XtreemOS

Enabling Linux for the Grid

Bag-of-Tasks Scheduling under Budget Constraints

Ana Oprescu, *Thilo Kielmann*

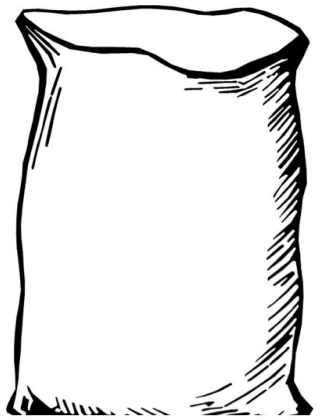
Vrije Universiteit, Amsterdam

kielmann@cs.vu.nl

Information Society Technologies

XtreemOS IP project
is funded by the European Commission under contract IST-FP6-033576

Bags of Tasks

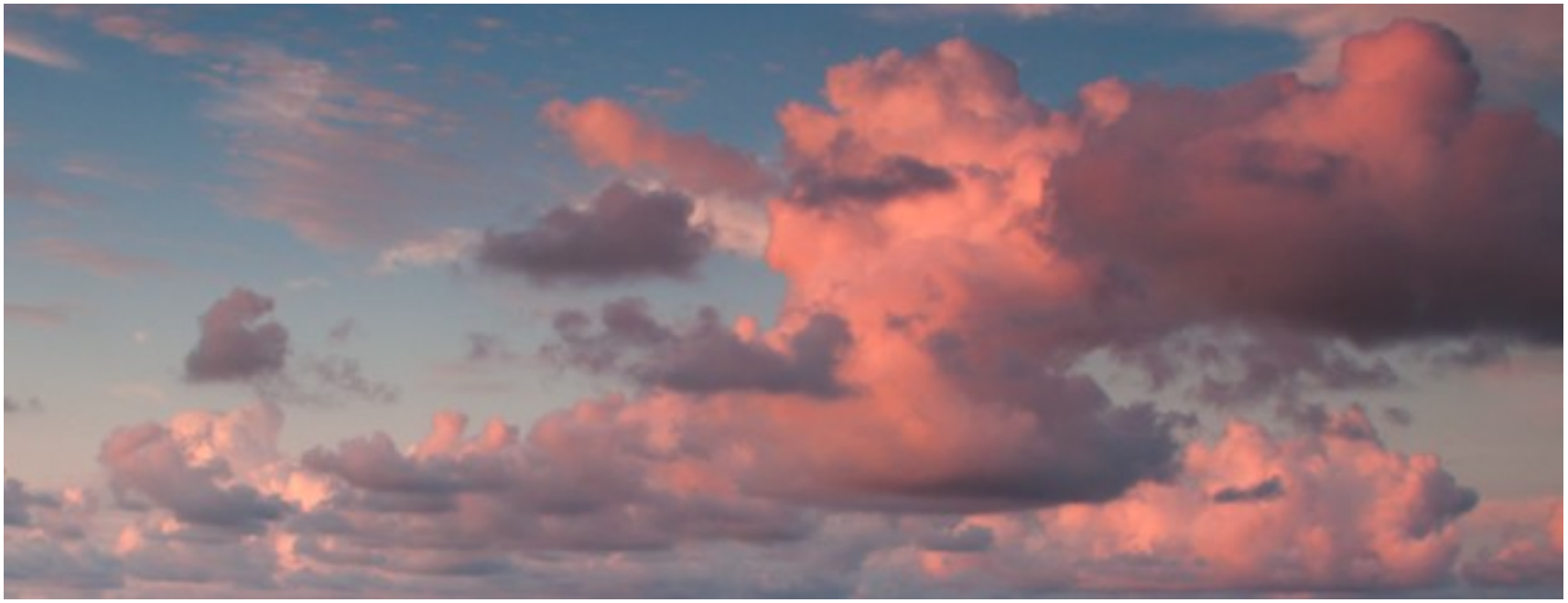


- **Parameter sweep applications**
- **High-throughput computing (Condor like**
- (OK, it's also a simple model to study...)

- **Execution model (traditionally)**
  - “Grab and run!”
  - Scientific users simply allocate all machines they can get hold of
  - Computations for free, best effort execution
  - Networks of workstations, clusters, grids,...

