Cloud & Control

Any Program on 2000 or 2 Machines

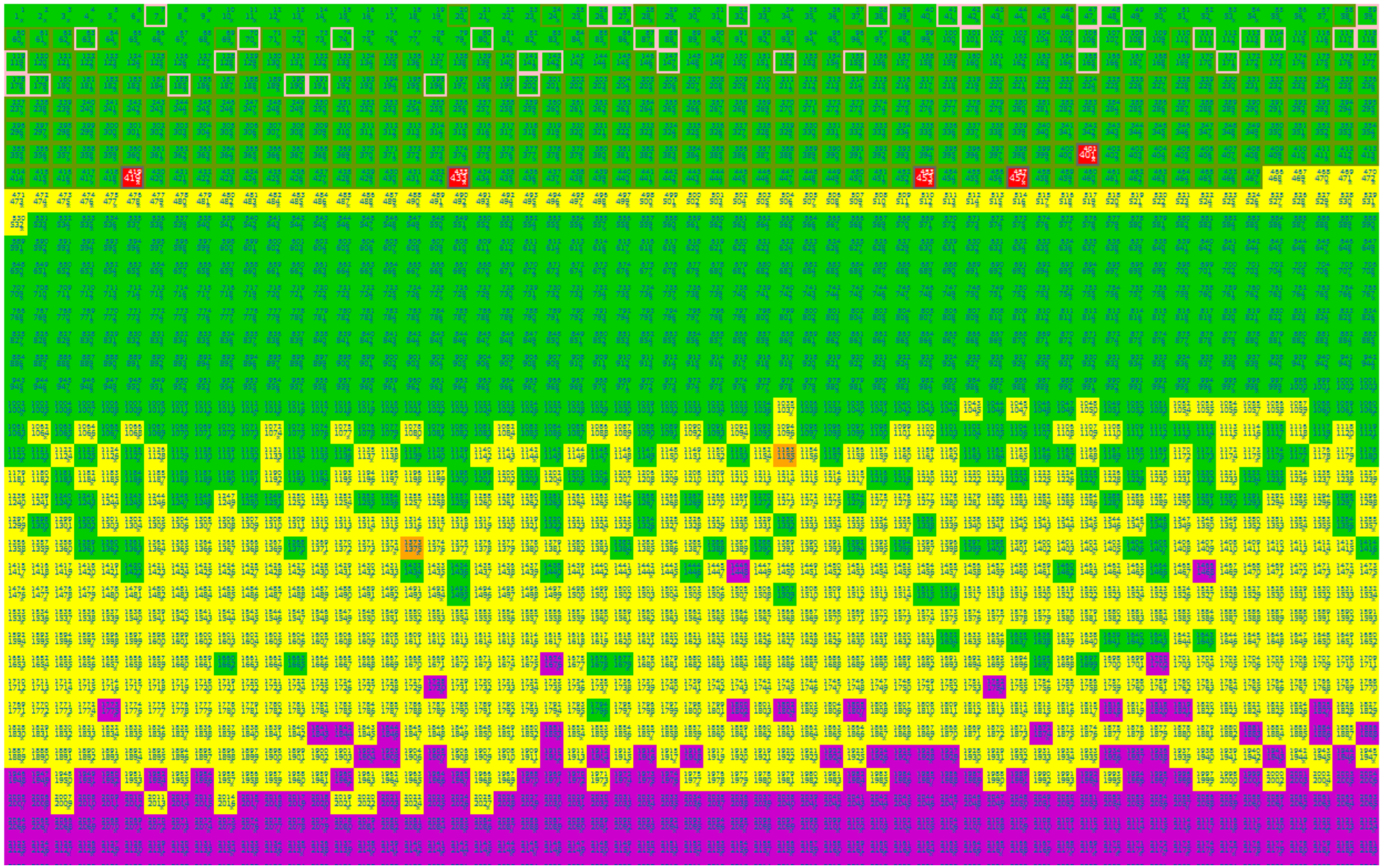
Session ID: HT2-203

Session Classification: General Interest

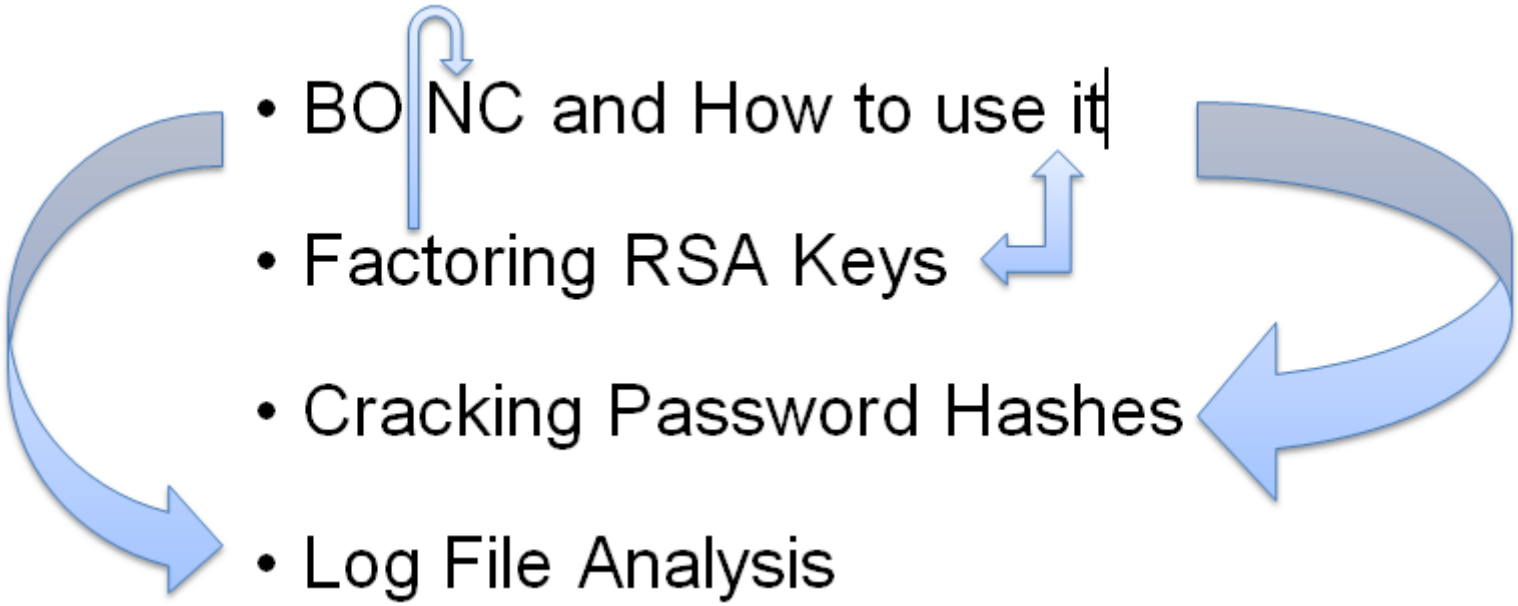
RSA CONFERENCE 2012

Workunit Status

Success | Working | Unsent | Canceled | Admin-Aborted | Error | Resource Limit | Download Error | Detached | No Reply | Unknown



- BOINC and How to use it
- Factoring RSA Keys
- Cracking Password Hashes
- Log File Analysis

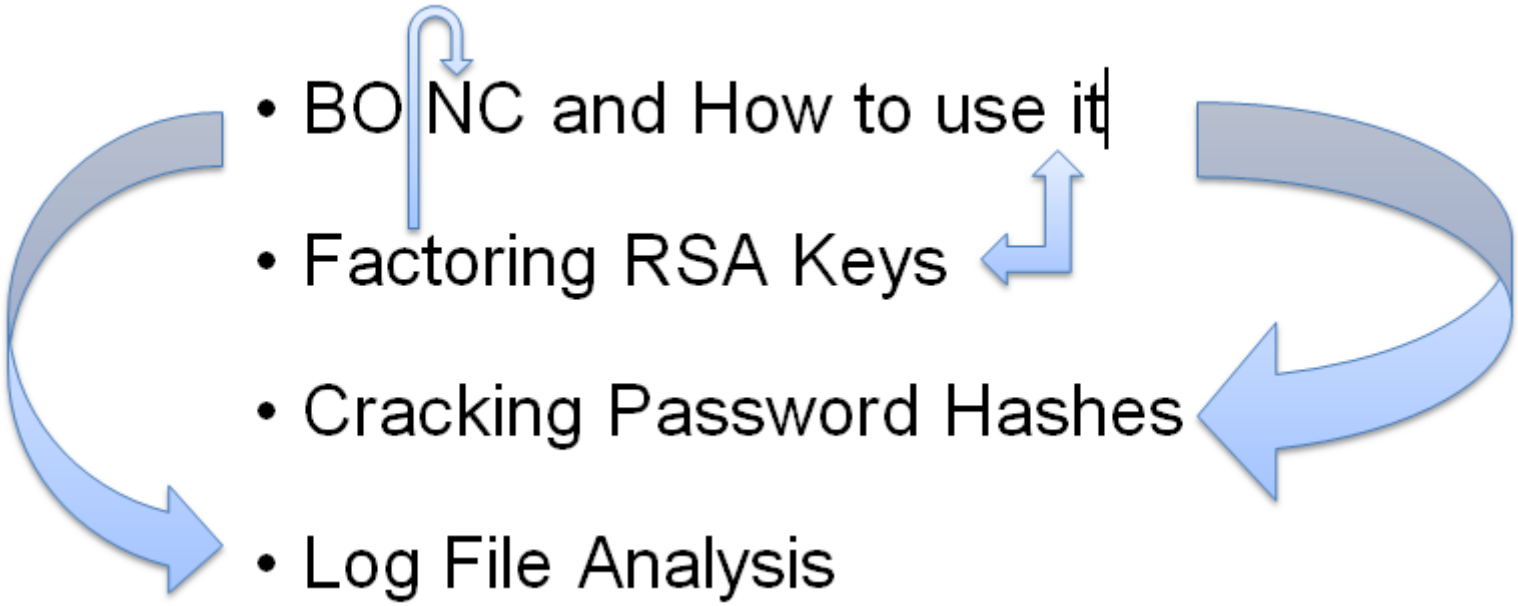


You have interesting problems!

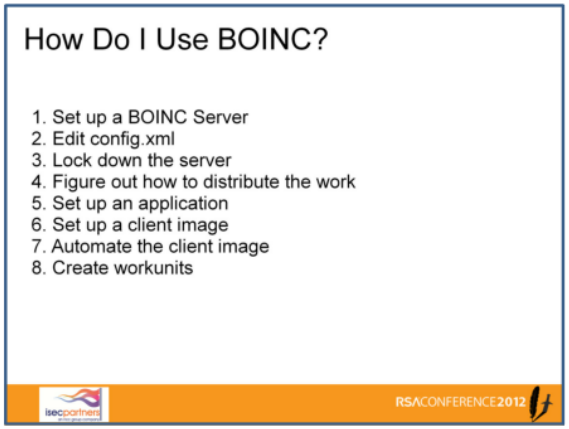
- Fuzzing
- Document Analysis
- SMT Solving

Would BOINC Help?

How would you fit your problem into BOINC?



Materials!



Overview Info:
- http://boinc.berkeley.edu/trac/wiki/BasicConcepts

Resources For Setup:
- http://boinc.berkeley.edu/trac/wiki/QuickStart

Config File:
- http://www.boinc-wiki.info/Project_Configuration_File
- http://boinc.berkeley.edu/trac/wiki/ProjectConfigFile
- http://boinc.berkeley.edu/trac/wiki/ProjectOptions
- http://boinc.berkeley.edu/trac/wiki/ProjectDaemons
- http://www.boinc-wiki.info/BOINC_Server-Side_Daemon_Program

Some of the Daemons n the config file:
- http://boinc.berkeley.edu/trac/wiki/BackendPrograms
- http://boinc.berkeley.edu/trac/wiki/FileDeleter
- http://www.boinc-wiki.info/Assimilator_Daemon
- http://www.boinc-wiki.info/Validator_Daemon

create_work
- http://www.boinc-wiki.info/Generating_Work#Creating_Work_Unit_Records

- Slides w/ references
- Sample Templates
- Scripts

BOINC

RSA CONFERENCE 2012

History of BOINC

1999 - SETI@home launches to the public

2004 - BOINC project begins
- First BOINC Project launches (protein prediction)

2008 - GPU powered applications introduced

~2 million users
~6 million computers

Top projects (by credit):

1 SETI@home
2 MilkyWay@home
3 Collatz Conjecture
32ish SHA-1 Collision Search

Why would I use it?

Handles
- network problems
- client errors
- server/client reboots
- file integrity

Supports
- running time limits
- multiple platforms
- untrustable clients
- GPUs and odd applications
- credit/reputation & teams
- assimilation/validation

How Do I Use BOINC?

1. Set up a BOINC Server
2. Edit config.xml
3. Lock down the server
4. Figure out how to distribute the work
5. Set up an application
6. Set up a client image
7. Automate the client image
8. Create workunits

Is it hard?

1. Set up a BOINC Server - Easy
2. Edit config.xml - Easy
3. Lock down the server - Should be easy
4. Figure out how to distribute the work - Could be tricky
5. Set up an application - Trial and Error
6. Set up a client image - Easy
7. Automate the client image - Easy
8. Create workunits - Potentially annoying

HACKERS CAN TURN YOUR HOME COMPUTER INTO A BOMB

... & blow your family to smithereens!

By RANDY JEFFRIES / *Weekly World News*

WASHINGTON — Right now, computer hackers have the ability to turn your home computer into a bomb and blow you to Kingdom Come — and they can do it anonymously from thousands of miles away!

Experts say the recent "break-ins" that paralyzed the Amazon.com, Buy.com and eBAY websites are tame compared to what will happen in the near future.

Computer expert Arnold Yabenson, president of the Washington-based consumer group National CyberCrime Prevention Foundation (NCPF), says that as far as computer crime is concerned, we've only seen the tip of the iceberg.

"The criminals who knocked out those three major online businesses are the least of our worries," Yabenson told *Weekly World News*.

"There are brilliant but unscrupulous hackers out there who have developed technologies that the average person can't even dream of. Even people who are familiar with how computers work have trouble getting their minds around the terrible things that can be done.

"It is already possible for an assassin to send someone an e-mail with an innocent-looking attachment connected to it. When the receiver downloads the attachment, the electrical current and molecular structure of the central processing unit is altered, causing it to blast apart like a large hand grenade.

"As shocking as this is, it shouldn't surprise anyone. It's just the next step in an ever-escalating progression of horrors conceived and instituted by hackers."

Yabenson points out that these dangerous sociopaths have already:

- Vandalized FBI and U. S. Army websites.
- Broken into Chinese military networks.
- Come within two digits of cracking an 37-digit Russian security code that would have sent deadly missiles hurtling toward five of America's major cities.

"As dangerous as this technology is right now, it's going to get much scarier," Yabenson said.

"Soon it will be sold to terrorists cults and fanatical religious-fringe groups.

"Instead of blowing up a single plane, these groups will be able to patch into the central computer of a large airline and blow up hundreds of planes at once.

"And worse, this e-mail bomb program will eventually find its way into the hands of anyone who wants it.

"That means anyone who has a quarrel with you, holds a grudge against you or just plain doesn't like your looks, can kill you and never be found out."

KABOOM! It might not look like it, but an innocent home computer like this one can be turned into a deadly weapon.

Sickos can wreak death and destruction from thousands of miles away!

Arnold Yabenson.

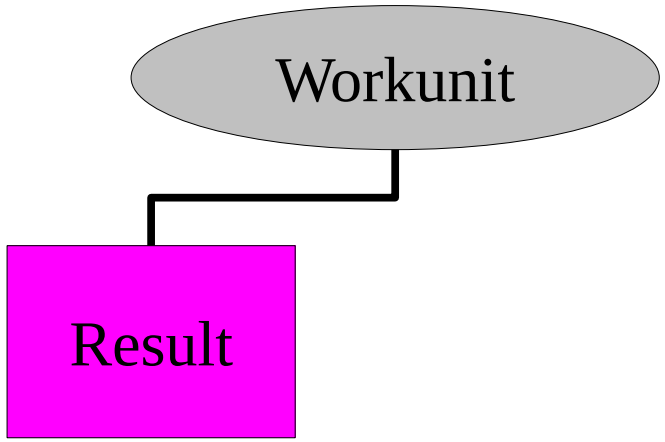
who's got two thumbs and isn't responsible for you getting owned?

