Thesis subject:

Integrity issues on the cloud

. . . or not

Jakub Muszyński

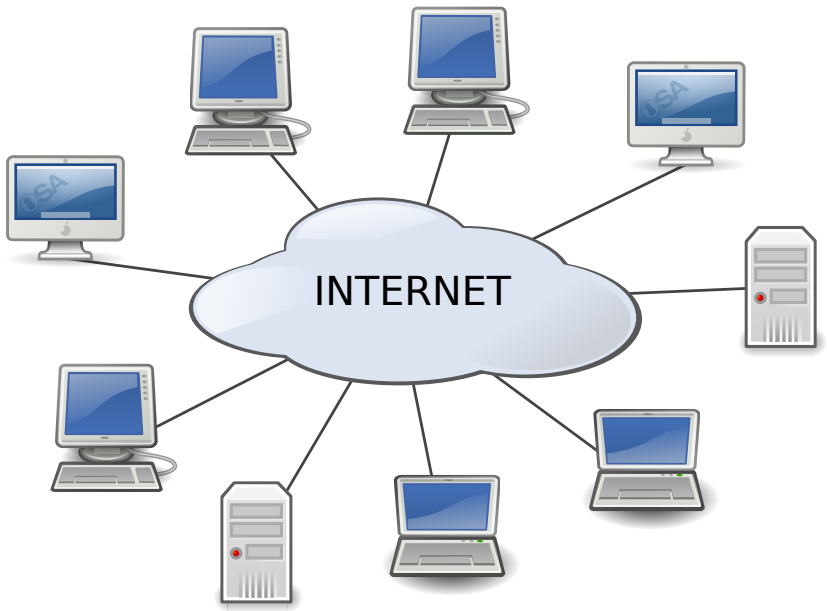
7th November 2014

Computer Science and Communications

(CSC) Research Unit

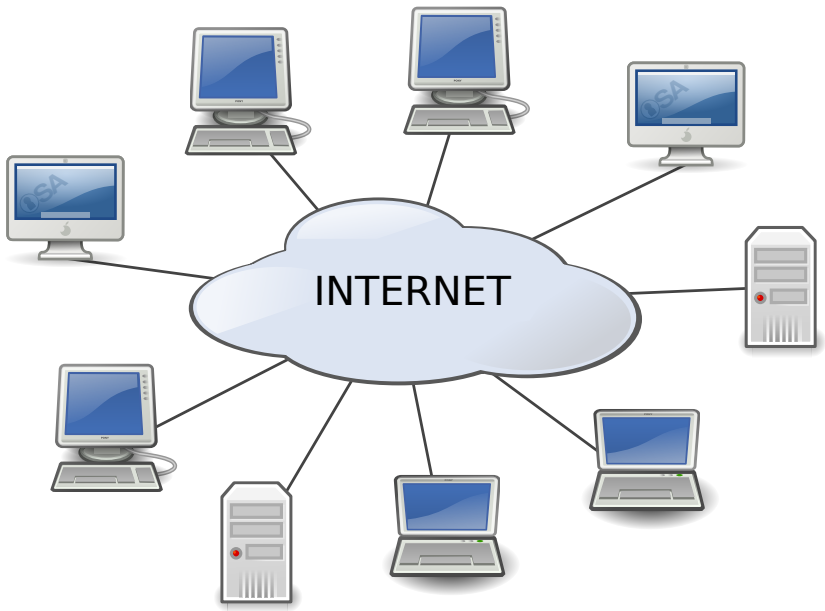
UNIVERSITÉ DU LUXEMBOURG

DGVCSs



- Desktop Grids and Volunteer Computing Systems:
  - ↪ Steal computing cycles from idle computers (the case $\simeq$ 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.*)

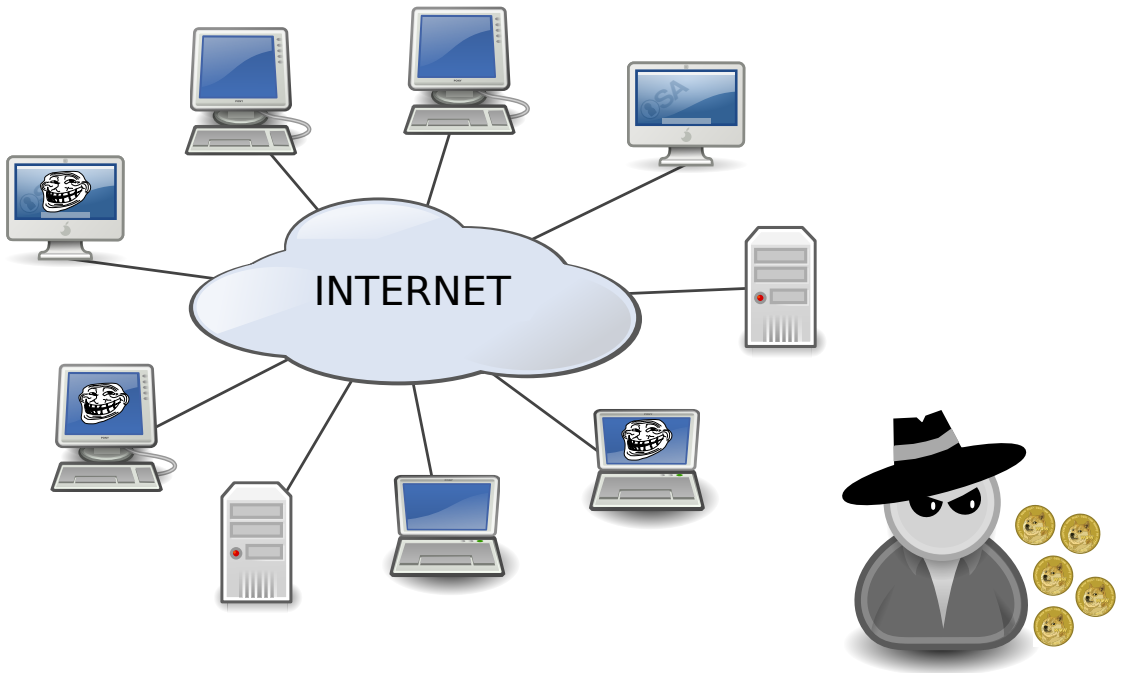
DGVCSs



- Desktop Grids and Volunteer Computing Systems:
  - ↪ Steal computing cycles from idle computers (the case $\simeq$ 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.*)

