Scaling a Virus lab

Robert Sandilands, Director Anti-virus <roberts@commtouch.com>

Overview

- Why do we need to scale?
- Scaling?
- What are the choices available?
- Condor basics
- How we made Condor work for us
- Conclusions

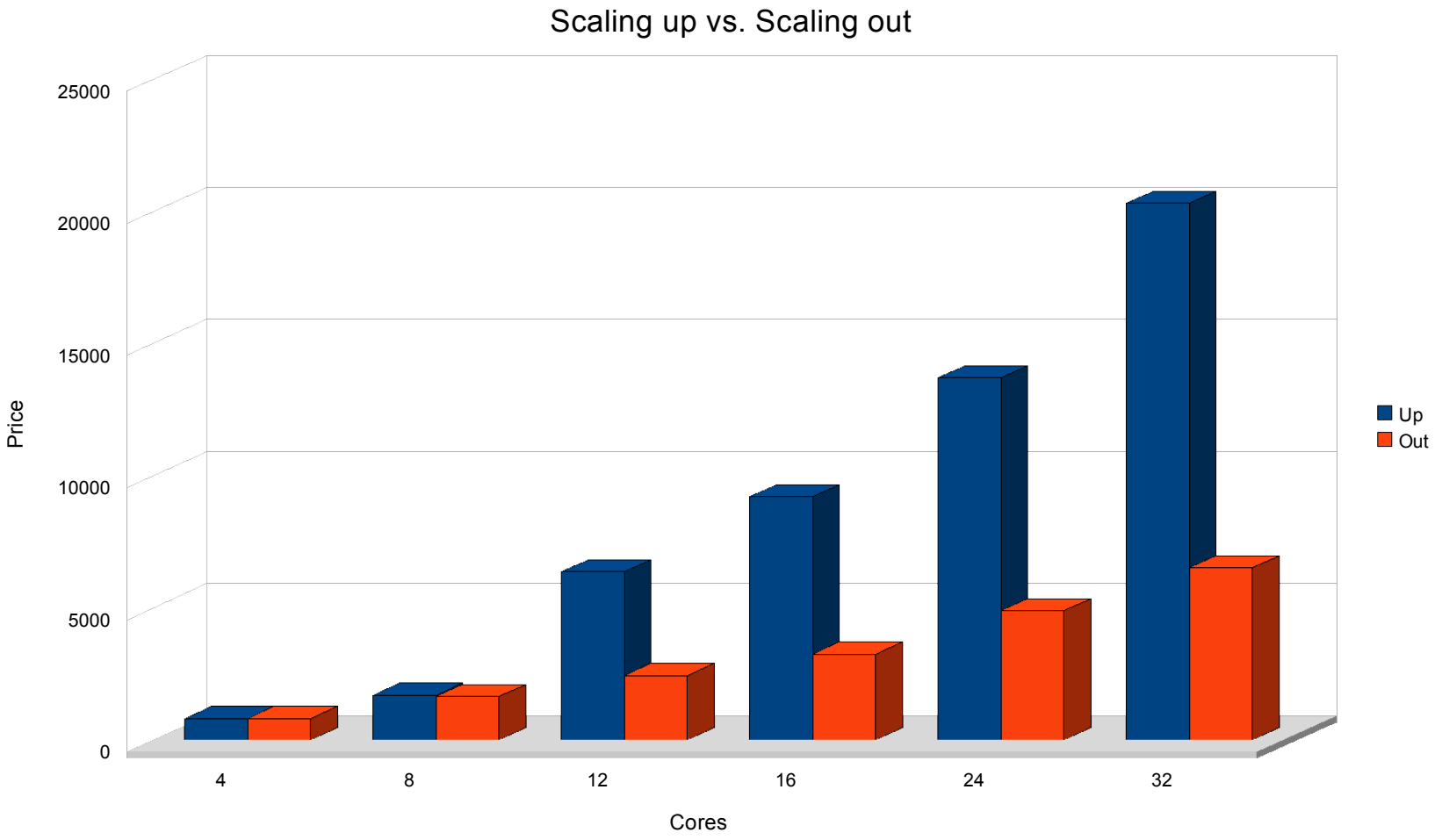
What is scaling?

- Add more resources to be able to
  - Solve a problem faster
  - Process more data
- Resources can be people and/or computers
- You have to process samples faster than you receive them otherwise you have the dreaded backlog
- Two types of scaling
  - Up – faster computers or people
  - Out – more computers or people

Why do we need to scale?