Lifecycle of a unit of work

1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'results'
5. Client computes and upload the outcome of the 'result'
6. Server Validation
7. Server Assimilation
8. Server Deletion

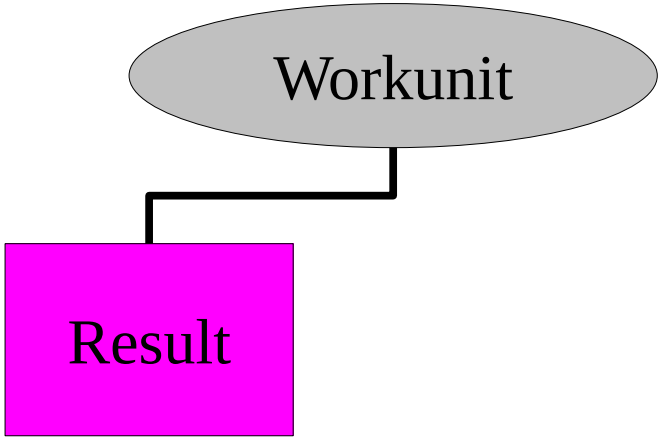
Lifecycle of a unit of work

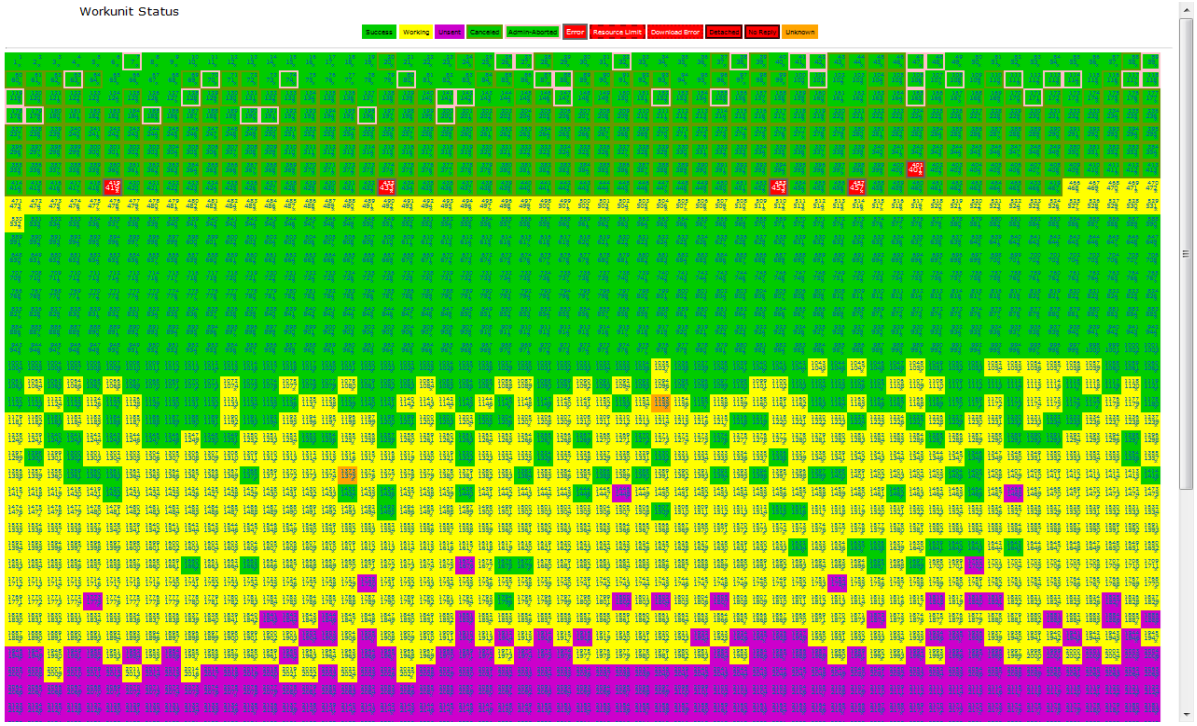
1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'results'
5. Client computes and upload the outcome of the 'result'
6. Server Validation
7. Server Assimilation
8. Server Deletion



Lifecycle of a unit of work

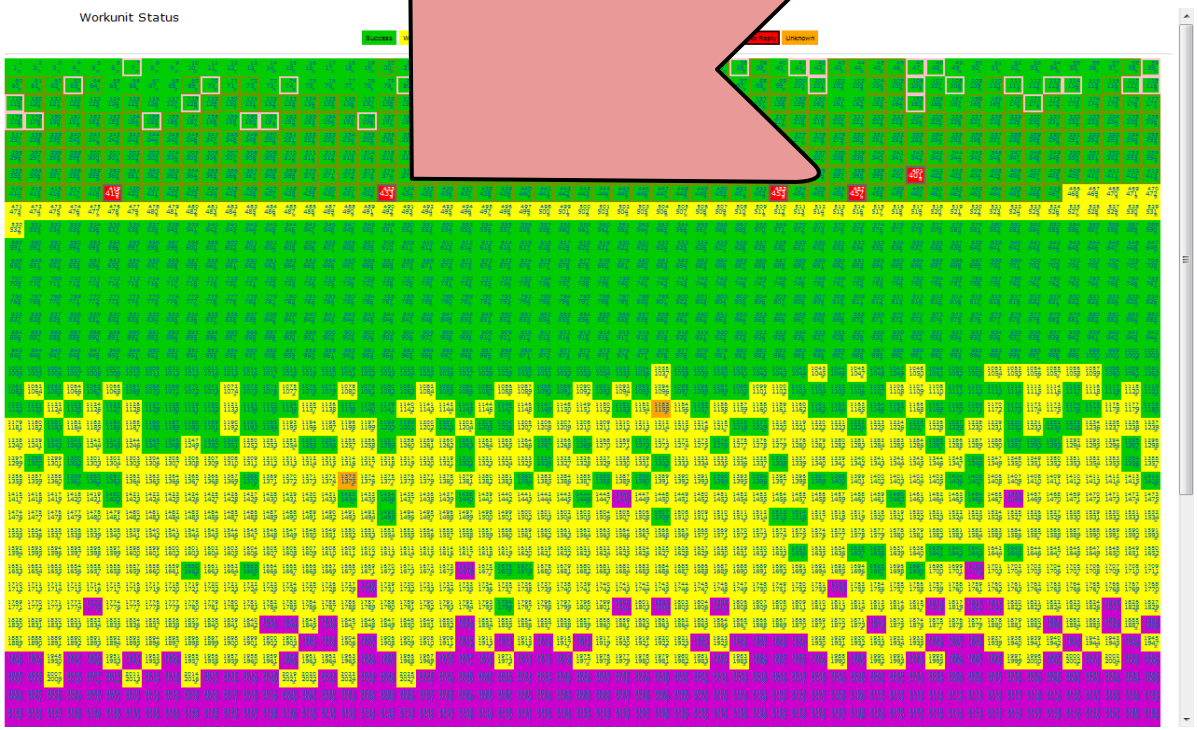
1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'results'
5. Client computes and upload the outcome of the 'result'
6. Server Validation
7. Server Assimilation
8. Server Deletion

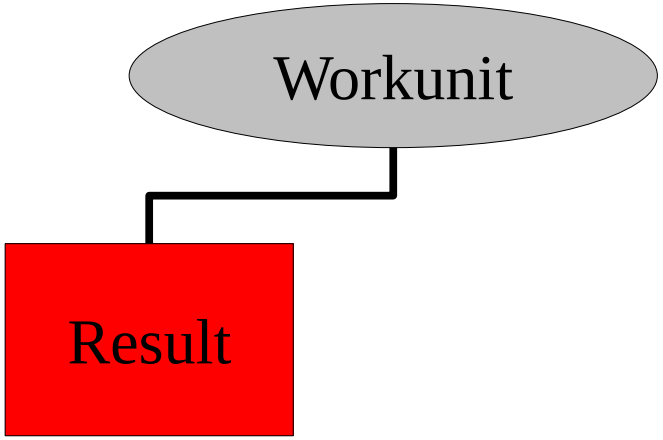




Lifecycle of a unit of work

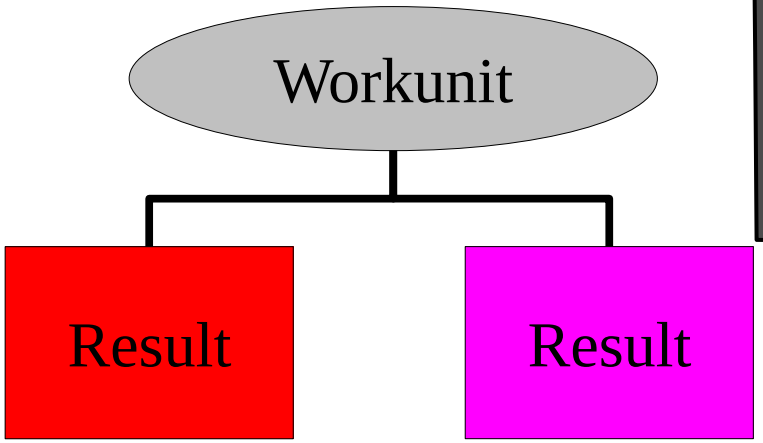
1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'resu
5. Client computes and upload the outco sult'
6. Server Validation
7. Server Assimilation
8. Server Deletion

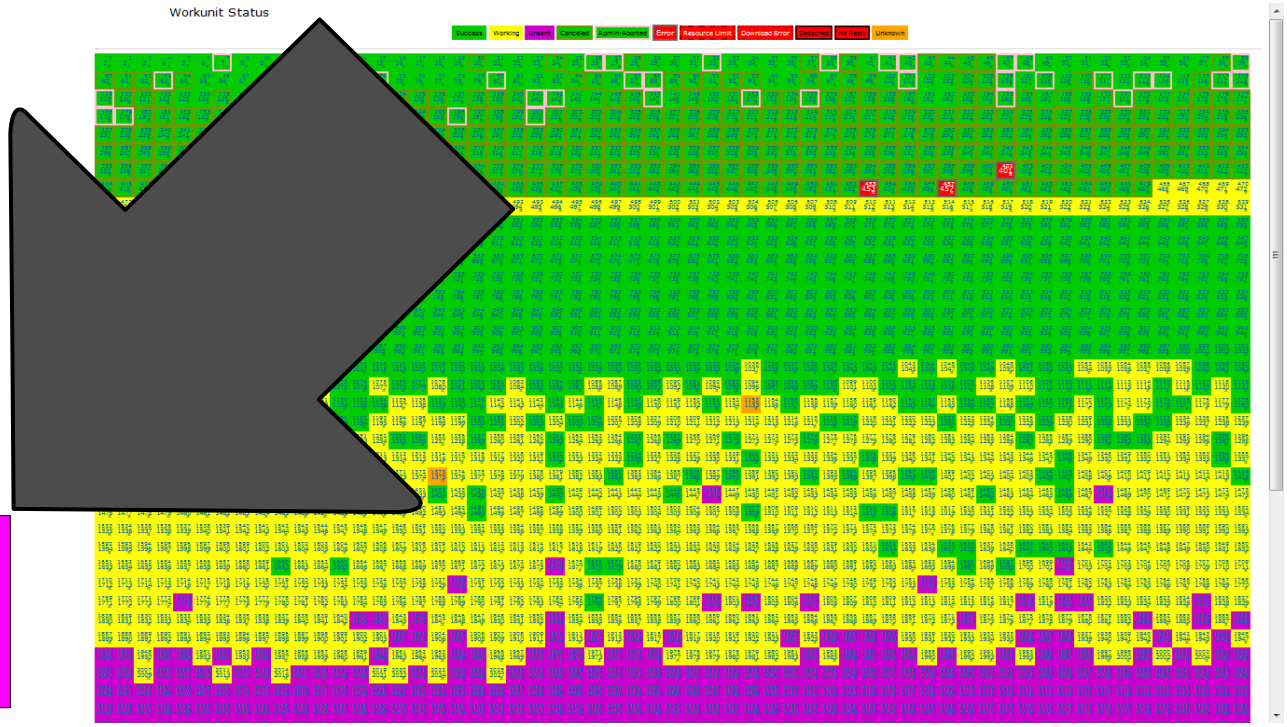




Lifecycle of a unit of work

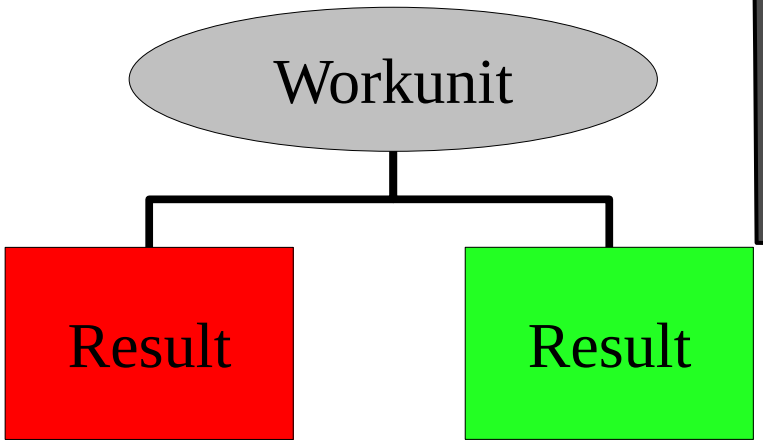
1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'results'
5. Client computes and upload the outcome of the 'result'
6. Server Validation
7. Server Assimilation
8. Server Deletion

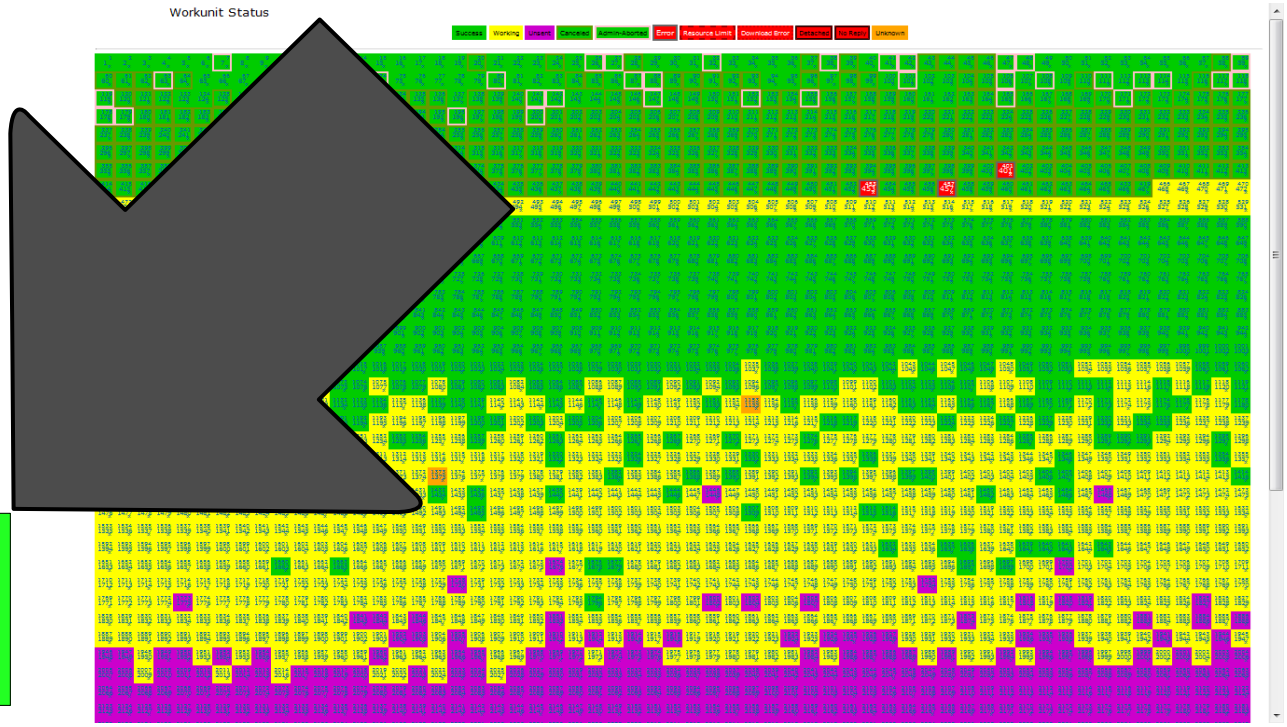




Lifecycle of a unit of work

1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'results'
5. Client computes and upload the outcome of the 'result'
6. Server Validation
7. Server Assimilation
8. Server Deletion





Lifecycle of a unit of work

1. Create boinc workunits
2. Load them into the server
3. Server creates 'results'
4. Client connects and is assigned 'results'
5. Client computes and upload the outcome of the 'result'
6. Server Validation
7. Server Assimilation
8. Server Deletion

Can actually be really complicated!