UNIVERSITÉ DU LUXEMBOURG

Cheaters



Rewards attract cheaters!

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

UNIVERSITÉ DU LUXEMBOURG

Cheating Faults

- 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 the worst impact** (create delay)
  - ↪ **Do not work more at T'** than at T

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 so far

- Parallel and distributed Evolutionary Algorithms
  - ↪ Theoretical proofs of convergence
  - ↪ Theoretical analysis of expected runtime
  - ↪ Some applications (*e.g.* hash functions generation)
- P2P gossiping protocols
  - ↪ Main interest: Newscast for robust dEAs executions
  - ↪ Simulations of Cheating Faults & the analysis of their impact
  - ↪ Cheating-tolerance

Done so far

- Parallel and distributed Evolutionary Algorithms
  - ↪ Theoretical proofs of convergence
  - ↪ Theoretical analysis of expected runtime
  - ↪ Some applications (*e.g.* hash functions generation)
- P2P gossiping protocols
  - ↪ Main interest: Newscast for robust dEAs executions
  - ↪ Simulations of Cheating Faults & the analysis of their impact
  - ↪ Cheating-tolerance

@

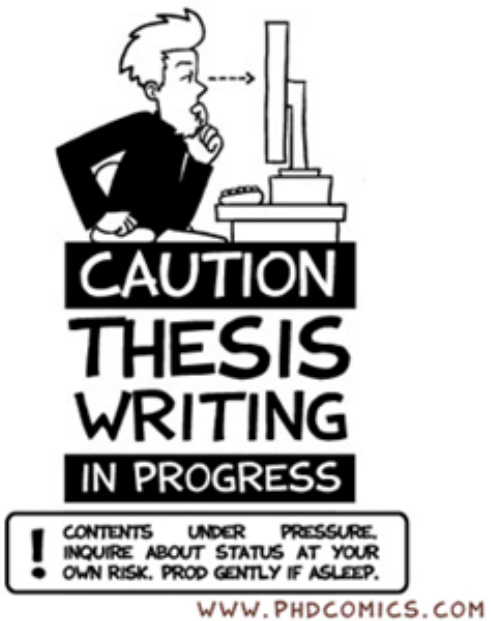
63 years of the CPU time

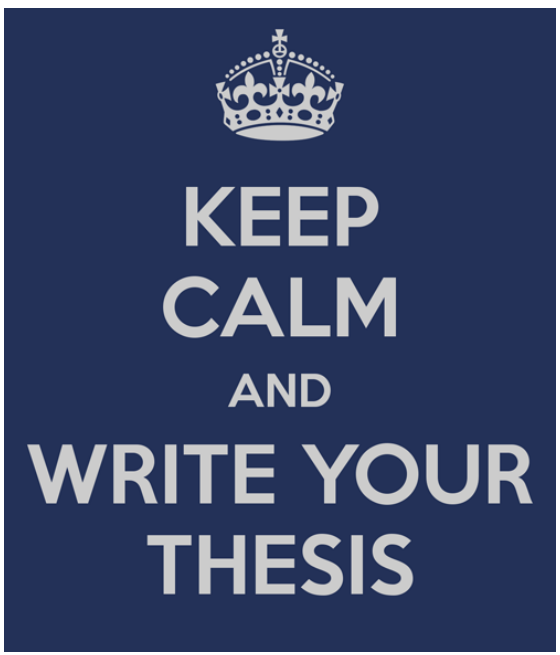
uni.lu
UNIVERSITÉ DU
LUXEMBOURG

Current

T.H.E.S.I.S.

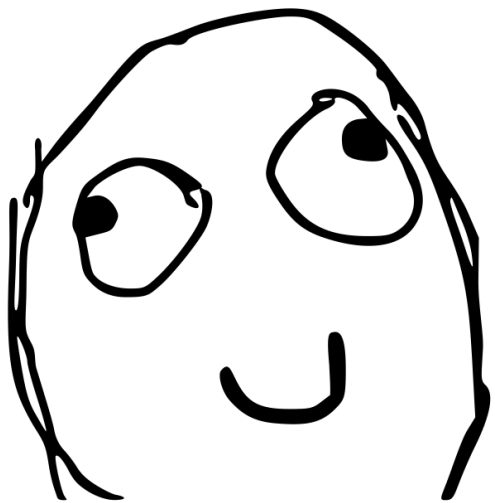
True Happiness Ended Since It Started ;-)





UNIVERSITÉ DU LUXEMBOURG

Thank you!



UNIVERSITÉ DU LUXEMBOURG

Appendix

ABFT	Algorithmic Based Fault Tolerance
dEA	distributed Evolutionary Algorithm
DG	Desktop Grids
DGVCS	Desktop Grids and Volunteer Computing Systems
EA	Evolutionary Algorithm
pEA	parallel Evolutionary Algorithm
UL	University of Luxembourg