- Every month we get more samples than we did the month before
- The total size of samples are growing
- The number of samples any analyst has to deal with is increasing
- We need to build smarter detection
- People should be efficient
  - Do stuff they are good at
  - Hard to get people

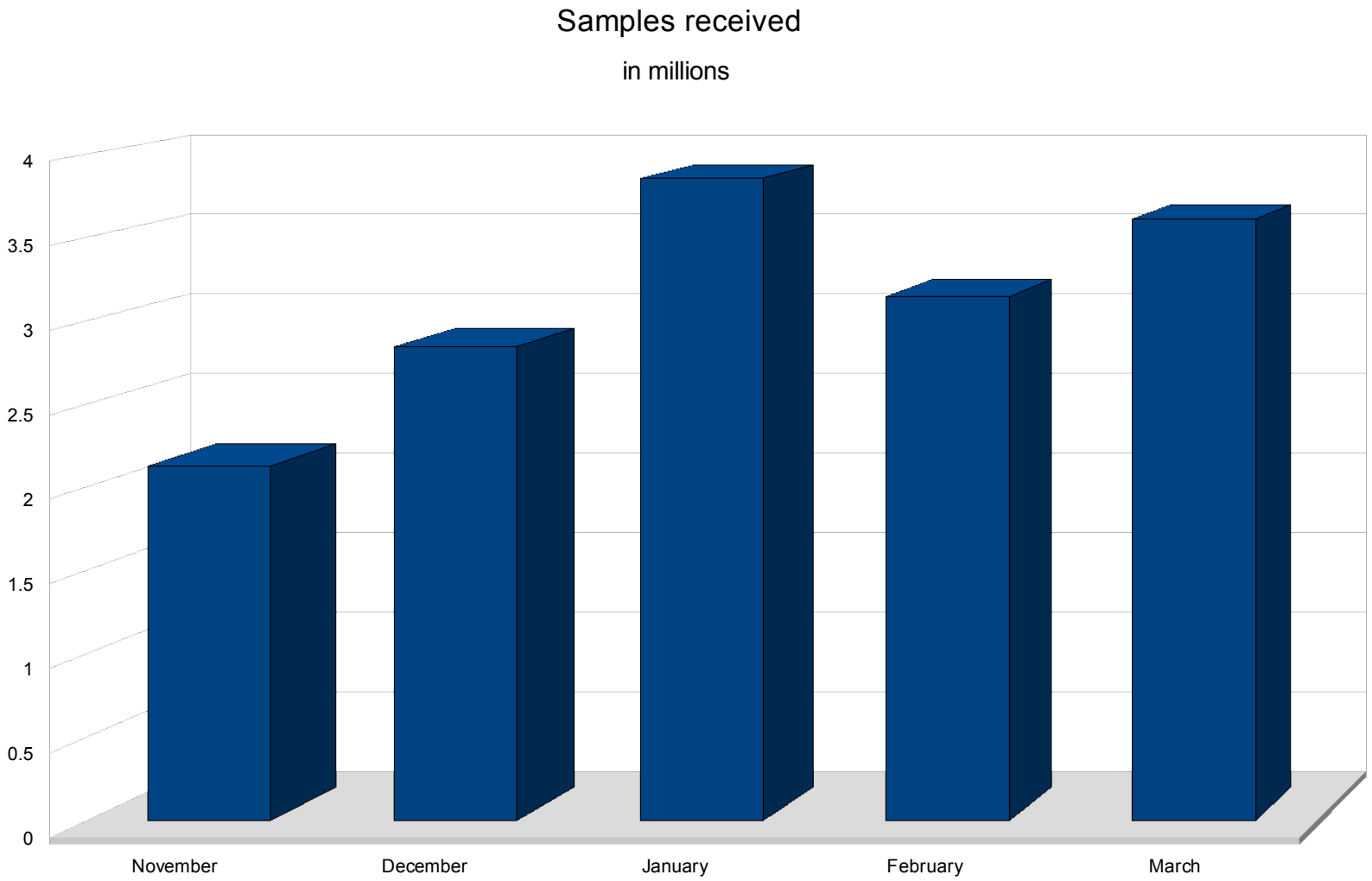
Cost of scaling up vs. scaling out



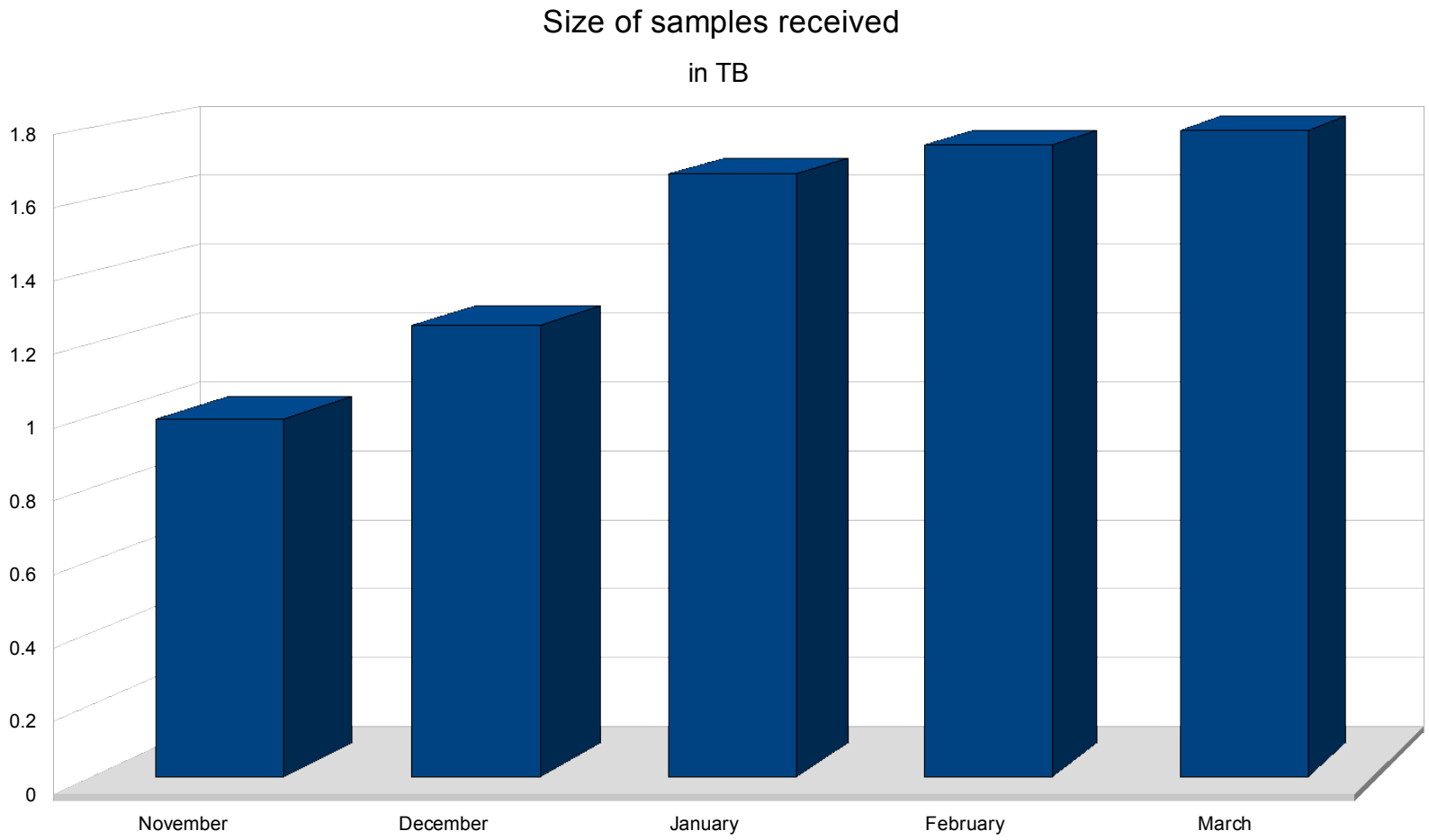
Why do we need to build smarter detection?