But for us.... no.
- sample_bitwise_validator
- sample_assimilator

512 Bit RSA Key Factoring

RSA CONFERENCE 2012

History

p*q = n <- n is a semiprime
5 * 3 = 15 <- 15 is a semiprime
(76-digit p) * (76 digit q) = (155 digit n)

- Aug 1999 - 512 Bit Factored for the first time (publicly)
- 2004 - GGNFS, msieve and factLat.pl in development
- July 2009 - TI83+ Signing Key Factored
- Aug 2009 - Factoring Service Offered: $5000/key
- Sept 2009 - All TI Signing Keys factored
- Dec 2009 - 768 Bit factored for the first time (publicly)
  40 + 1500 + 155 = 1695 Core-Years

How Do I Factor

1. Trial Division?
   - Is it divisible by 2? 3? 5? 7? 11? 13?

2. Pollard Rho

3. ECM

4. General Number Field Sieve

How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. $f(x)$ & $g(x)$ of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"

How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

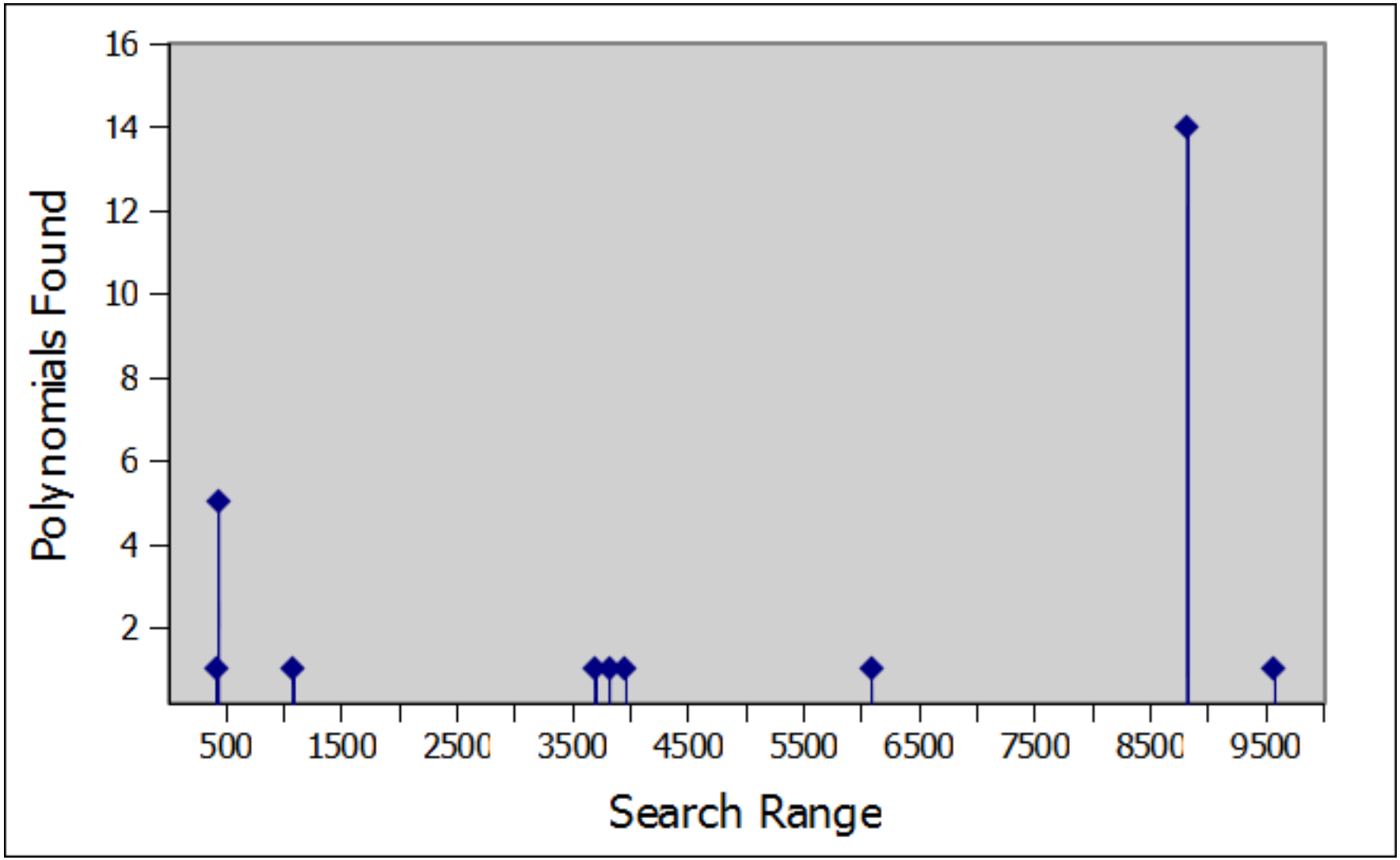
1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"

Slow & Unparallelizable

512 Bit ~8 Core-Days
768 Bit ~155 Core-Years*

How Do I Factor

1.Polynomial Selection



msieve by jasonp

Beautiful C Code
All Factoring Algorithms
- Trial Division
- Phollard Rho
- ECM
- GNFS

Actively Developed & Maintained
Active Support Channel
Active Community

Polynomial Selection

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

Sieving

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

Combine

1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"

msieve by jasonp

Beautiful C Code
All Factoring Algorithms
- Trial Division
- Phollard Rho
- ECM
- GNFS

Actively Developed & Maintained
Active Support Channel
Active Community

Polynomial Selection

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

Sieving

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

Combine

1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"

msieve by jasonp

jasonp?

Polynomial Selection

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

Sieving

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

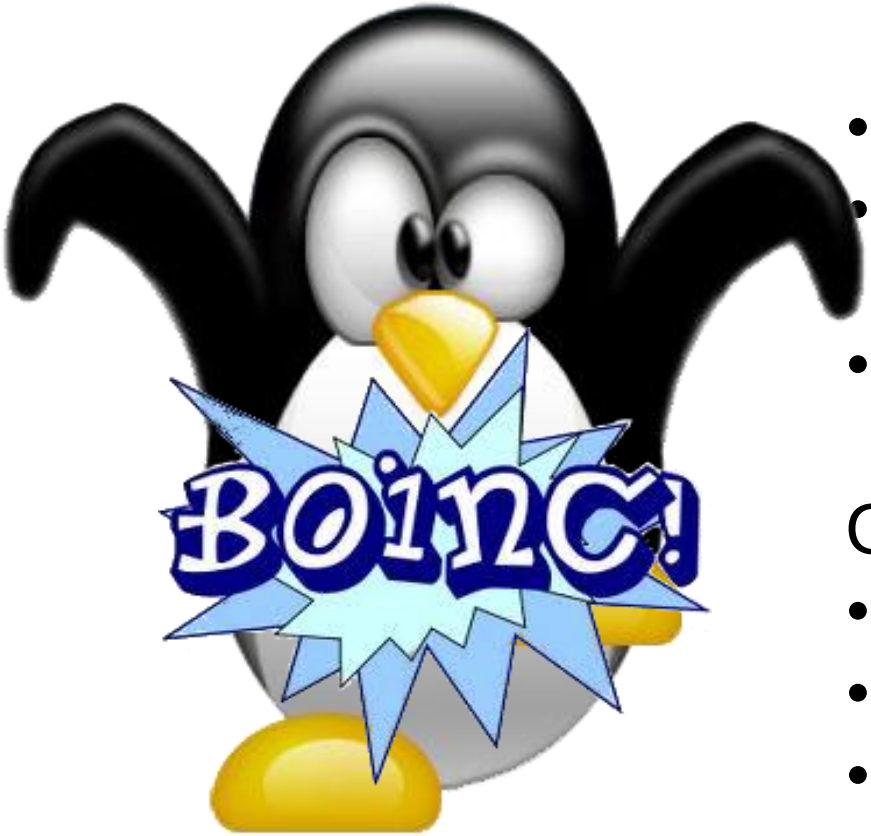
Combine

1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"

BOINC-izing an Open Source App

- fopen -> boinc_fopen
- boinc_init()
- boinc_finish(return_value)
- link with boinc libs

BOINC-izing an Open Source App



- fopen -> boinc_fopen

- boinc_init()
- boinc_finish(return_value)

- link with boinc libs

Optional:
- Checkpointing
- Critical Sections
- boinc_fraction_done
- boinc_need_network

Rewiring msieve into a BOINC Application

```
@@ -2852,7 +2891,33 @@
+#ifdef HAVE_BOINC
 int main(int argc, char **argv)
+{
+ int newArgc, ret;
+ char** newArgv;
+ myboincstart(&newArgc, &newArgv, argv[0]);
+ ret = sieve_main(newArgc, newArgv);
+ boinc_finish(ret);
+ return ret;
+}
+
+
+int sieve_main(int argc, char **argv)
+#else
+int main(int argc, char **argv)
+#endif
 {
```

Rewiring msieve into a BOINC Application

```
void myboincstart(int* argc, char *** argv, char* name)
{
    char in[500], out[500];
    boinc_init();
    boinc_resolve_filename("input_data", in, 500);
    boinc_resolve_filename("output_data", out, 500);

    *argc = 0;
    argv = new char*[7];
    argv[(*argc)++] = name;
    argv[(*argc)++] = "-i";
    argv[(*argc)++] = in;
    argv[(*argc)++] = "-nf";
    argv[(*argc)++] = out;
    argv[(*argc)++] = "-np";
    argv[(*argc)++] = "\0";
}
```

*abbreviated example

BOINC-izing an Open Source App

- fopen -> boinc_fopen
- boinc_init()
- boinc_finish(return_value)
- link with boinc libs

BOINC-izing an Open Source App

- fopen -> boinc_fopen

- boinc_init()
- boinc_finish(return_value)

- boinc_resolve_filename

```
fopen("logfile", "w")

boinc_resolve_filename("logfile", buffer);
boinc_fopen(buffer, "w")
// buffer -> workunit12345_0_1
```

- link with boinc libs

Application Templates

Input

```
<file_info>
   <number>0</number>
    [ <sticky /> ]
    [ <nodelete /> ]
</file_info>
<workunit>
  <file_ref>
   <file_number>0</file_number>
   <open_name>rsakey</open_name>
   [ <copy_file/> ]
  </file_ref>

  <target_nresults>1</target_nresults>
</workunit>
```

Application Templates

Input

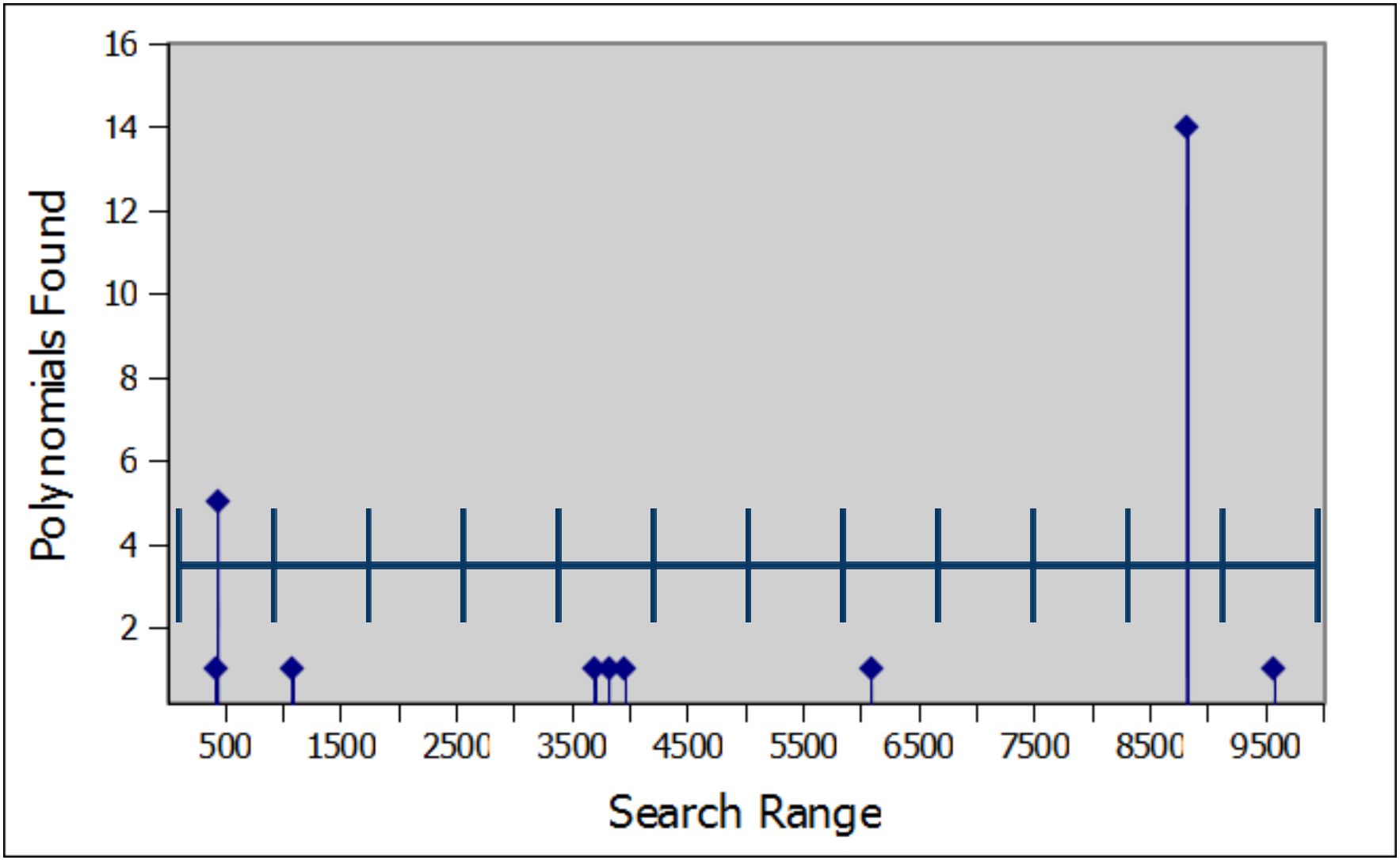
```
<file_info>
   <number>0</number>
    [ <sticky /> ]
    [ <nodelete /> ]
</file_info>
<workunit>
  <file_ref>
    <file_number>0</file_number>
    <open_name>rsakey</open_name>
    [ <copy_file/> ]
  </file_ref>

  <target_nresults>1</target_nresults>
</workunit>
```

Output

```
<file_info>
  <name><OUTFILE_0/></name>
  <generated_locally/>
  <upload_when_present/>
  <url><UPLOAD_URL/></url>
</file_info>
<result>
 <file_ref>
  <file_name><OUTFILE_0/>
              </file_name>
 <open_name>logfile</open_name>
 [ <copy_file>0|1</copy_file> ]
 [ <optional>0|1</optional> ]
 </file_ref>
</result>
```