XtreemOS
Enabling Linux for the Grid

The promise of the cloud



- **Elastic computing, get exactly the machines you need, exactly when you need them...**
- **Well, did we mention you have to pay for the hour?**

XtreemOS
Enabling Linux for the Grid

“Quality of Service”

- **Small Instance, $0.085 per hour**
  - 1.7 GB of memory, 1 EC2 Compute Unit (ECU)
- **High-memory extra large, $0.50 per hour**
  - 17.1 GB memory, 6.5 ECU
- **High CPU medium, $0.17 per hour**
  - 1.7 GB of memory, 5 EC2 Compute Units

Which one is faster for *my* application???

Which one is cost efficient???

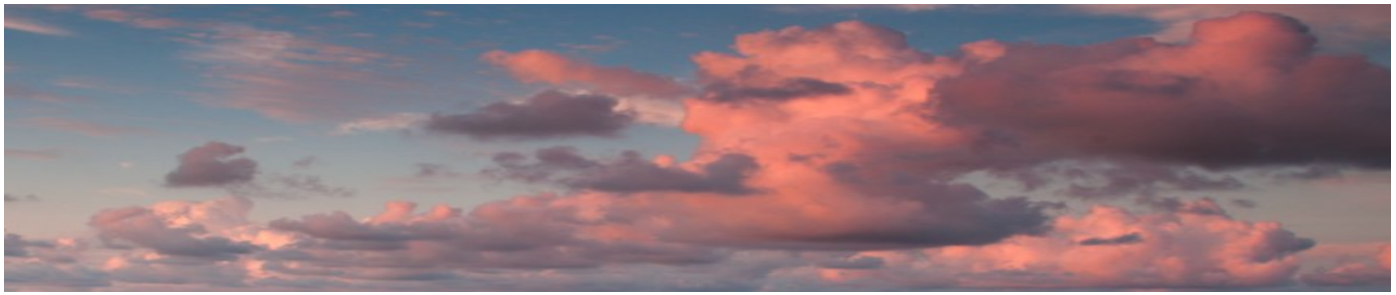
XtreemOS
Enabling Linux for the Grid

What's in a bag?

- **Many *independent* tasks**
  - Let's focus on the budget here...
- **Runtimes are unknown to the user**
- **Tasks have *some* runtime distribution, but we don't know it either**
- **Tasks can be aborted / restarted if needed**

XtreemOS
Enabling Linux for the Grid

What's in a cloud?



- **A cloud offering provides machines of certain properties like CPU speed and memory**
  - All machines in a cloud offering are homogeneous
  - There is an upper limit of machines per cloud that a user can get
- **A machine is charged per Accountable Time Unit (ATU); 1 hour, for example**
- **We call a cloud offering (machine type, price, max. number) a *cluster***
  - We are HPC guys, after all...

XtreemOS
Enabling Linux for the Grid

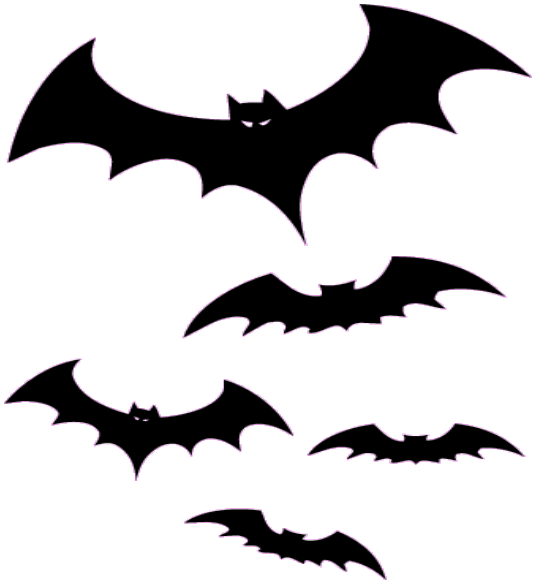
What's the problem?