- The major source of samples is not new variants generated by malware authors but polymorphism
- We can detect all known samples by hash, but
  - Inefficient
  - Large databases
  - Does not really help customers
- Proactive detection
- Smart signatures is not a silver bullet, it is just a valuable weapon against this type of problem

Samples Received



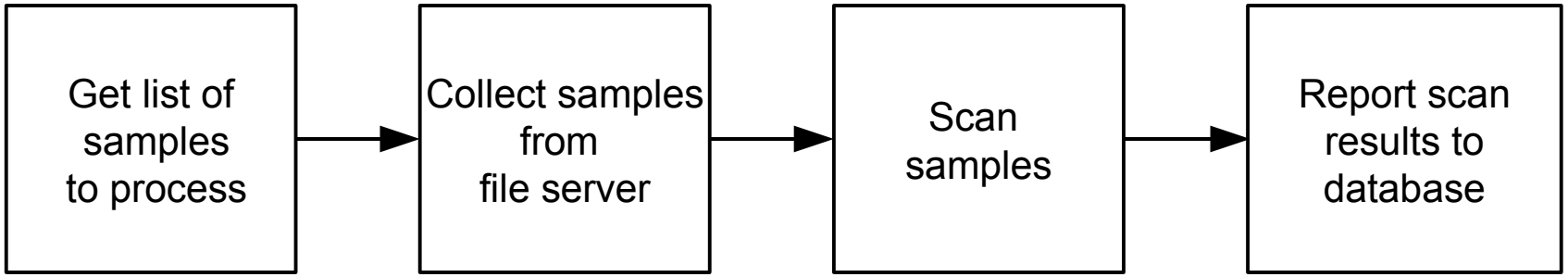
Total size of samples received



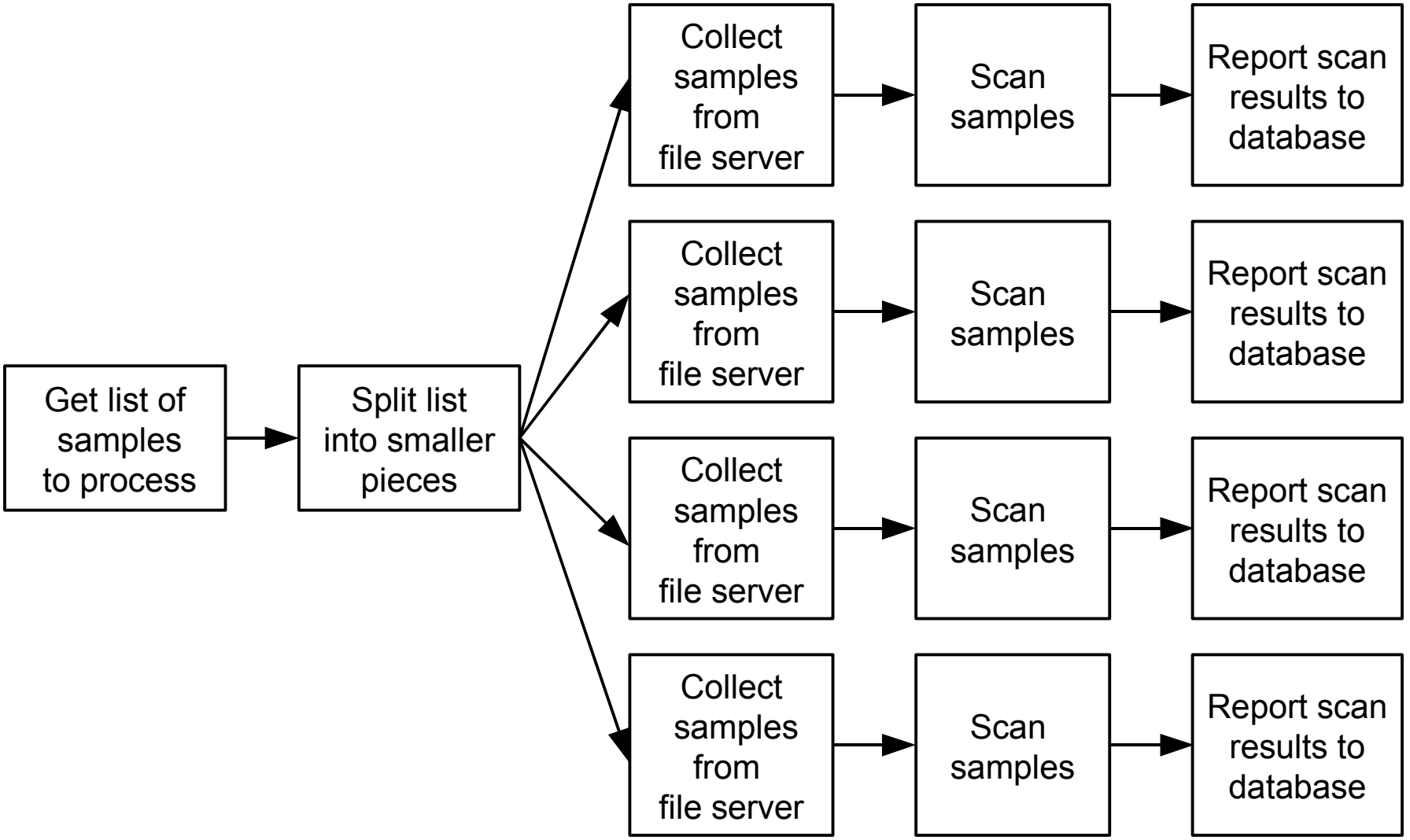
Scaling up or scaling out?