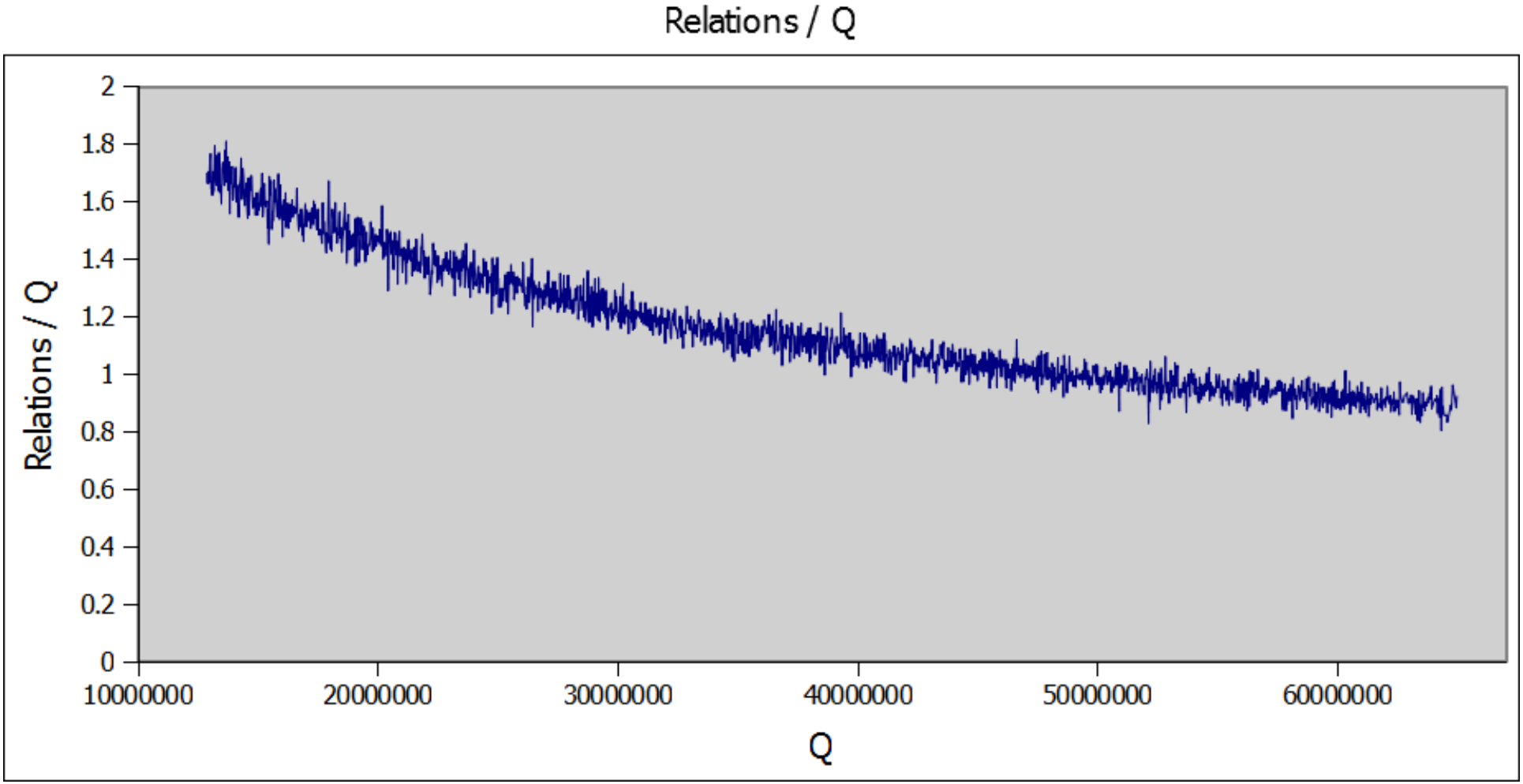
How Do I Factor

1.Polynomial Selection



How Do I Factor

1. Polynomial Selection
2. Sieving

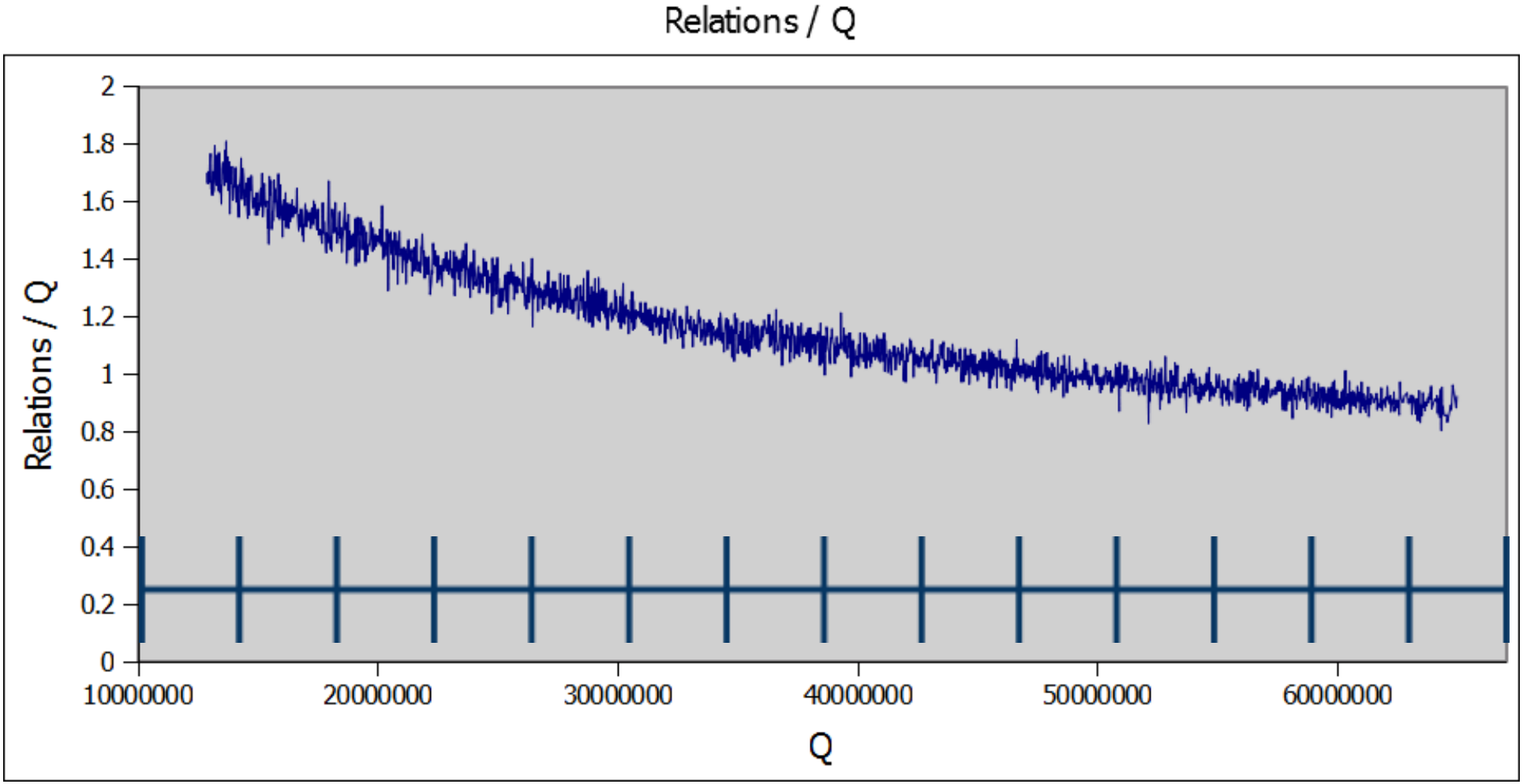


Sieving with GGNFS in BOINC

```
+#ifdef HAVE_BOINC
+int boincstart(int argc_init, char **argv) {
+  boinc_init();
+  boinc_resolve_filename("input_data", path_in, sizeof(path_in) );
+  boinc_resolve_filename("output_data", path_out,sizeof(path_out));
+  argv[argc_init++]="-R";
+  argv[argc_init++]="-a";
+  argv[argc_init++]="-o";
+  argv[argc_init++]=path_out;
+  argv[argc_init++]=path_in;
+  return argc_init;
+}
int main(int argc, char **argv) {
+       int app_argc, retcode;
+       char* app_argv[ARGVCOUNT];
+       app_argv[0] = argv[0];
+       app_argc= boincstart(1,app_argv);
+       retcode= main_lasieve(app_argc,app_argv);
+       boinc_finish(retcode);
+       return retcode;
+}
+int main_lasieve(int argc, char **argv)
+#else
+int main(int argc, char **argv)
+#endif
```

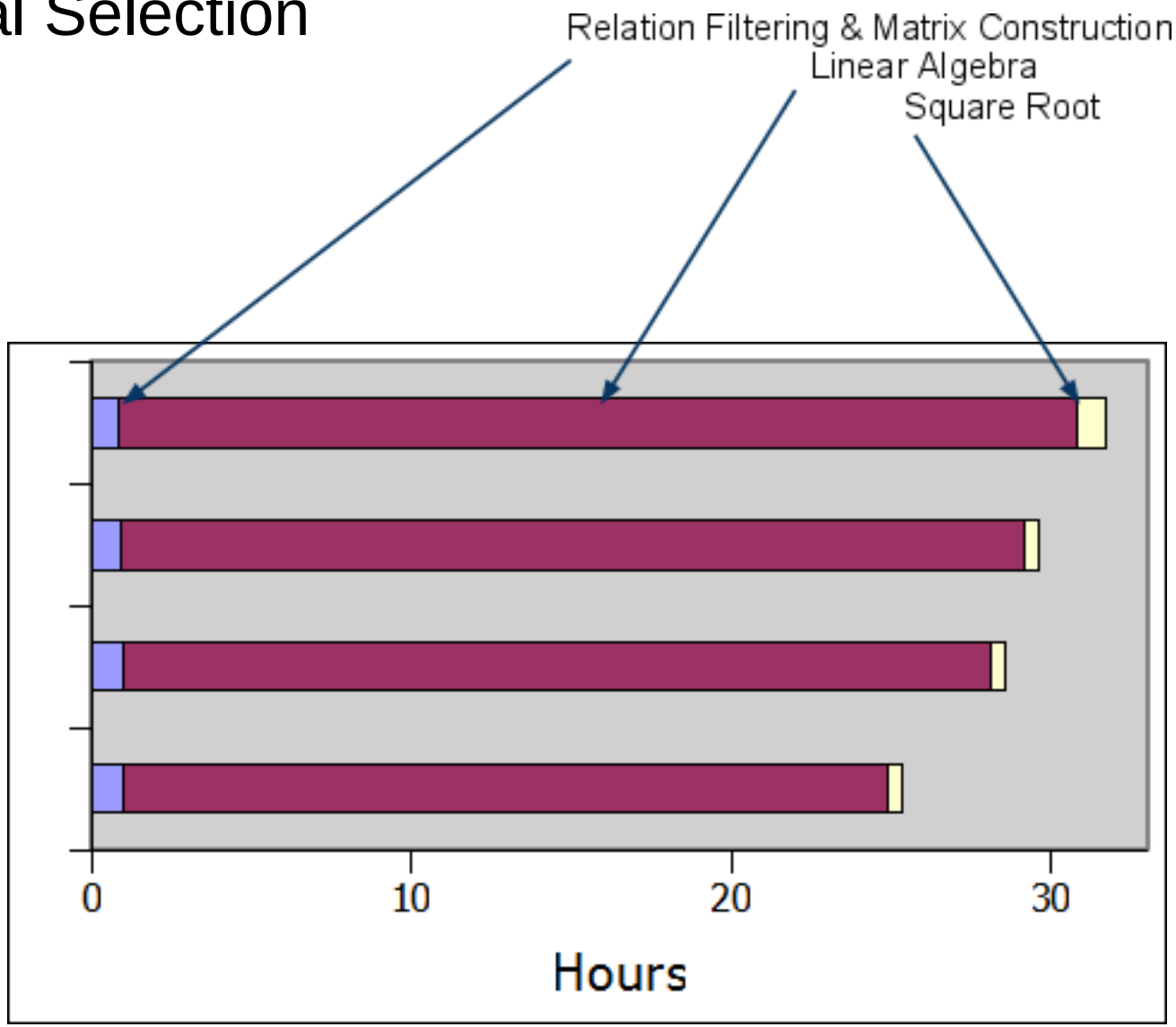
How Do I Factor

1. Polynomial Selection
2. Sieving



How Do I Factor

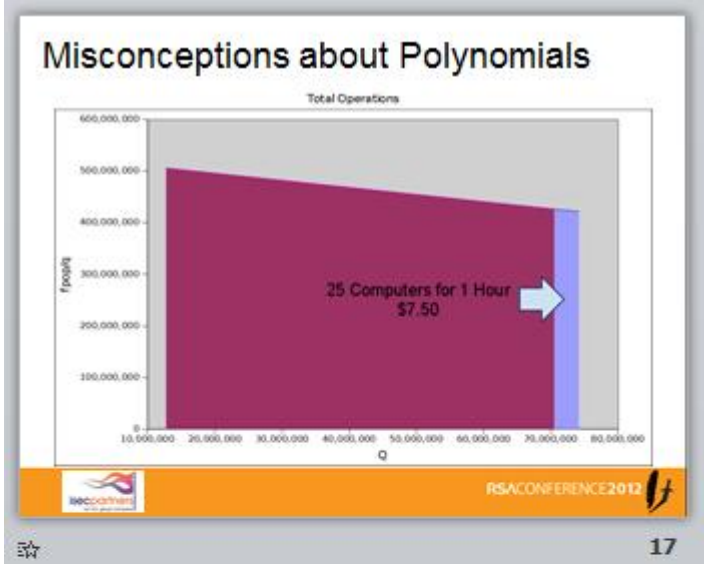
1. Polynomial Selection
2. Sieving
3. Combine

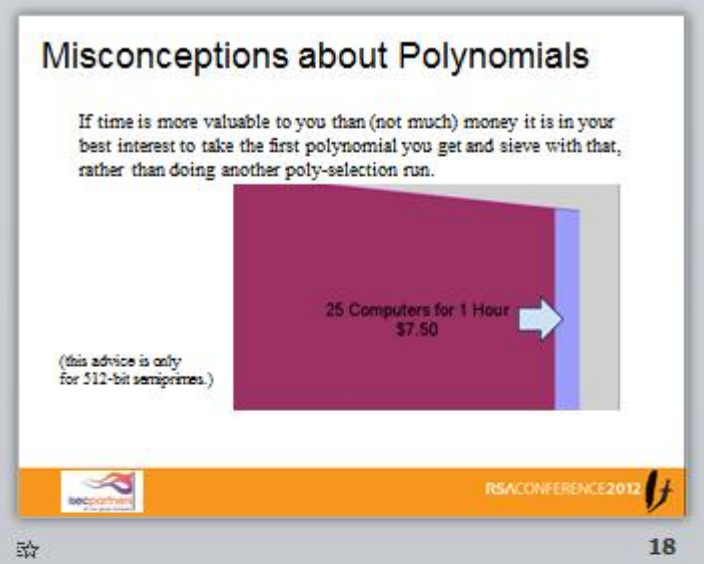


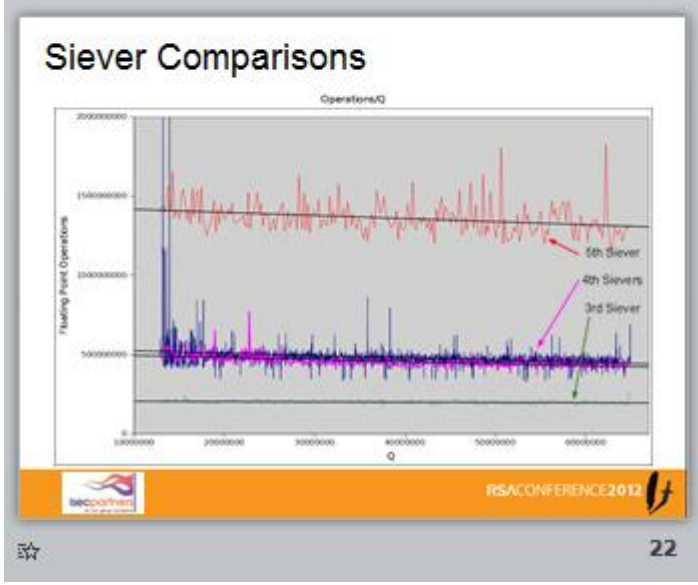
The payoff

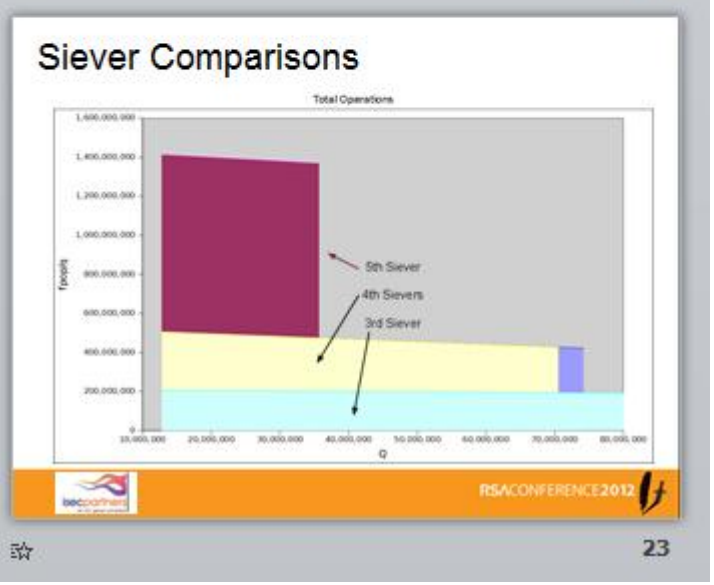
```
$ wget -q https://www.eff.org/files/syrian-facebook-attack.pem
$ openssl x509 -noout -modulus -in syrian-facebook-attack.pem
Modulus=D5997DCA6577FCD964FE316987BDED93BA4D9644844629CF26CDA9CC
        EED253AD2EE646EE1CF8AC95D18FA014A2EC29672009BD684F79579A
        AA8D7E73E797F6B3
$ python
>>> n = int('D5997DCA6577FCD964FE316987BDED93BA4D9644844629CF26C
             DA9CCEED253AD2EE646EE1CF8AC95D18FA014A2EC29672009BD
             684F79579AAA8D7E73E797F6B3', 16)
>>> n
111871175171822184890047853438937107834419894175266549329387466 5
   918216098748833844280207239400866608597143161438766170346657 8
   380319053521569571009086355123L
>>> p = 10431832711621412355078235716253440775473942492929486 91
        8608964371166209731389 9
>>> q = 10724019284472797831715458754067770262540924005825331 69
        6456831084693273770517 7
>>> n - (p * q)
0L
```

Factoring Details









Moved to their own slide deck for time/relevance

Available on github.