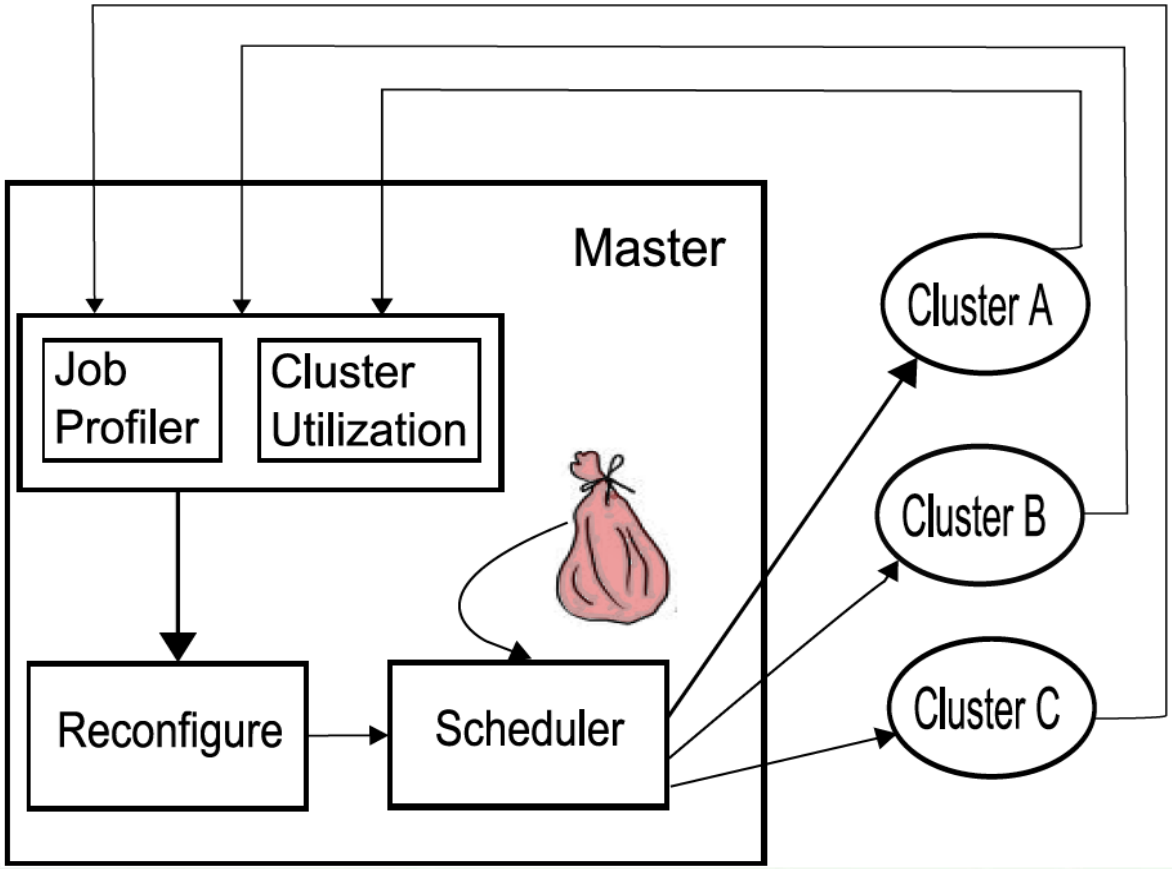
- **We are on a budget.**
- **We know nothing.**

- **We want to**
  - Run all tasks from our bag on (cloud) clusters, without spending more than our budget
  - Allocate/release machines dynamically while learning how fast our tasks execute on the different clusters
  - If we learn that our budget is too low, give up
  - Minimize makespan of the whole bag, if we can make it within budget

