The 7 deadly sins of cloud computing [2]
Cloud-scale resource management [1]

Michael Sevilla

University of California, Santa Cruz

May 20, 2013

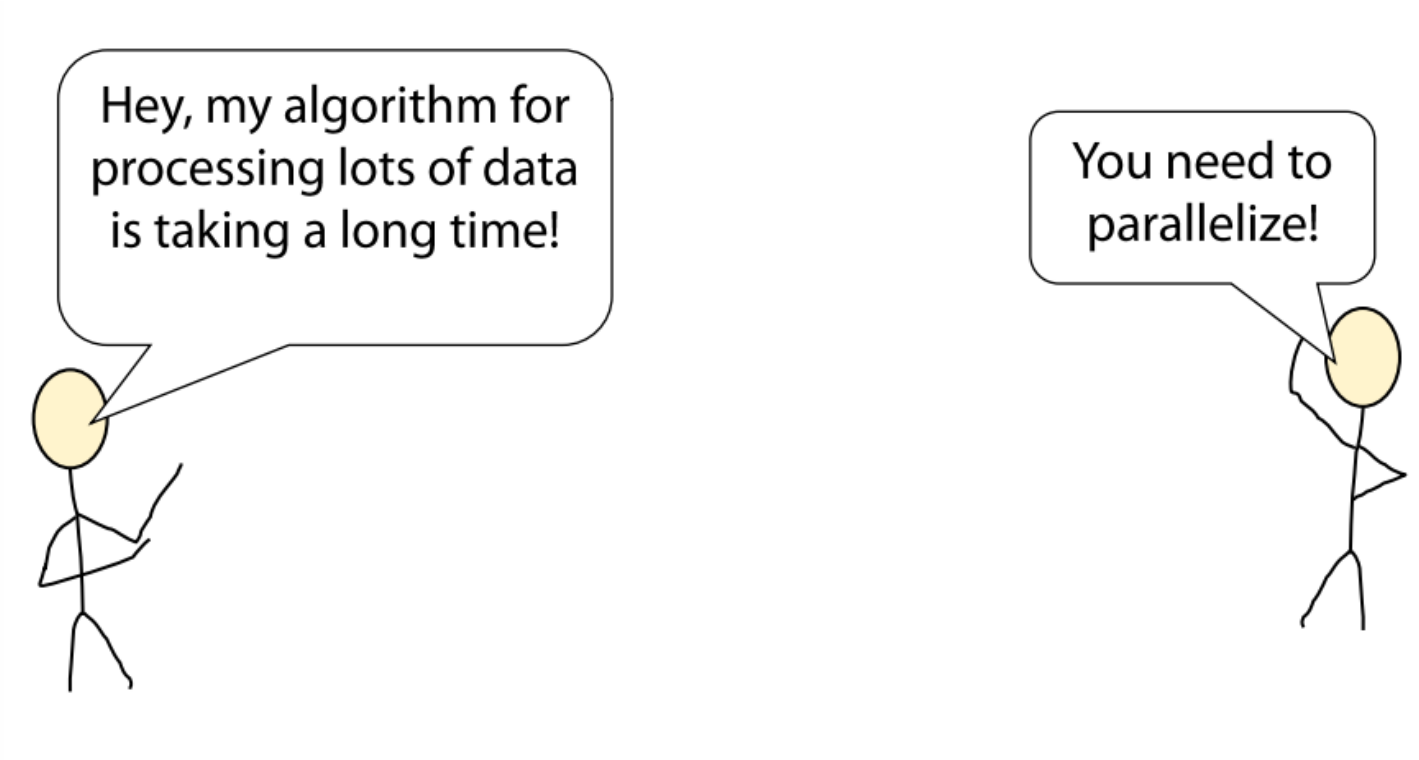
Deadly sins of cloud computing of research

sin (*n.*) - common simplification or shortcut employed by researchers; may present threat to scientific integrity and practical applicability of research

- Sin 1: unnecessary distributed parallelism
- Sin 2: assuming performance homogeneity
- Sin 3: picking low-hanging fruit
- Sin 4: forcing the abstraction
- Sin 5: unrepresentative workloads
- Sin 6: assuming perfect elasticity
- Sin 7: ignoring fault tolerance

Unnecessary distributed parallelism...