So Far

- BOINC
  - Why and How
  - Applications - Open source ⇨ BOINC Application
- Factoring RSA

Next

- BOINC
  - Close Source Applications
  - GPU Applications
- Hands-off Cracking Passwords
- Log File Analysis

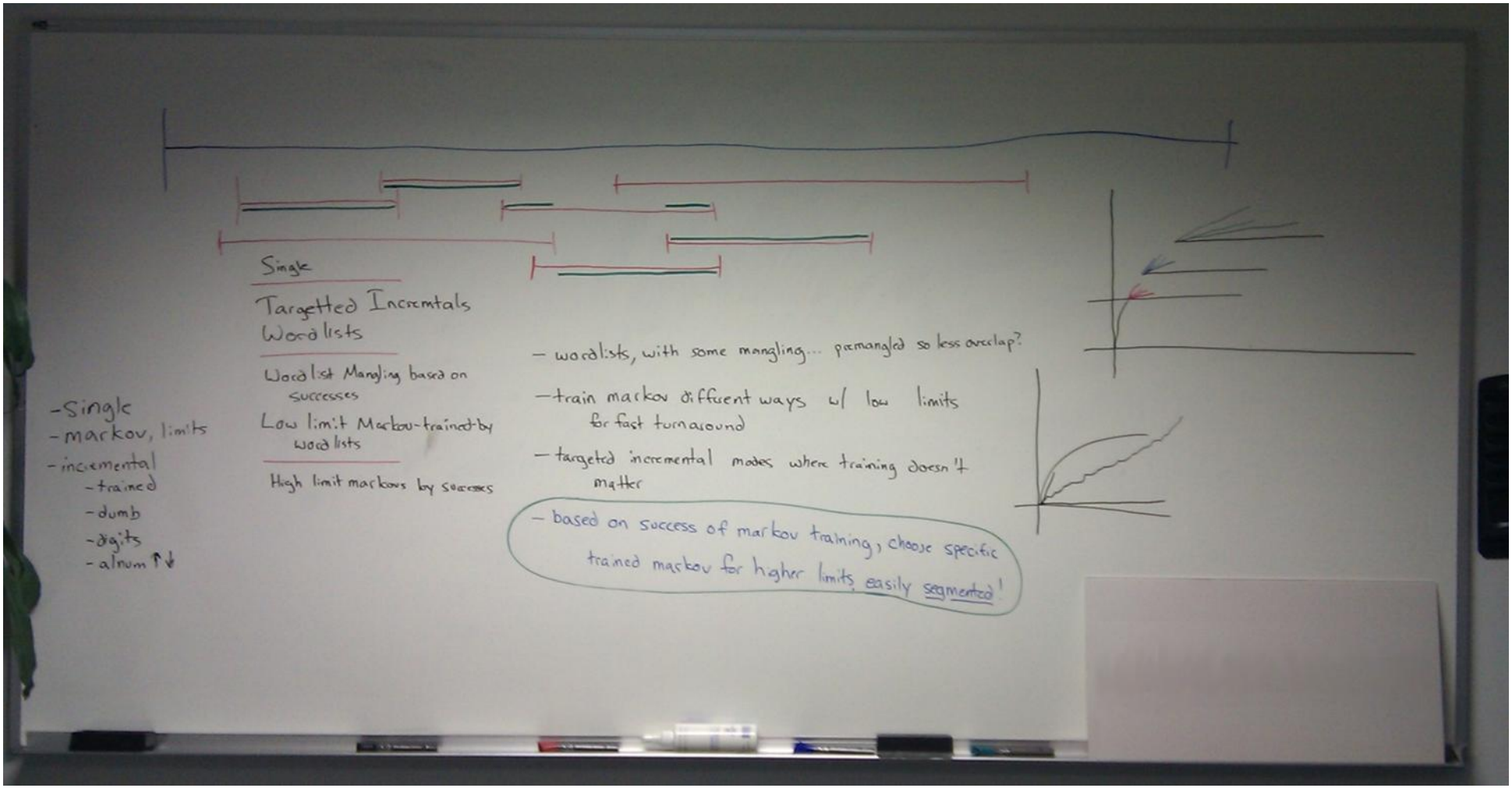
Cracking

How do you Parrallelize Cracking?

- john
  - Several MPI Patches for john - but only on clusters
  - Mode:External - but non-trivial overhead when splitting
  - Cheap Hacks (bad idea)

- hashcat family (hashcat, oclHashcat, cudaHashcat)
  - Not much you can do

Enter the Magic

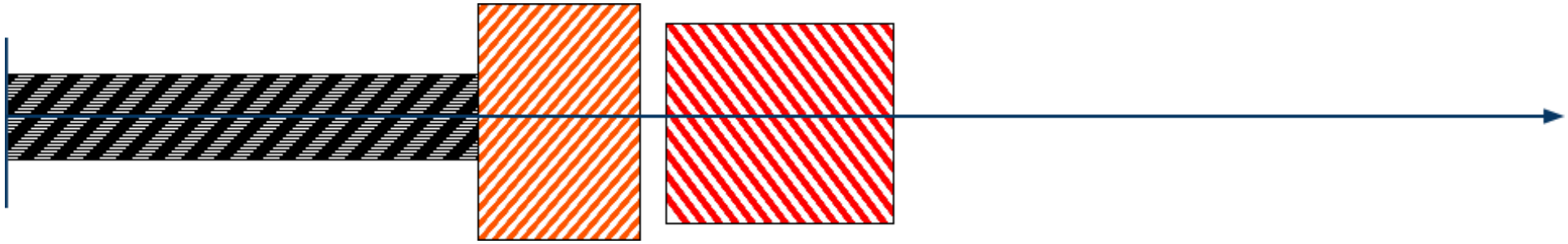


All Possible Passwords

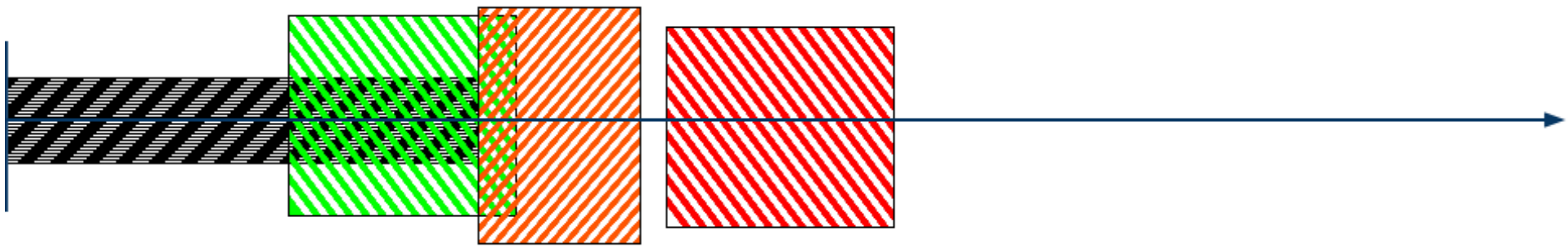
Brute Force

Wordlist

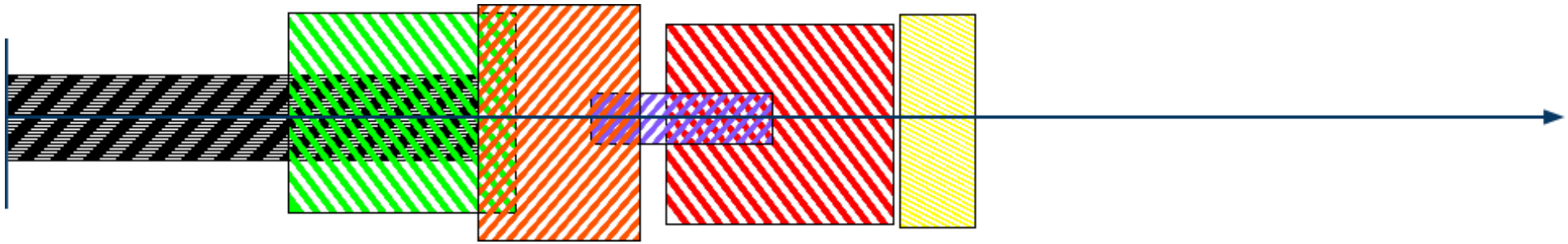
Two Wordlists!



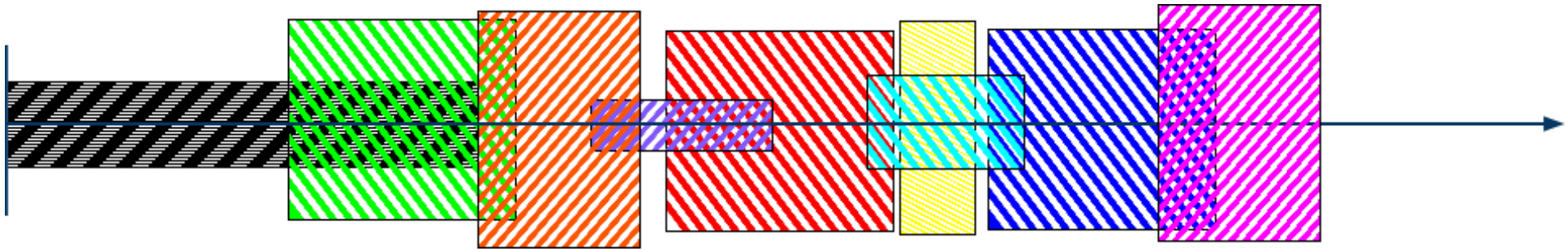
Let me try this…



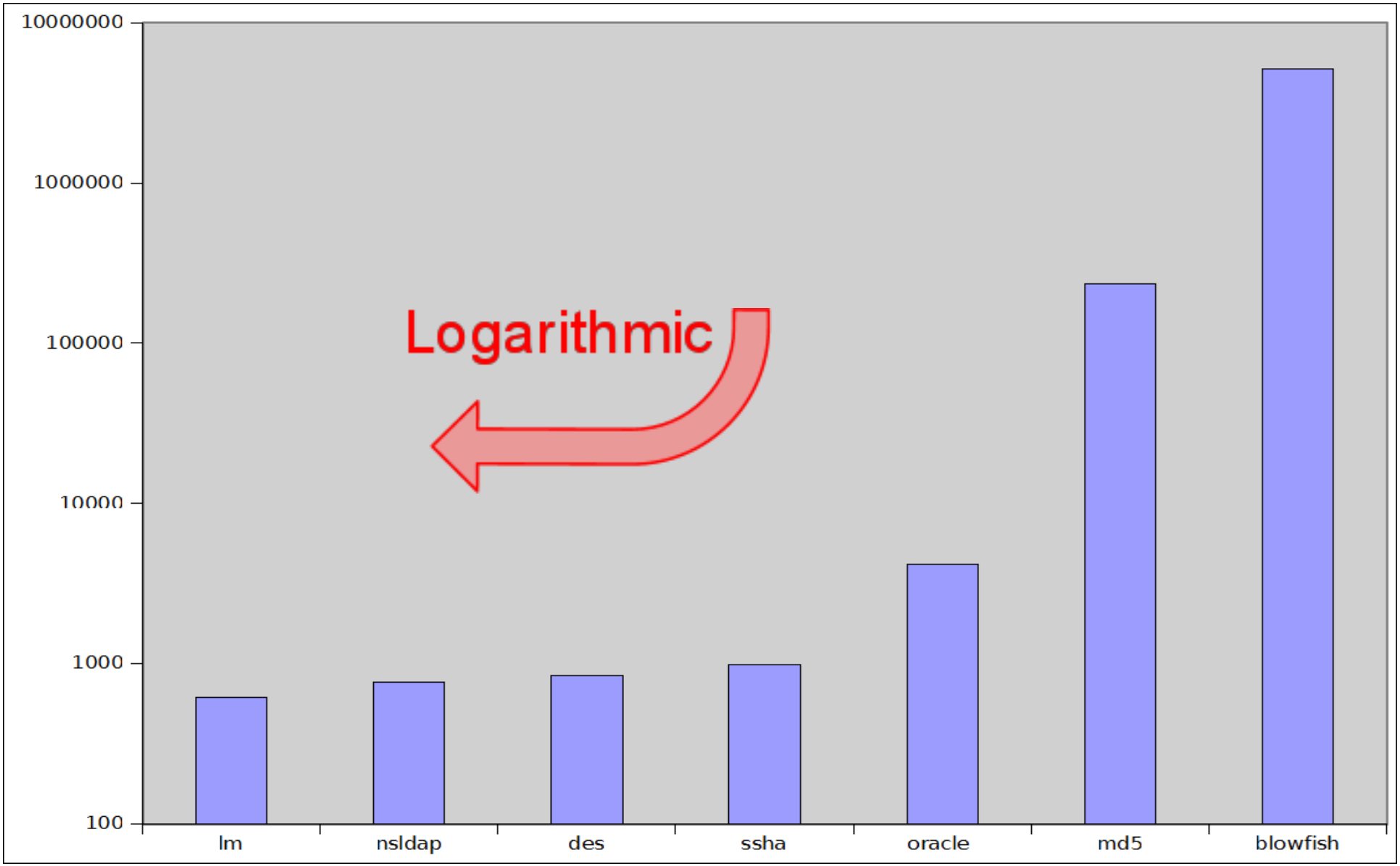
Foreign Wordlists!



Kitchen Sink.