BaTS

- **Budget-constrained task scheduler**

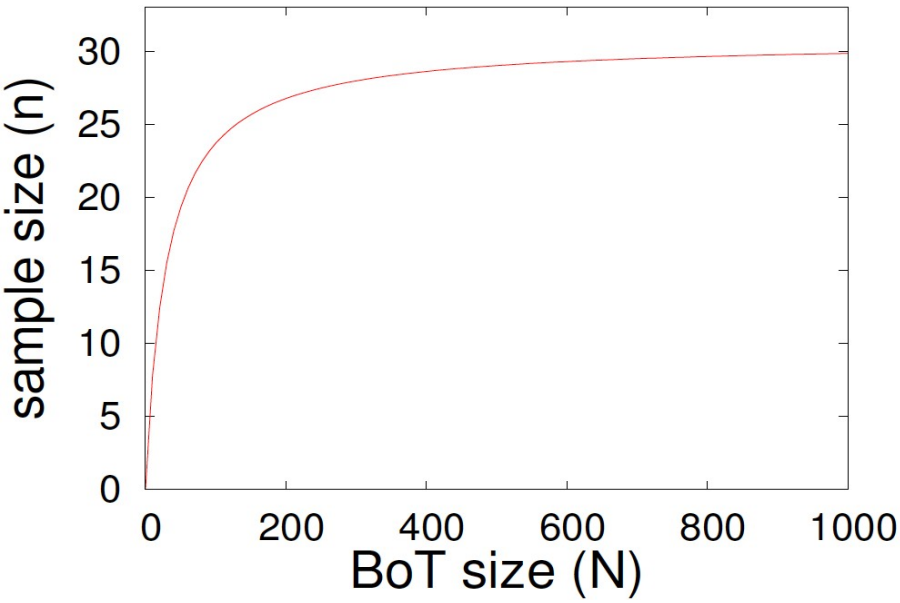




Job Profiler

Estimating task runtimes, for each cluster:

- Keep a moving average, update during execution
- Initialize the average using a small, initial sample
  - Statistics for sampling with replacement
- For the initial sample, keep an ordered list of runtimes



Disclaimer:
This is going to be statistics for ~~dummies~~ engineers...

Job Profiler (2)

For each cluster:

- **Start with a set of initial workers**
- **Run the initial sample**

At regular monitoring intervals:

- **Reconfigure based on estimates**
  - Remaining problem (less tasks and money left)
  - For updating the moving average, running tasks are estimated by the average of the “tail” from the current runtime to the end of the distribution of the sample set
- **Run more tasks**

Cluster Configuration

- **From the average speed of each cluster, (in tasks per minute) we can compute estimates for makespan (Te) and cost (Be) for a configuration from nodes of multiple clusters:**

$$T_e = \frac{N}{\sum_{i=1}^{C_{nc}} \frac{a_i}{T_i}} \quad ; \quad B_e = \left\lceil \frac{T_e}{ATU} \right\rceil * \sum_{i=1}^{C_{nc}} a_i * c_i$$

- **We minimize Te while keeping Be <= B using a modified Bounded Knapsack Problem (BKP)**
  - The BKP can be solved in pseudo-polynomial time, as 0-1 knapsack problem via linear programming
- **BaTS chooses the configuration with minimal Te for Be <= B**

XtreemOS
Enabling Linux for the Grid

Cluster Monitoring

BaTS regularly re-evaluates the current cluster configuration:

- **The moving averages converge during the run**
- **Execution on real machines adds some complexity:**
  - Individually requested from the cloud provider, startup time until ready
  - Each machine has its own end of the next ATU
  - Tasks have runtime granularity, may leave machine time unused
- **For each reconfiguration, BaTS keeps track of**
  - Time on machines we already paid for
  - Actual speed (tasks/minute) achieved per cluster

XtreemOS
Enabling Linux for the Grid

Let's try it out

- **DAS-3 multi-cluster system**
- **Emulate 2 clusters (clouds) of 32 machines each**
- **Machine allocation by job submission via SGE**
  - (without competing users)
- **Bag of 1000 tasks with predefined runtimes**
  - Normal distribution mean = 15min, stddev = 2.27 min
  - [Iosup et al., HPDC 2008] show that bags typically have some normal dstribution
- **Task "execution" by sleep(runtime)**
- **Fast/slow machines emulated by linearly modifying the sleep time**
- **Compare BaTS to a round-robin scheduler (RR), always using 32+32 machines**

