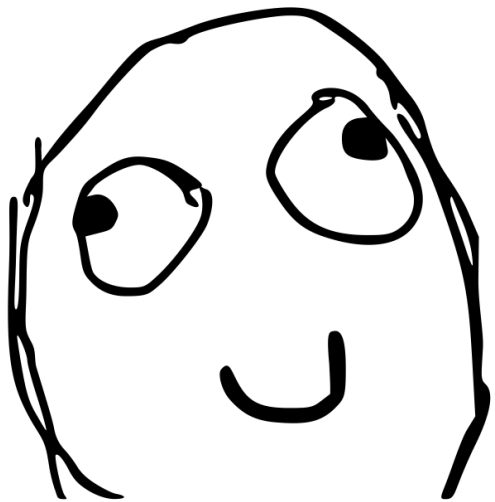


In progress & to do

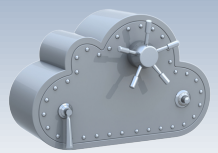
- **Convergence time** of dEAs in presence of crash-faults/cheaters
  - ↪ Main focus on **distributed cEAs and island models**
    - ↪ Influence of communication topology on the resilience against faults
  - ↪ From **empirical results** to **theoretical proofs**
  - ↪ What is the **impact of faults**?
  - ↪ **Till what threshold** we can ignore them?



Thank you!



UNIVERSITÉ DU LUXEMBOURG



Appendix

DG	Desktop Grids
DGVCS	Desktop Grids and Volunteer Computing Systems
ABFT	Algorithmic Based Fault Tolerance
cEA	cellular Evolutionary Algorithm
EA	Evolutionary Algorithm
UL	University of Luxembourg