Not all hashes are created equal

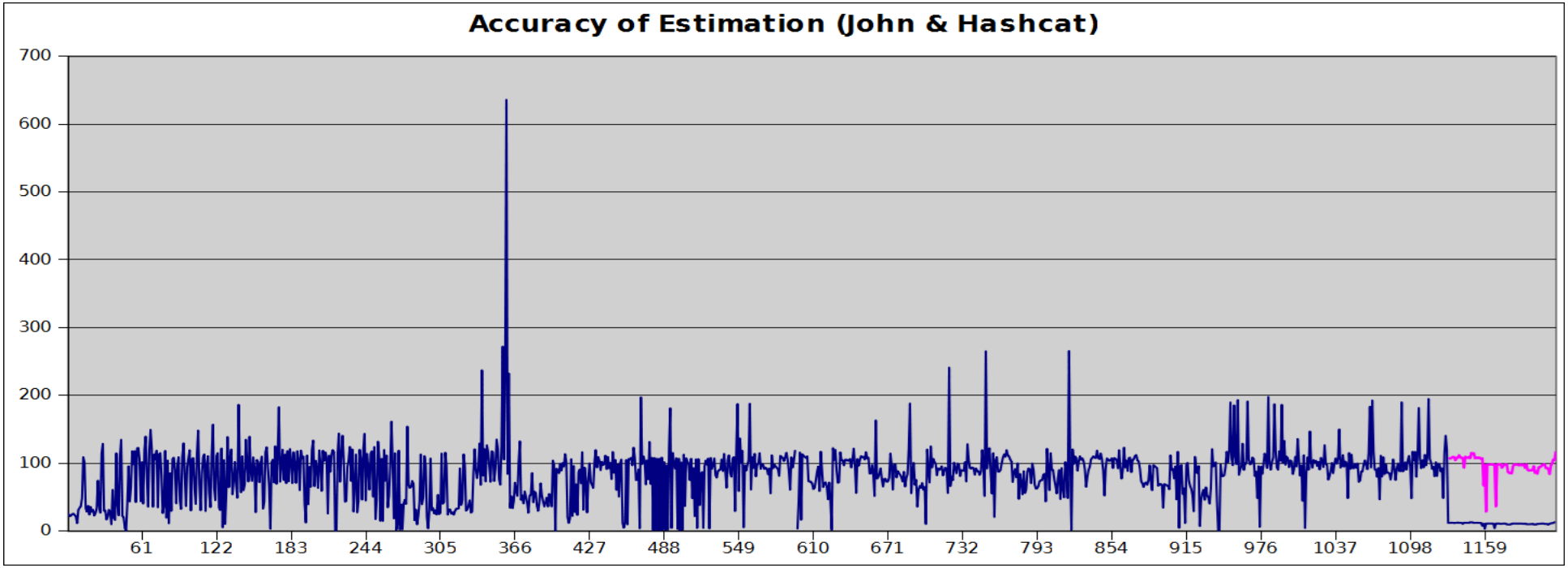


My Approach

Phase 1: --single
Phase 2: 1 hour incremental

My Approach

Phase 1: --single
Phase 2: 1 hour incremental



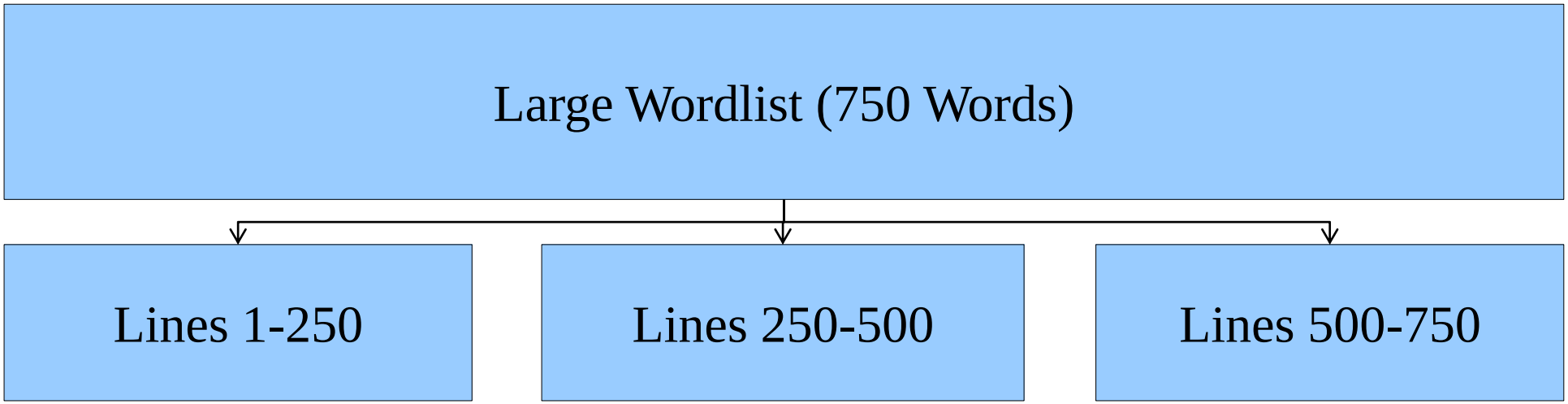
My Approach

Phase 1: --single
Phase 2: 1 hour incremental

Large Wordlist

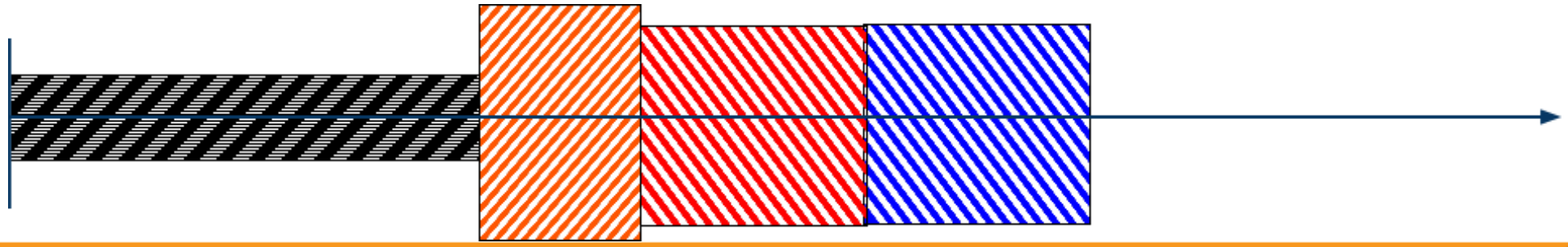
My Approach

Phase 1: --single
Phase 2: 1 hour incremental



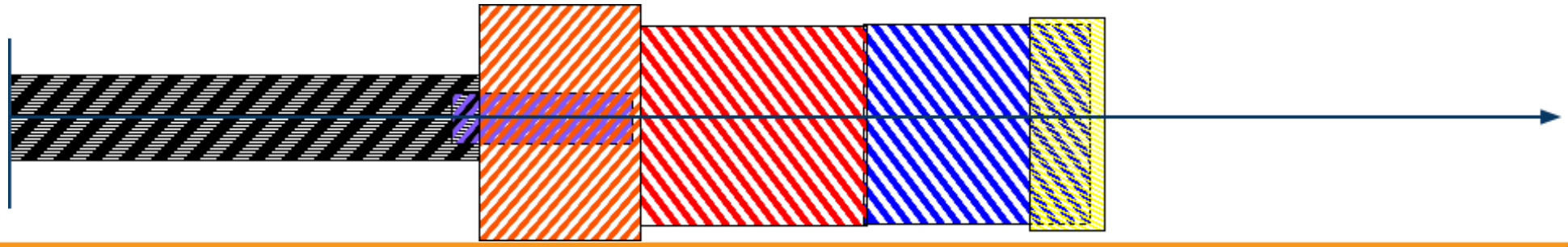
My Approach

Phase 1: --single
Phase 2: 1 hour incremental
Wordlists



My Approach

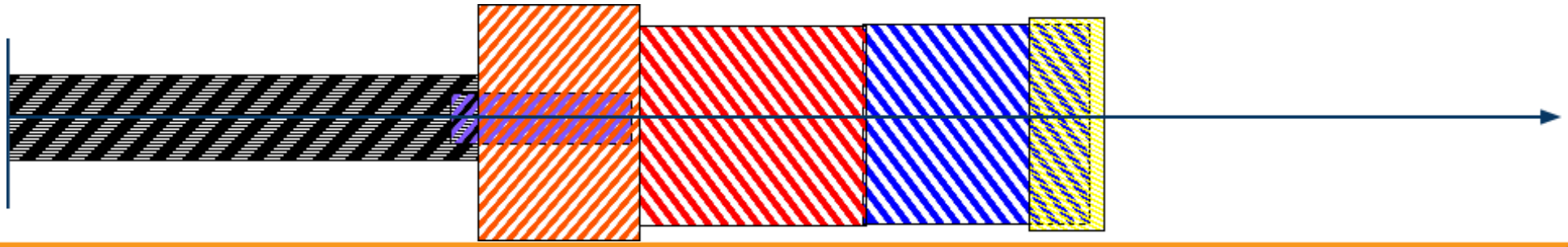
Phase 1: --single
Phase 2: 1 hour incremental
Wordlists
Phase 3: Wordlist Rules
High-Probability Markov Words



My Approach

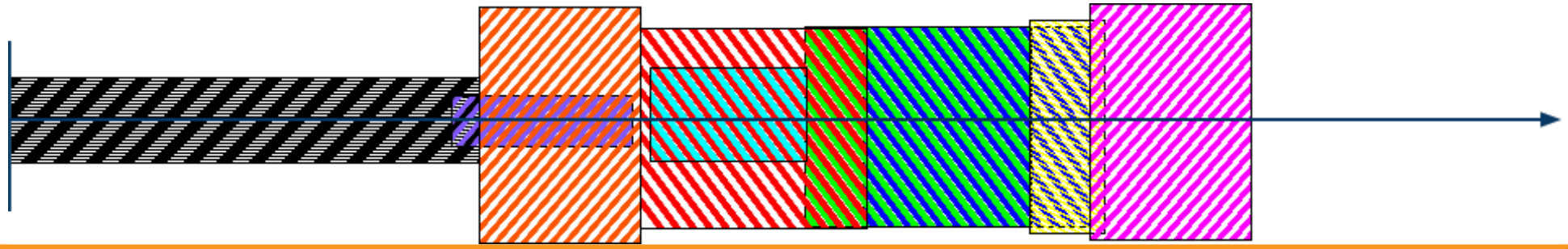
Phase 1: --single
Phase 2: 1 hour incremental
Wordlists
Phase 3: Wordlist Rules
High-Probability Markov Words

Very carefully pruned wordlists.

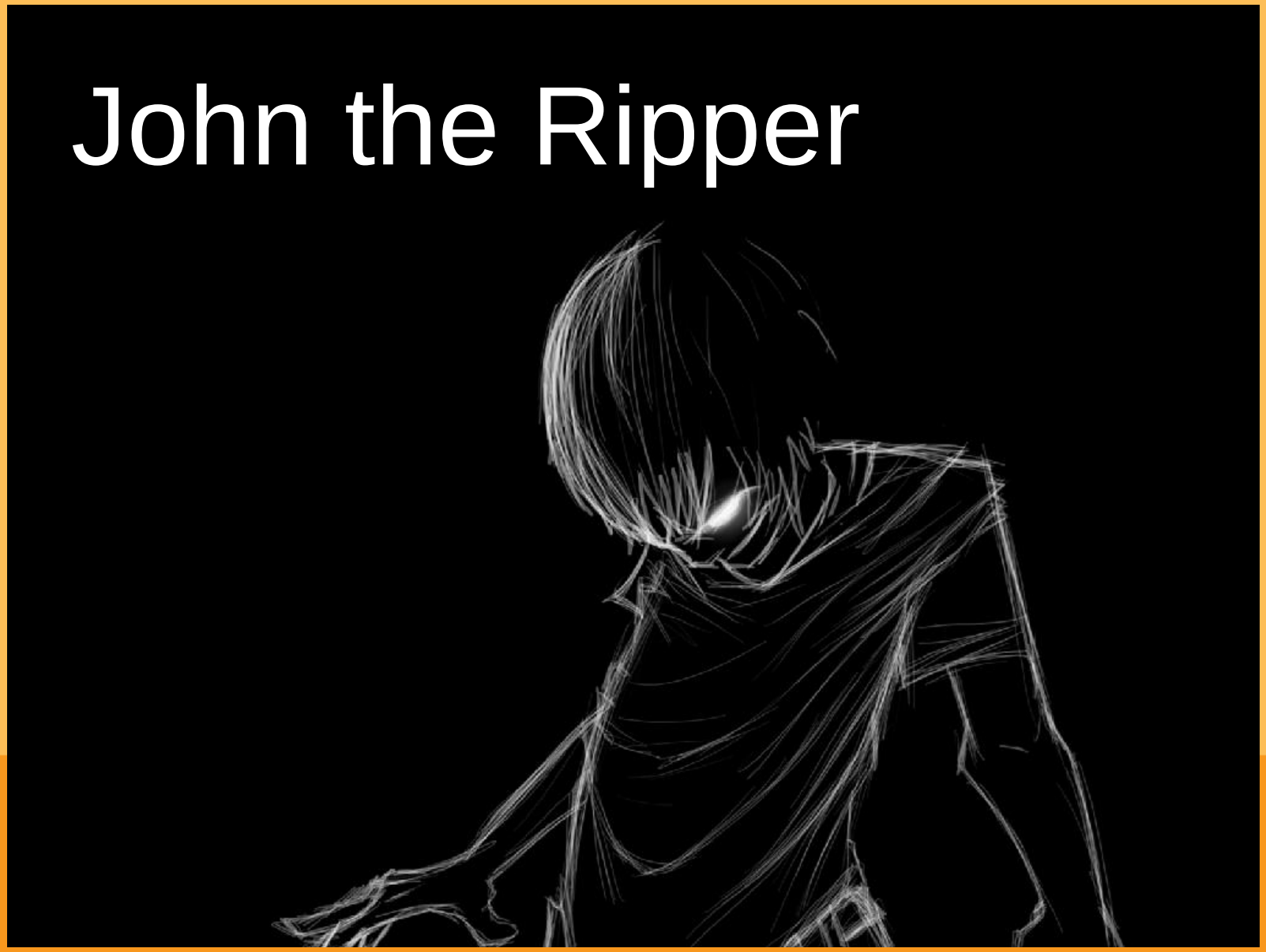


My Approach

Phase 1: --single
Phase 2: 1 hour incremental
Wordlists
Phase 3: Wordlist Rules
High-Probability Markov Words
Phase 4: Phase 3 Markovs + Rules
Low-Probability Markov Words
Phase 5: Phase 4 Markovs + Rules



John the Ripper



Rewiring John into a BOINC App

```
+int main(int argc, char **argv) {
+       int status = boinc_init();
+       boinc_resolve_filename("john.conf", confFile, sizeof(confFile) );
+       boinc_resolve_filename("passwordlist", passlist, sizeof(passlist) );
+
+       int i, newArgc = 2, hasWordlist = 0;
+       for(i=1; i < argc; i++) {
+            newArgc++;
+            hasWordlist = strstr(argv[i], "<<WORDLIST>>") ? i : hasWordlist; }
+       if(hasWordlist) {
+           boinc_resolve_filename("wordlist", wordlistName, 512 );
+           snprintf(wordlistParameter, 612, "--wordlist=%s", wordlistName); }
+
+       newArgv[i=0] = argv[0];
+       for (i++; i<argc; i++) newArgv[i] = i == hasWordlist ?
+                                                wordlistParameter : argv[i];
+       newArgv[i] = passlist;
+       int ret = john_main(newArgc, newArgv);
+       boinc_finish(ret);
+       return ret;
+}
+int john_main(int argc, char **argv)
+#else
int main(int argc, char **argv)
+#endif
```

*heavily abbreviated and trimmed

Application Versions

Add a new application:
1. Update project.xml
2. xadd

Add a new version:
1. copy files correctly
2. update_versions

Application Versions

```
apps/
  name/
    name_version.minor_platform[.ext]
```

Add a new application:
1. Update project.xml
2. xadd

Add a new version:
1. copy files correctly
2. update_versions

Application Versions

```
apps/
  name/
    name_version.minor_platform[.ext]

  msieve/
    msieve_148.1_linux
```

Add a new application:
1. Update project.xml
2. xadd

Add a new version:
1. copy files correctly
2. update_versions

Application Versions

```
apps/
  name/
    name_version.minor_platform[.ext]

  msieve/
    msieve_148.1_linux


  newapp/
    newapp_1.0_linux/
      newapp_1.0_linux
      resourcefile.dat
      somethingelse.db
```

Add a new application:
1.Update project.xml
2.xadd

Add a new version:
1.copy files correctly
2.update_versions

Application Versions

```
apps/
  name/
    name_version.minor_platform[.ext]

  msieve/
    msieve_148.1_linux


  newapp/
    newapp_1.0_linux/
      newapp_1.0_linux
      resourcefile.dat
      somethingelse.db
    newapp_1.1_linux/
      subfolder/
        stuff.db
```

Add a new application:
1.Update project.xml
2.xadd

Add a new version:
1.copy files correctly
2.update_versions

Hashcat
hashcat, oclhashcat, oclhashcat+, oclhashcat-lite

BOINC & Closed Source Apps: Wrapper Apps

job.xml

```
<job_desc>
  <task>
    <application>hashcat</application>
    [ <stdin_filename>name</...> ]
    [ <stdout_filename>name</...> ]
    [ <stderr_filename>name</...> ]
    [ <command_line>--foo bar</...> ]
    [ <append_cmdline_args/> ]
  </task>
  <task>
      …
  </task>
</job_desc>
```

- Features!
  - <daemon />
  - <multi_process />
  - <setenv>
- genwrapper
  - functionally bash
  - for, while, if
  - cat, egrep, sed, awk, sort, gzip, unix2dos,...

App Plans & GPU Stuff

```
apps/
  name/
    name_version.minor_platform[.ext]

  msieve/
    msieve_148.1_linux
```

App Plans & GPU Stuff

```
apps/
  name/
    name_version.minor_platform[.ext]

  msieve/
    msieve_148.1_linux

  cudahashcat+/
    cudahashcat+_3.1_linux__cuda
```

App Plans & GPU Stuff

apps/
  name/
    name_version.minor_platform[.ext]

  msieve/
    msieve_148.1_linux

  cudahashcat+/
    cudahashcat+_3.1_linux__cuda

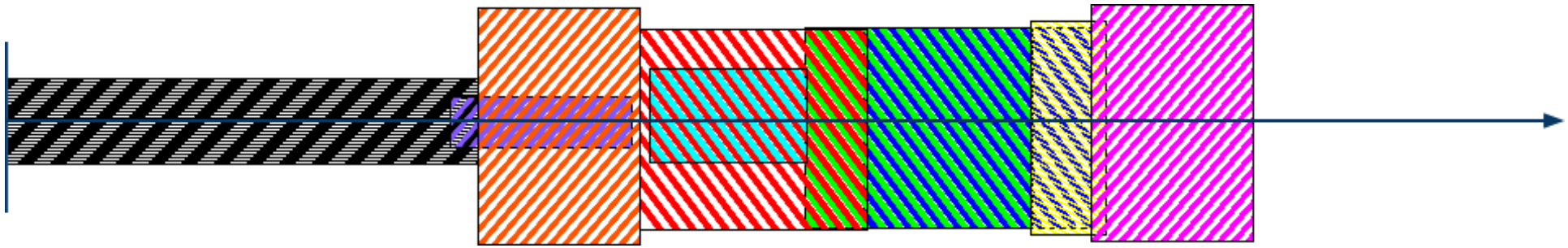
__mt - Multi-threaded
__cuda

Specific GPU Targets:
    __cuda_fermi
    __cuda_opencl
    __ati14
    ...