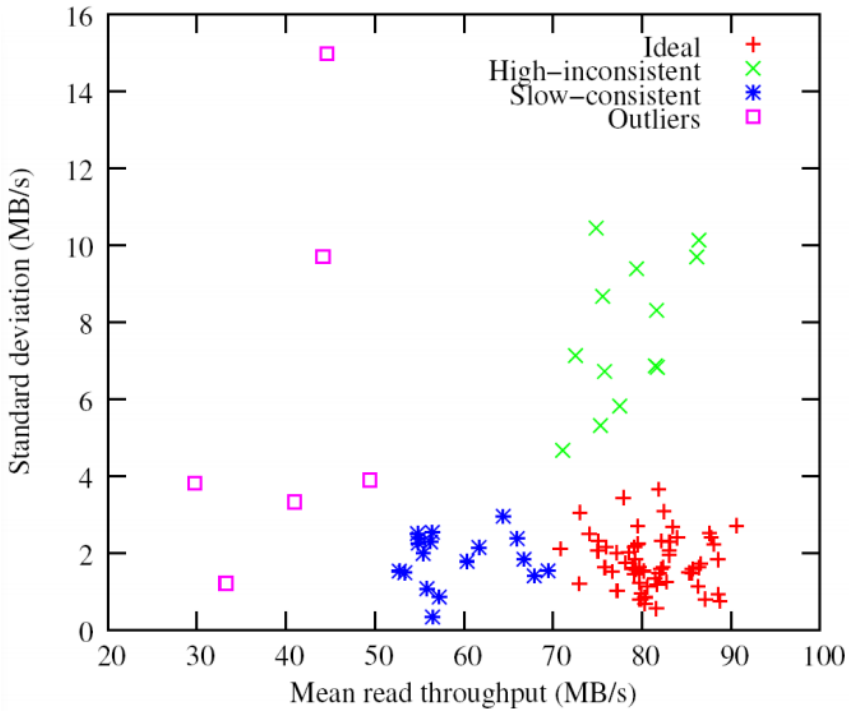
- ... is not free (overhead)
- ... has diminishing returns (scalability)



Suggestion: (1) compare to serial, (2) case-by-case decision

Assuming performance homogeneity...

- ... despite unknown environment (availability, load, hw)



Disk read performance

100 EC2 `m1.small` instances

Mean and standard deviation over 9 samples

Suggestion: rigor[1], repeatability[2], reproducibility[3]

[1] repeat experiments, error bars, quantify variance, explain outliers/results

[2] repeat at different times/hardware

[3] hardware config, communication network, datasets

Picking the low hanging fruit...

- ... is not compose-able or hard
- ... the ideas never change[4]

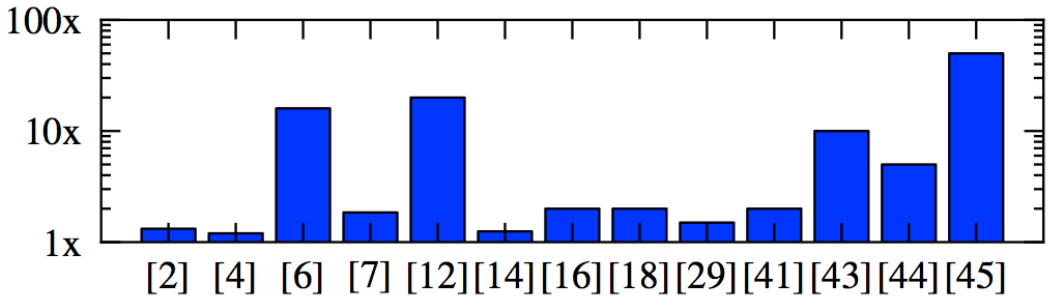


Figure 1: Maximum speedups over Hadoop claimed by a selection of research endeavours; N.B. log-scale y-axis.