- Scaling up
  - More memory, more disks, more cores, faster cores
    - Expensive and inefficient
    - Sometimes the only choice
- Scaling out
  - Clusters and grids
    - More machines
    - Each machine relatively inexpensive
    - More management
    - More cost efficient

Scaling up


Get list of samples to process
Collect samples from file server
Scan samples
Report scan results to database

Scaling out


Get list of samples to process
Split list into smaller pieces
Collect samples from file server
Scan samples
Report scan results to database
Collect samples from file server
Scan samples
Report scan results to database
Collect samples from file server
Scan samples
Report scan results to database
Collect samples from file server
Scan samples
Report scan results to database

What are the choices available?

- Software
  - MPI
  - PVM
  - OpenMP
  - CUDA/OpenCL
  - Distributed Ruby and other specialist languages
  - Map Reduce (Hadoop)
- Or job management

The embarrassing side

- Amdahl's law
  - Efficiency
  - Some parts serial, some parallel
  - Limits, but good base
- Text sorting
  - Quick sort not easy to run distributed
  - Merge sort better, but not perfect
- Malware Scanning
  - Embarrassing

Job management

- Job management
  - Break tasks up into small bits
  - Distribute tasks
- No silver bullets
  - Sometimes you have to use the Hadoop or MPI type technologies

Condor Basics

- URL: http://www.cs.wisc.edu/condor/
- 23 year old technology
- Flexible, job management
- Can integrate with Hadoop, MPI, Boinc, Globus
- Multiple OS support including Windows
- Job management
- Slots

Dagman Basics

- "DAGMan (Directed Acyclic Graph Manager) is a meta-scheduler for Condor. It manages dependencies between jobs at a higher level than the Condor Scheduler"
- Job inter-dependencies
  - Some jobs will only execute after previous job(s) succeeded
- Great match for a virus lab

Condor
High Throughput Computing

How we made it work for us: Example 1

- Mass adding
  - Evil, but unfortunately needed
  - Simplistic detection
  - Serial:
    - 1 week for 10,000 samples
  - Parallel:
    - 4 hours for 30,000 samples

Mass adding

- It used to be a serial process
  - Filtering followed by signature generation
  - Re-filter followed by signature validation
  - Took about a week requiring that we re-filter
- In parallel
  - So much faster that we can exclude the re-filtering
  - The CPU intensive processes run simultaneously on up to 20 slots.
  - Some tasks have to run in serial