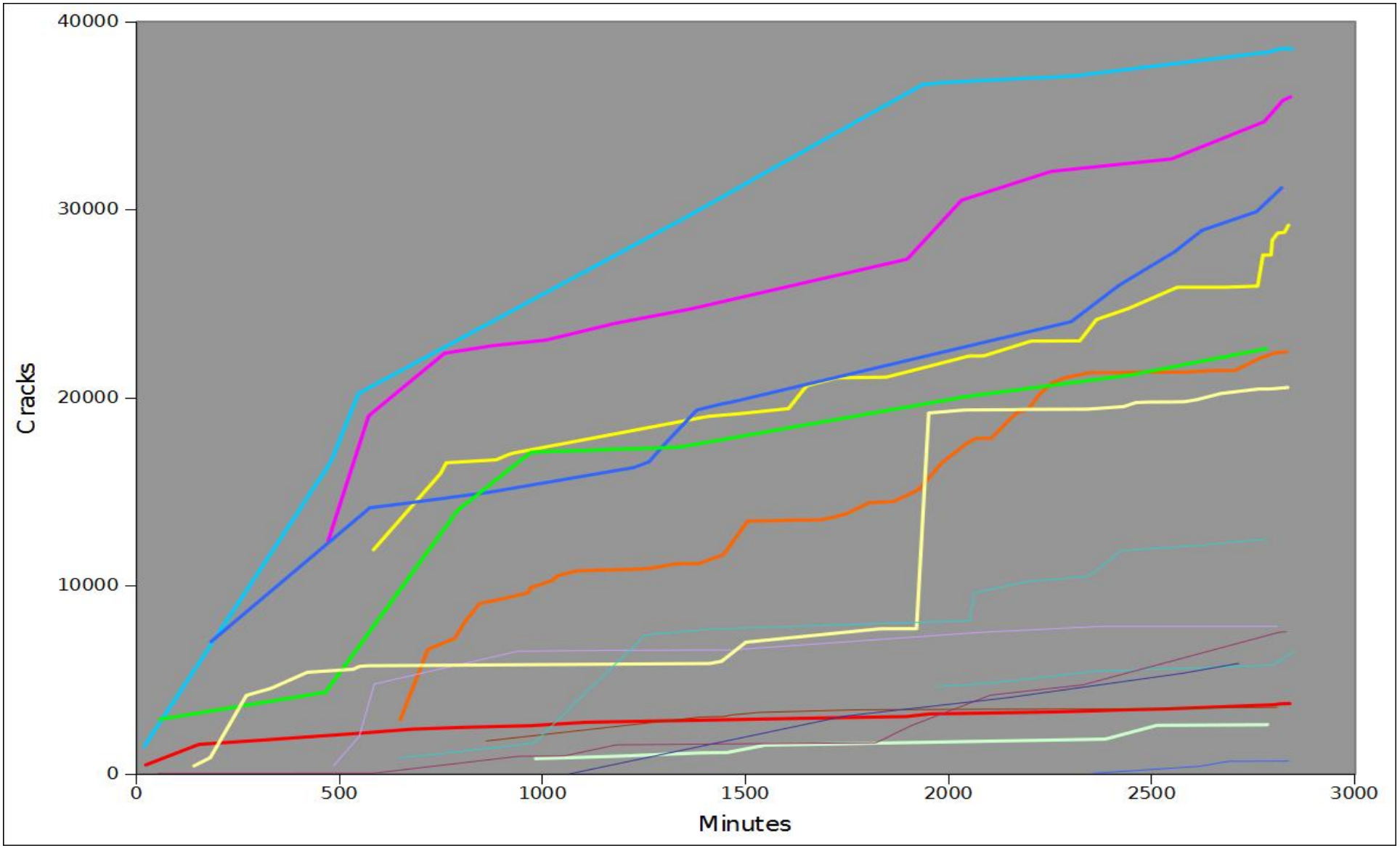
__nci - Non-CPU Intensive
__sse3
__vbox32 - VirtualBox

My Approach

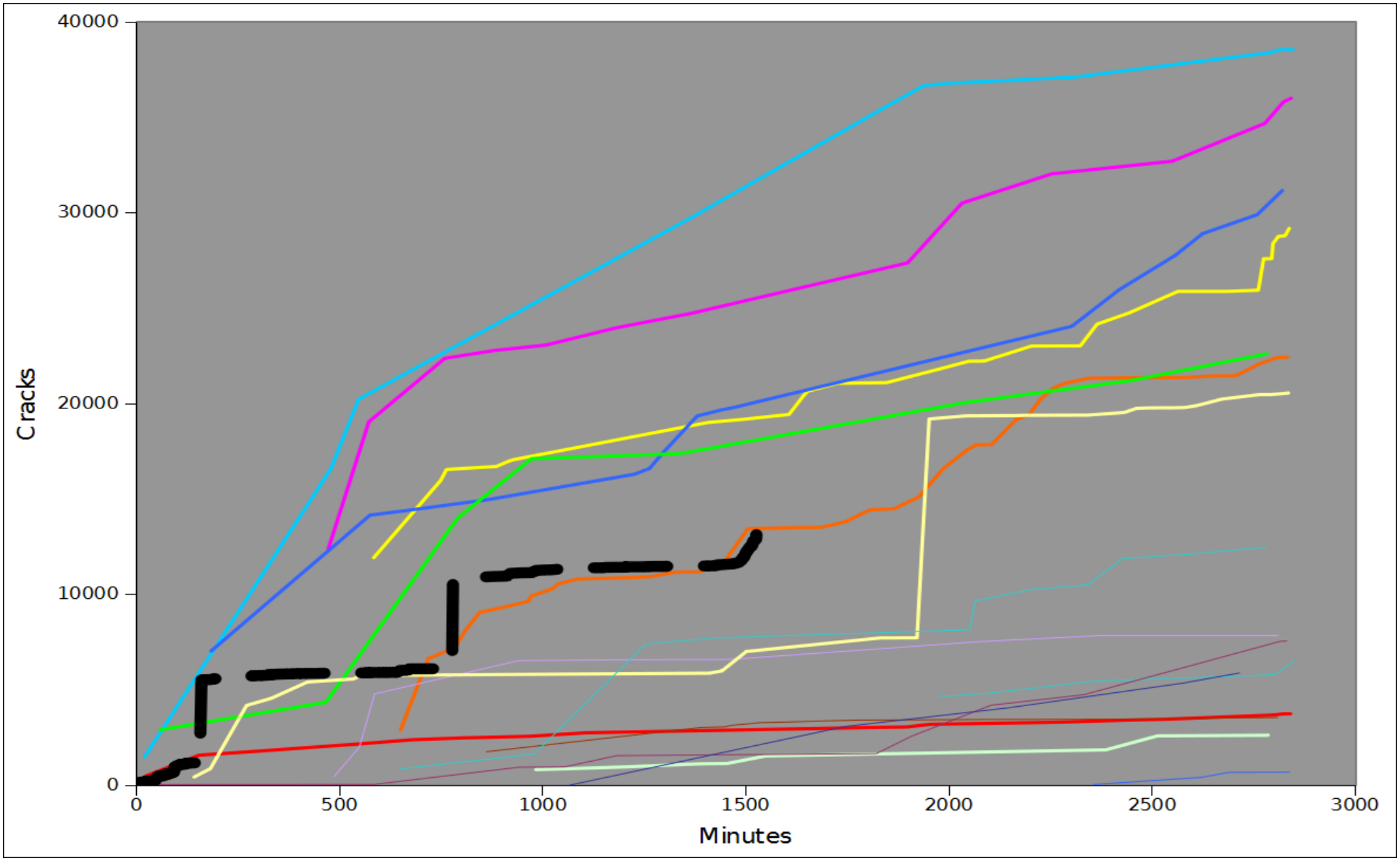
Phase 1: --single
Phase 2: 1 hour incremental
Wordlists
Phase 3: Wordlist Rules
High-Probability Markov Words
Phase 4: Phase 3 Markovs + Rules
Low-Probability Markov Words
Phase 5: Phase 4 Markovs + Rules



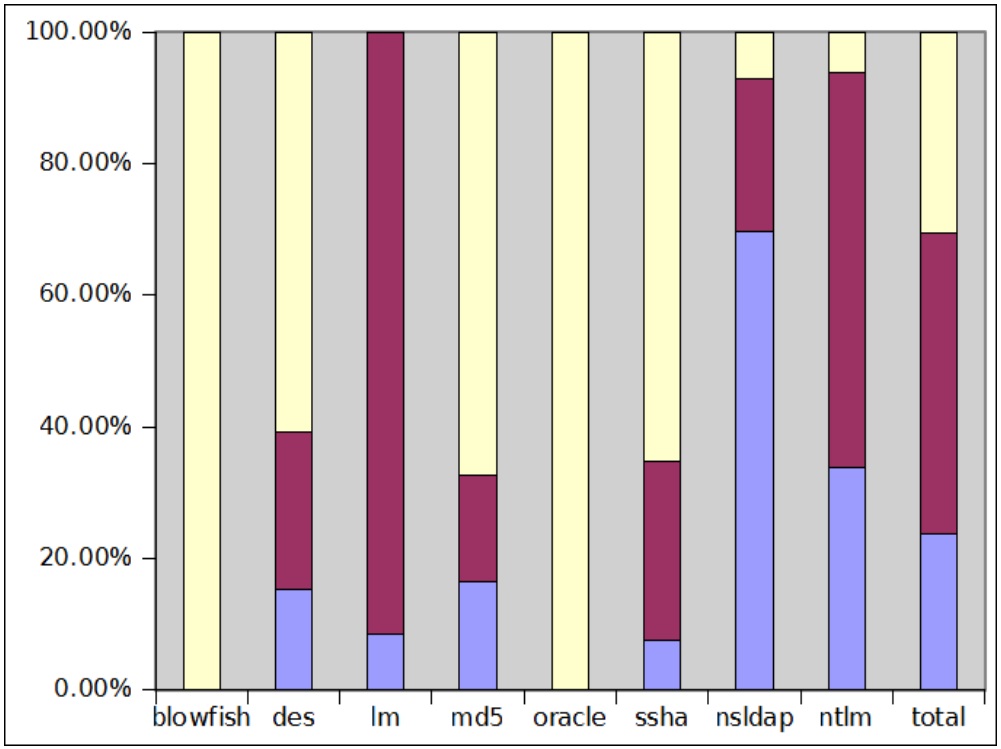
Benchmarking: 2010 Defcon Korelogic Crack Me If You Can Contest

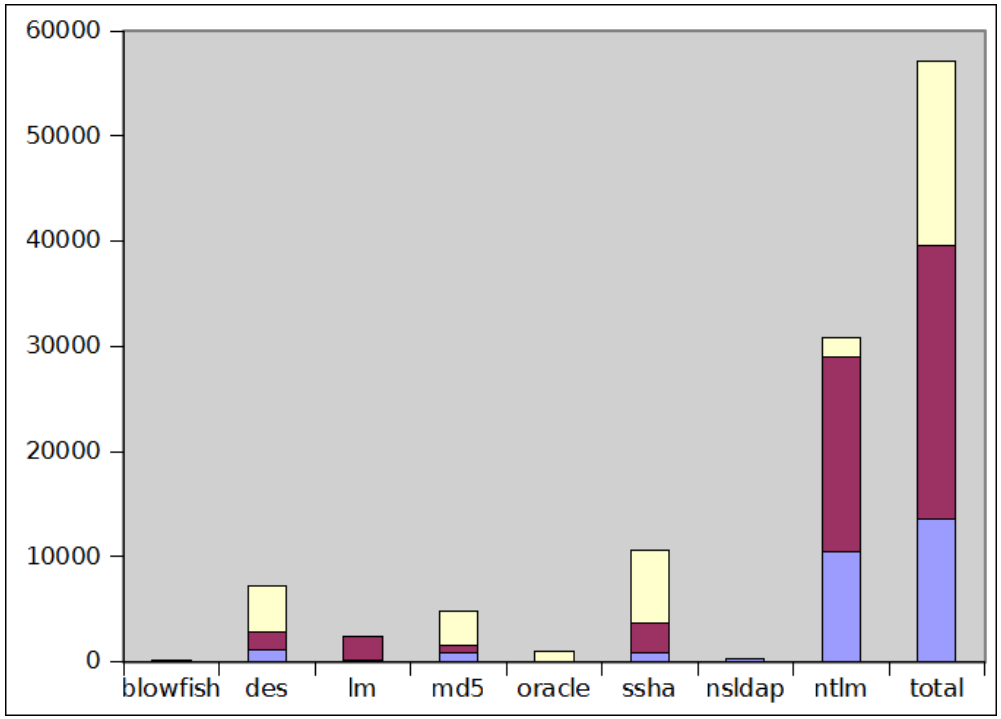


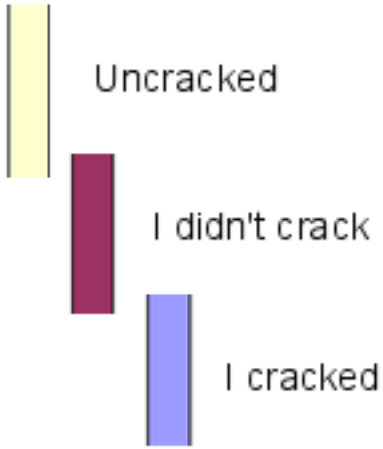
Abject Failure.

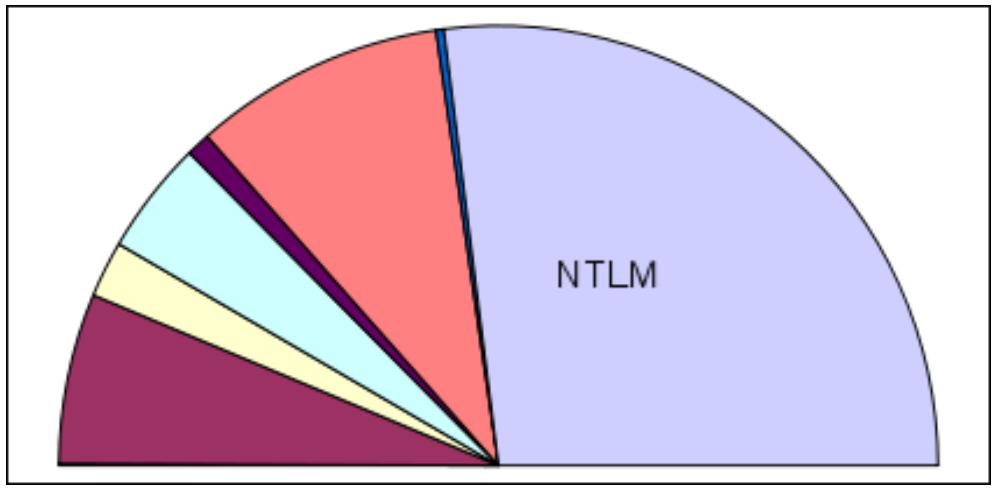


Failure by Hash Type









Lessons Learned

- Iterative Cracking

  new pattern -> maskprocessor -> rules -> cracks
  /\ \/
  new plains <- random rules <- new dic
- Automatic mangle rules creation
- Observations from cracked passwords