Suggestion: (1) quantify costs (2) vs. "best-of-breed"

[4] (1) in-memory caching, (2) memoization, (3) exploit data locality, (4) domain-specific algs., (5) load vs. job runtime tradeoff

Forcing the abstraction...

- ... because not every job is designed for MR
- ... adds layers (Hive, Pig Latin, FlumeJava, etc.)

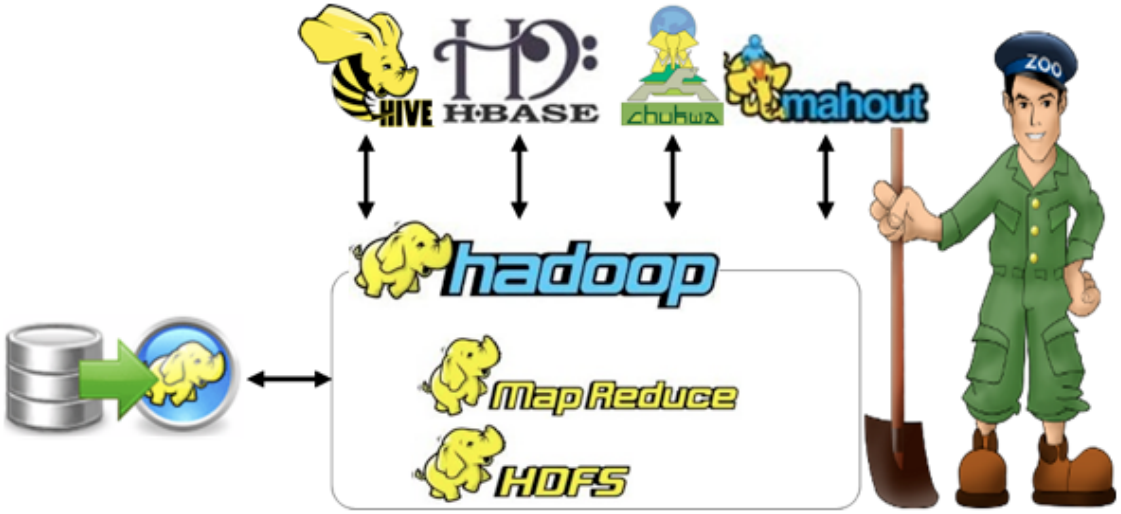


Image taken from http://dhuxxcuqauxw8.cloudfront.net

Suggestion: (1) domain-specific systems (2) > data

Unrepresentative workloads...

- ... are homogenous (same job/nature)

Image taken from http://veggieab.com/wp-content

Suggestion: new benchmarks (interference, types, spin-up)

Sins 6 and 7

Assuming perfect elasticity...

- ... expects infinite parallel speedup

Ignoring fault tolerance...

- ... can be a quantification for the speedup

Suggestion: argue for/against feature for the given workload

Conclusion

- Sin 1: unnecessary distributed parallelism
  → (1) compare to serial, (2) case-by-case decision
- Sin 2: assume performance homogeneity
  → rigor, repeatability, reproducibility
- Sin 3: picking low-hanging fruit
  → (1) quantify costs (2) vs. "best-of-breed"
- Sin 4: forcing the abstraction
  → (1) domain-specific systems (2) > data
- Sin 5: unrepresentative workloads
  → new benchmarks (interference, types, spin-up)
- Sin 6/7 perfect elasticity, ignoring fault tolerance

  → argue for/against feature for the given workload

"While we do not believe that this invalidates existing research, we see a danger of these sins becoming entrenched in the research agent and methodology."

- Schwarzkopf et al.