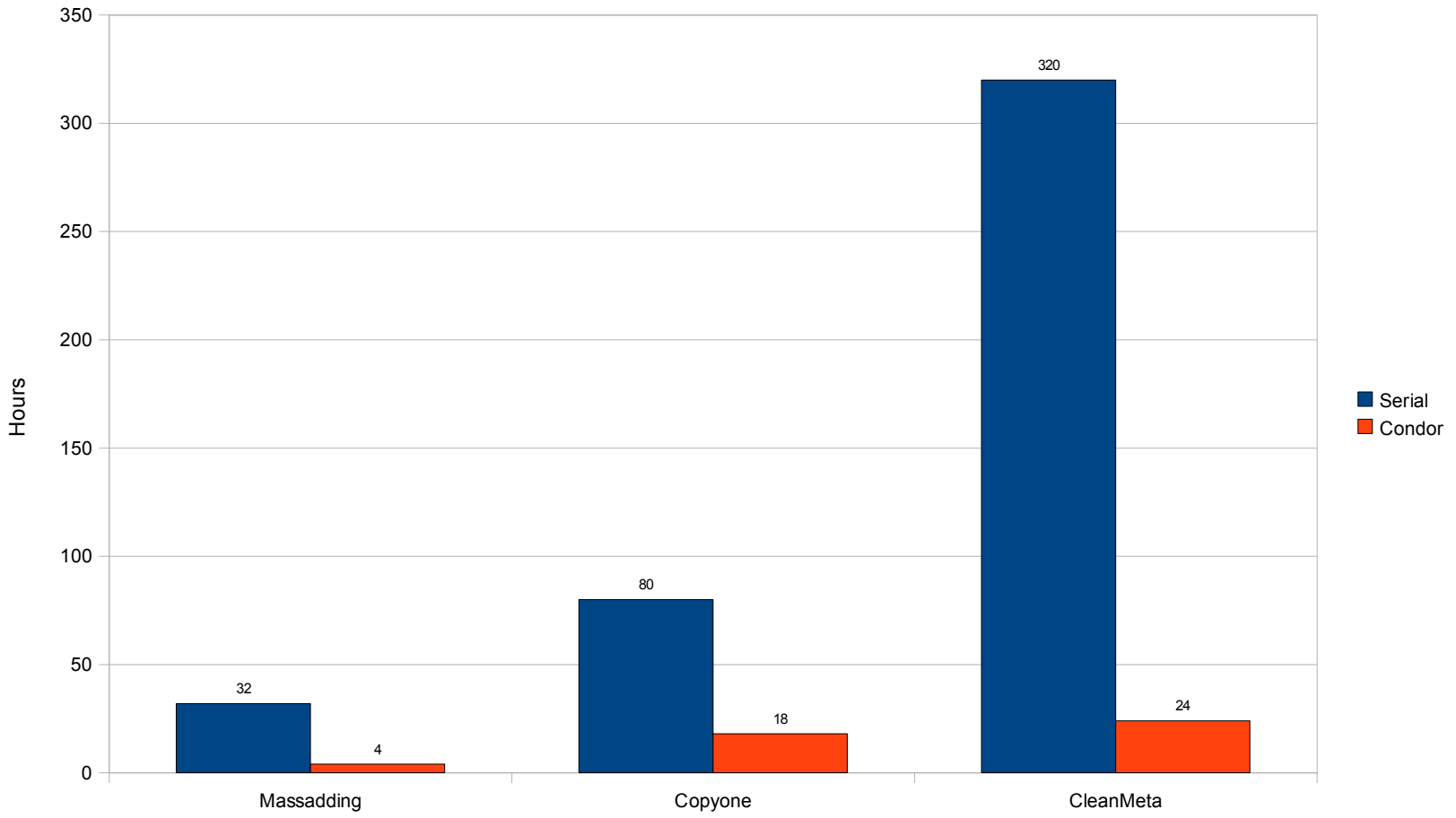
Extracting/processing metadata: Example 2

- Static metadata: Size, hash, date received
- Dynamic metadata:
  - Time or version dependent
  - The simplest metadata is detection information.
    - Not only for your own scanner but for other scanners
  - Not all tools play nice
  - Mostly Windows based 3rd party tools
- Example: copyone process went from 2 weeks to 18 hours using 90 Condor slots

How we made it work for us: Example 3

- Extracting/processing metadata
  - Clean files
  - Time sensitive
  - Helps with generic detection
  - Serial:
    - Several weeks for a few million clean files
  - Parallel:
    - 2 to 3 days for the same number of files
  - Parallel and optimizing the SQL
    - 1 day for the same number of files

How much faster?