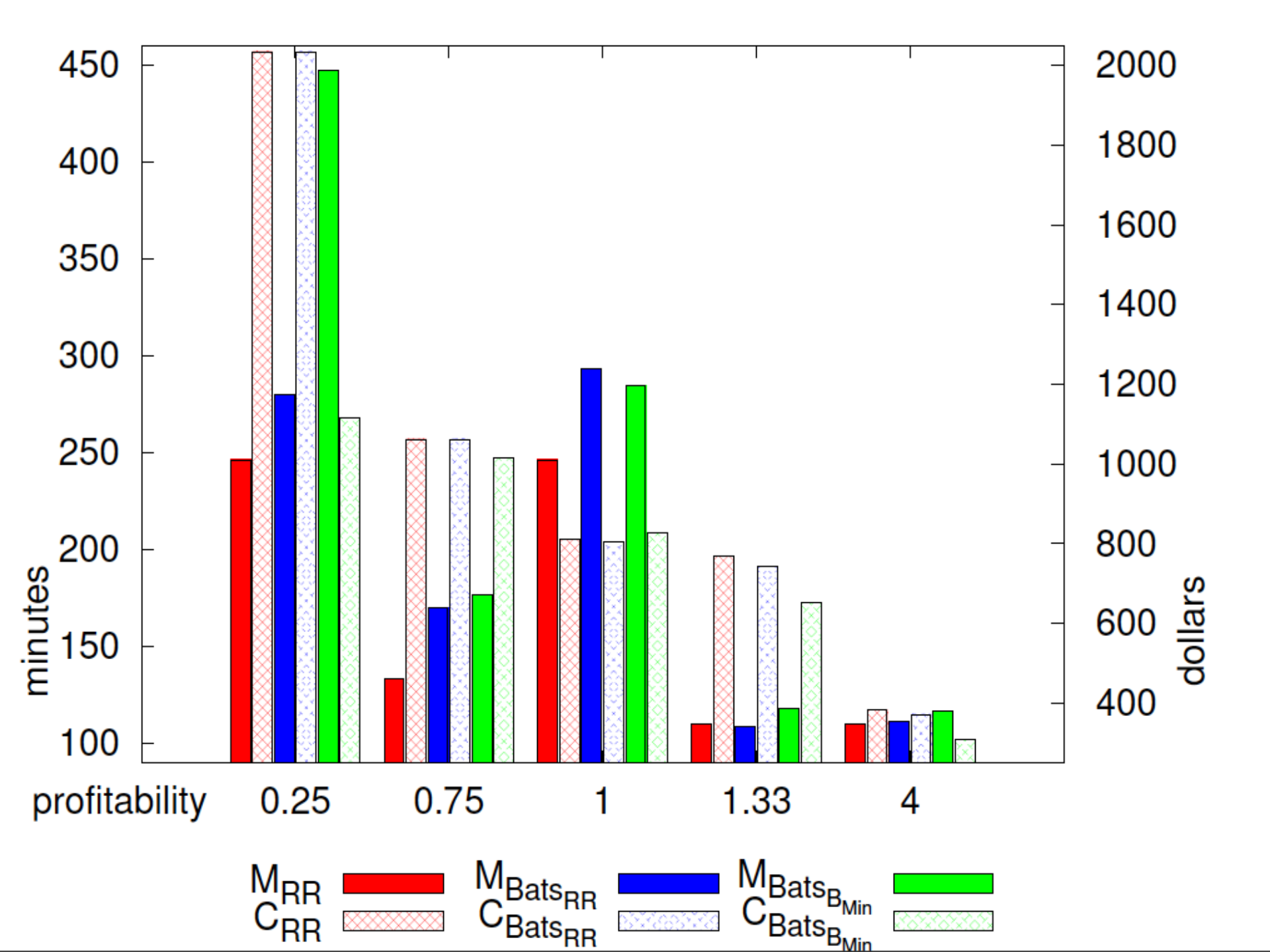
Profitability (experiment setup)

Cluster 1 running with normalized speed and cost

Cluster 2 has varying speed and/or cost

Design space for BaTS is in the profitability of cluster 2 w.r.t. Cluster 1

	cluster 1		cluster 2		average	
profitability	speed	cost	speed	cost	speed	cost
0.25	1	1	1	4	1	2.5
0.75	1	1	3	4	2	2.5
1	1	1	1	1	1	1
1.33	1	1	4	3	2.5	2
4	1	1	4	1	2.5	1


450
400
350
300
250
200
150
100
minutes
2000
1800
1600
1400
1200
1000
800
600
400
dollars
profitability
0.25
0.75
1
1.33
4
M_{RR}
C_{RR}
$M_{Bats_{RR}}$
$C_{Bats_{RR}}$
$M_{Bats_{B_{Min}}}$
$C_{Bats_{B_{Min}}}$

XtreemOS
Enabling Linux for the Grid

Conclusions

- **Choosing the right cloud offering is tough**
- **BaTS can help staying within budget while still performing reasonably well**
- **Guessing a proper budget up front is our current challenge**
  - Work in progress: pre sampling (even smaller)
- **Less low hanging fruit:**
  - DAG's instead of BoT's (dependencies)
  - BaTS for MapReduce?


Questions?