Cloud-scale Resource Management (i.e. VMs)

1. Resource model w/ "knobs" for
   - consolidation, guarantees, isolation
2. Challenges
   - scale, heterogeneity, compatibility, resources, storage
3. Solutions
   - {hierarchical, flat, statistical} scaling

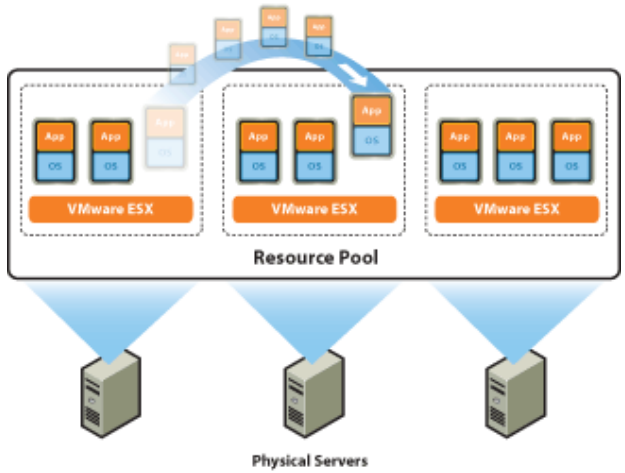


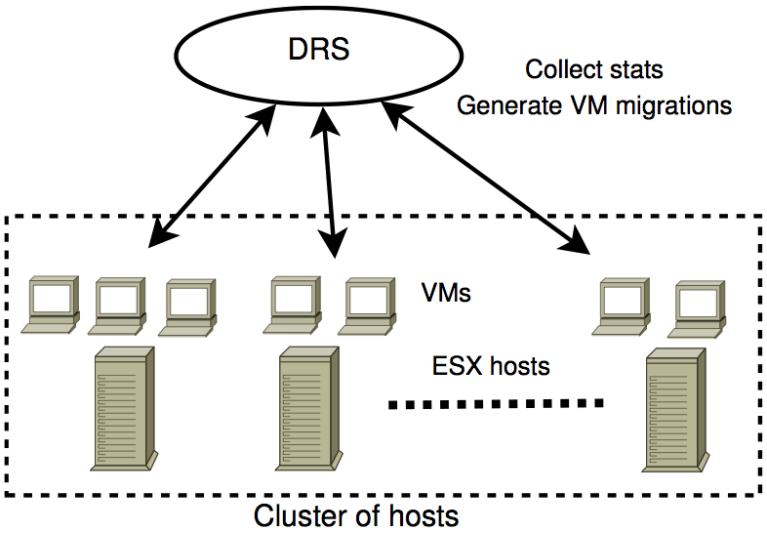
Image taken from http://ausweb.com.au/web-hosting/vmware-vcloud/

Goal: motivate further research in these areas

Resource models

E.g. VMWare's Distributed Resource Scheduler (DRS)

- high consolidation ratios
- service guarantees and utilization
- performance isolation



Knobs: express allocations of {CPU, memory, VM}

- reservation, limit, shares

Pools: express aggregate resources for VM groups

- isolation, sharing, hierarchies

Load balancing: initial {rsrcs, placement} + live migration

Resource management challenges

Scaling causes all kinds of problems!

- organization: collect inventory? manage users?
- heterogeneity: ++machines? attach storage? migrate? resource islands and load balancing? cluster-wide metrics? resource pool operations? processor speeds? compatibility between VM hosts?
- failures

Scaling techinques

hierarchical: layer resource management systems

? cluster metrics, load balance, manage resource pools

flat: distributed/decentralized resource management system

? compatability, >> failure models, consistency

statistical: smart operations at a small scale (repartition)

? manage resource pools

References I

Note: all images taken from the paper or the corresponding presentation (at HotCloud '11 and '13).

A. Gulati, G. Shanmuganathan, A. Holler, and I. Ahmad.

Cloud-scale resource management: challenges and techniques.
In *Proceedings of the 3rd USENIX conference on Hot topics in cloud computing*, HotCloud'11, pages 3–3, Berkeley, CA, USA, 2011. USENIX Association.

M. Schwarzkopf, D. G. Murray, and S. Hand.

The seven deadly sins of cloud computing research.
In *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing*, HotCloud'12, pages 1–1, Berkeley, CA, USA, 2012. USENIX Association.