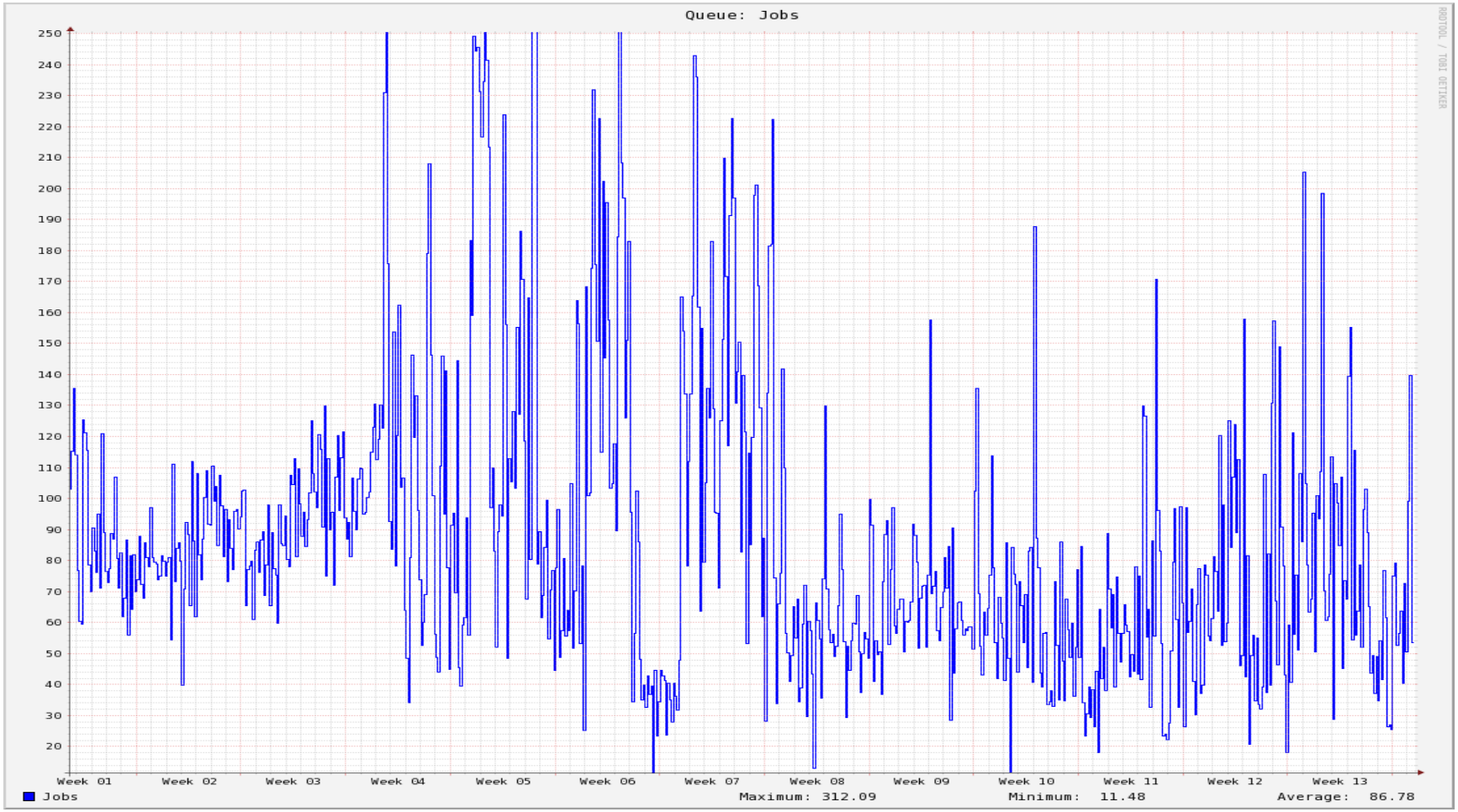
Example: Condor job

- executable = aiscan.exe
- universe = vanilla
- should_transfer_files=yes
- transfer_input_files=aivsecon.def, antivir.def
- RunAsOwner=True
- Requirements = (Arch == "INTEL" && OpSys == "WINNT51" || OpSys == "WINNT52")
- arguments= --list \\fileserver\virus\2010-02\001
- queue
- arguments= --list \\fileserver\virus\2010-02\002
- queue
- arguments= --list \\fileserver\virus\2010-02\003
- queue

Example: Dagman job

- JOB prepare prepare.sub DIR jobs
- JOB scan1 scan.sub DIR jobs
- JOB scan2 scan.sub DIR jobs
- JOB final final.sub DIR jobs
- VARS scan1 job="1"
- VARS scan2 job=”2”
- VARS prepare jobs="2"
- VARS final jobs=”2”
- PARENT prepare CHILD scan1 scan2
- PARENT scan1 scan2 CHILD final

Usage of our grid



Conclusions

- Several examples where tasks were optimized from weeks to less than one day.
- Tasks have to be embarrassingly parallel
- Most tasks in a virus lab are embarrassingly parallel
- You are capable of processing significantly more data so your databases, file servers and network infrastructure will be pressured by significantly more work load.
- In the end our analysts have more timely information and spend less time waiting for it

Scaling a Virus lab

Robert Sandilands, Director Anti-virus <roberts@commtouch.com>