- Cracked password lists
- Actually Crack LM

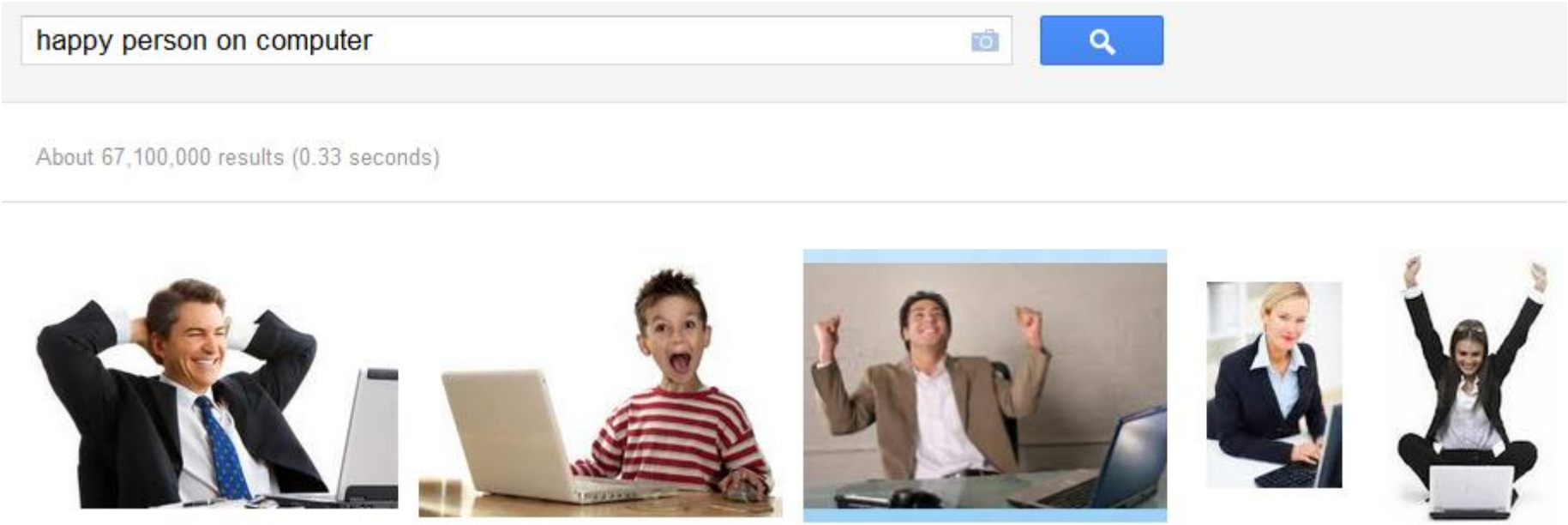
Log File Analysis

Assumptions

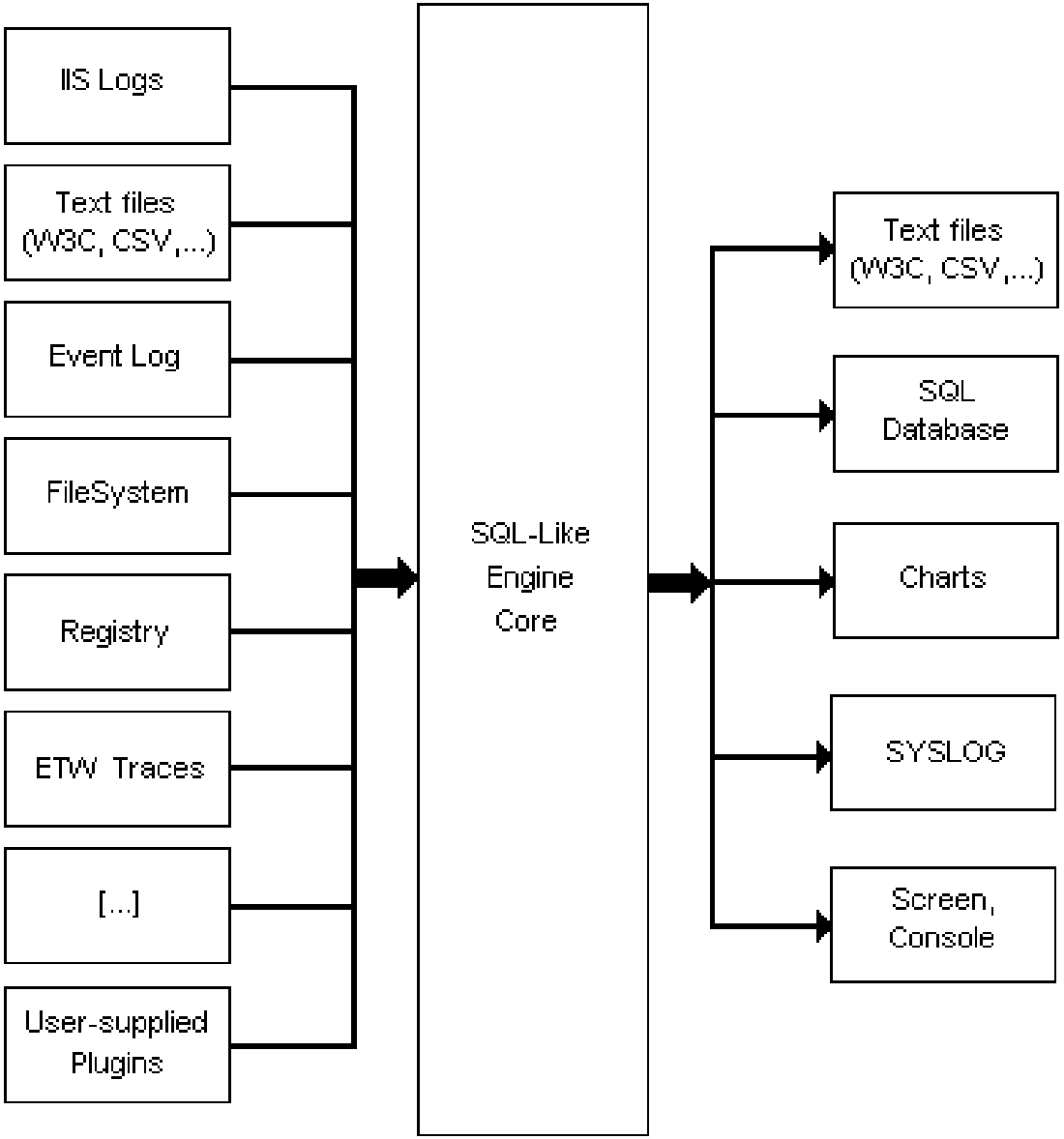
- Produce or collect a lot of data.
- Intelligent and curious people

Solution: Let them play with it

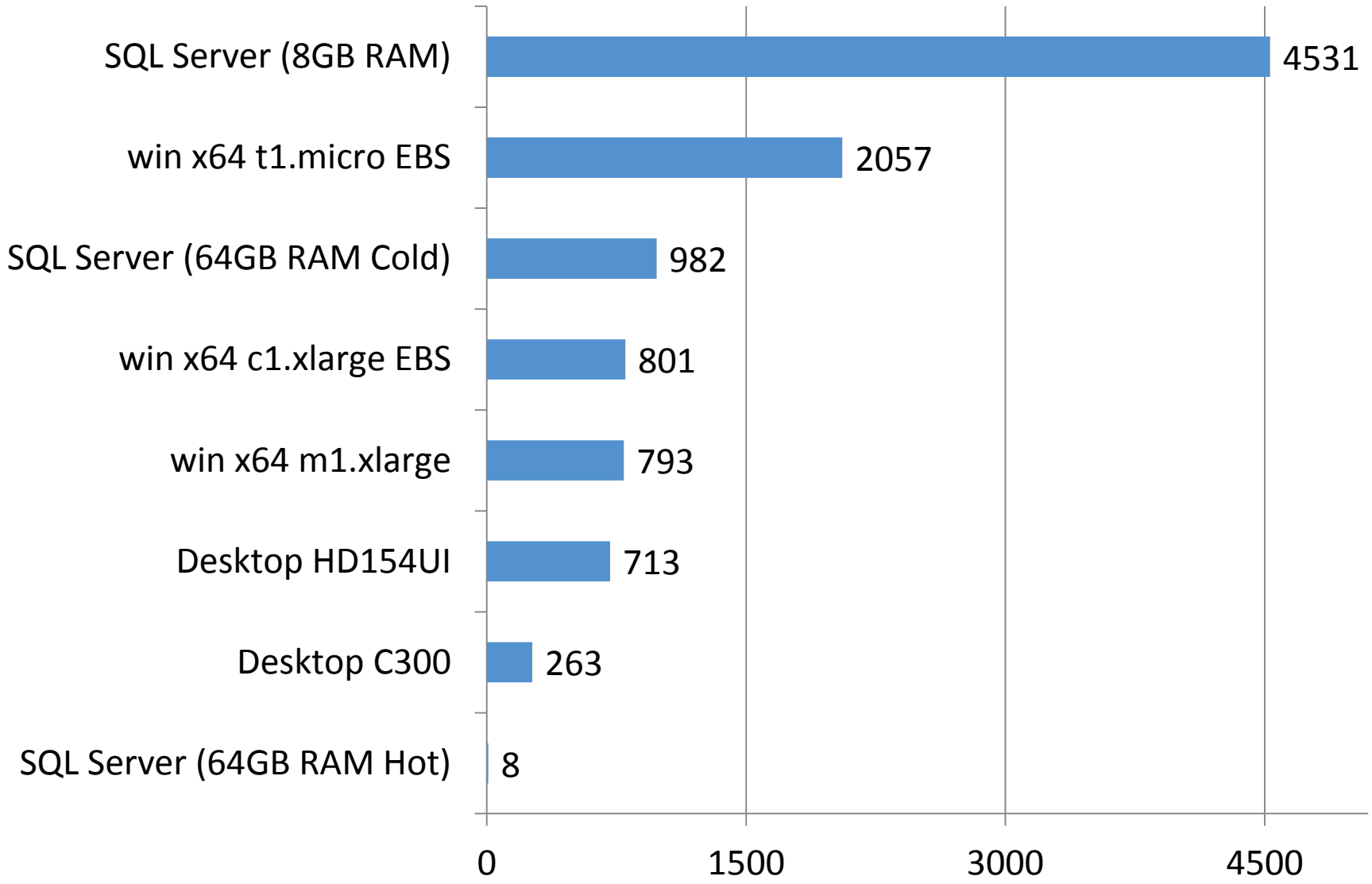
- Produce or collect a lot of data.
- Intelligent and curious people



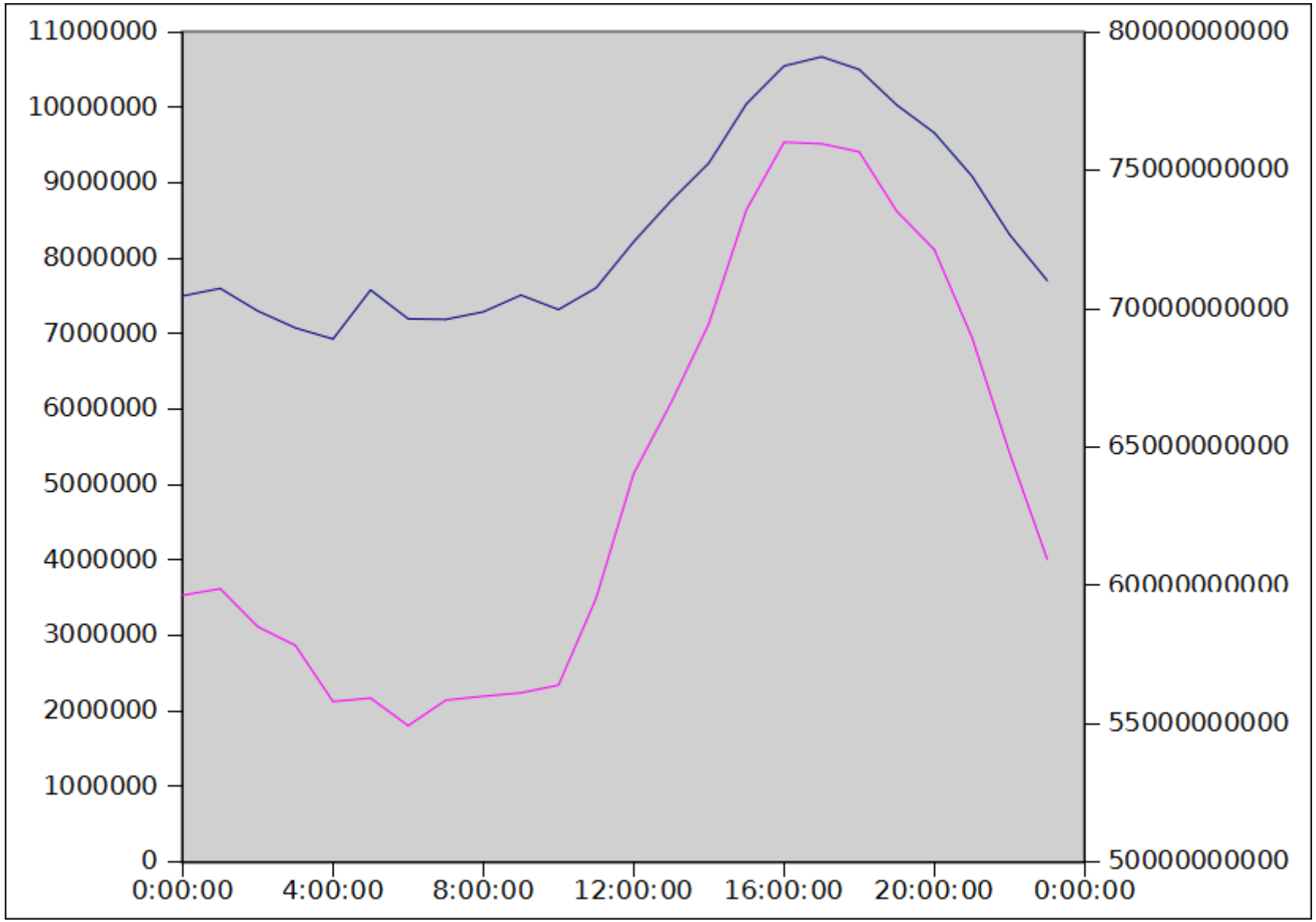
Enter Microsoft LogParser…



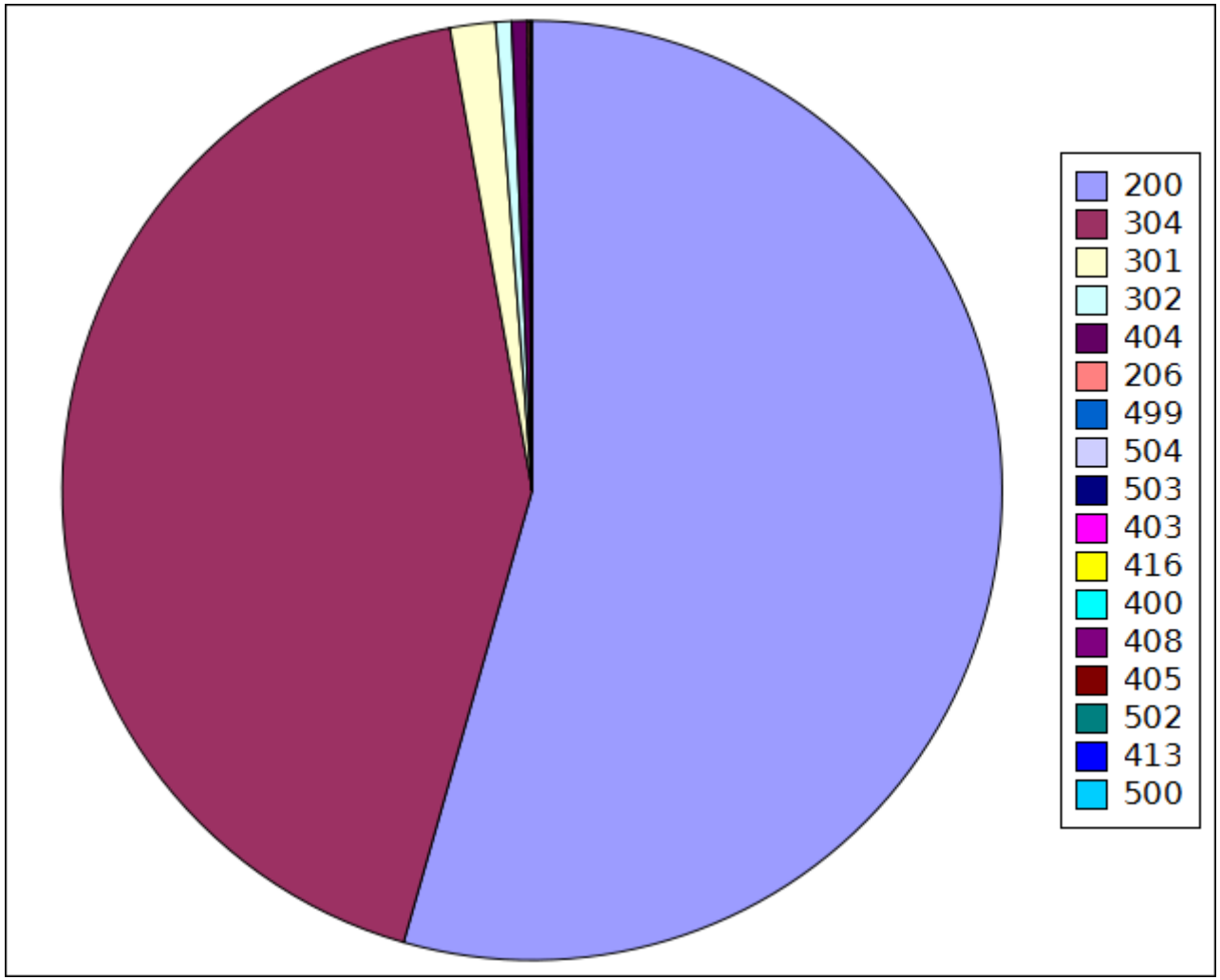
Are You Sure? Why BOINC?



What you get: Standard Stuff



What you get: Standard Stuff



What you get: Cool Stuff

	A	B	C
1	StatusCode	hits	Percentage
2	200	109340431	54.409745
3	304	85985190	42.787761
4	301	3139890	1.562465
5	302	1086009	0.540417
6	404	1071996	0.533444
7	206	179174	0.08916
8	499	125462	0.062432
9	504	19353	0.00963
10	503	4328	0.002154
11	403	3696	0.001839
12	416	936	0.000466
13	400	373	0.000186
14	408	221	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

What you get: Cool Stuff

	A	B	
1	StatusCode	hits	age
2	200	1093	.409745
3	304		42.787761
4	301		1.562465
5	302	09	0.540417
6	404	6	0.533444
7	206	179174	0.08916
8	499	125462	0.062432
9	504	19353	0.00963
10	503	4328	0.002154
11	403	3696	0.001839
12	416	936	0.000466
13	400	373	0.000186
14	408	221	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

What you get: Cool Stuff

	A	B	
1	StatusCode	hits	age
2	200	1093	.409745
3	304		42.787761
4	301		1.562465
5	302	9	0.540417
6	404	6	0.533444
7	206	179174	0.08916
8	499	125462	0.062432
9	504	19353	0.00963
10	503	4328	0.002154
11	403	3696	0.001839
12	416	936	0.000466
13	400	373	0.000186
14	408	221	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

206 Partial Content

The server is delivering only part of the resource due to a range header sent by the client. The range header is used by tools like wget to enable resuming of interrupted downloads, or split a download into multiple simultaneous streams.

What you get: Cool Stuff

	A	B	C
1	StatusCode	hits	Percentage
2	200	109340431	[illegible].409745
3	304	85985190	[illegible]87761
4	301	3139896	[illegible]465
5	302	1086[illegible]	[illegible]40417
6	404	10[illegible]	[illegible].533444
7	206	[illegible]	0.08916
8	499	[illegible]	0.062432
9	504	[illegible]53	0.00963
10	503	[illegible]8	0.002154
11	403	3696	0.001839
12	416	936	0.000466
13	400	373	0.000186
14	408	221	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

403 Forbidden

The request was a legal request, but the server is refusing to respond to it. Unlike a *401 Unauthorized* response, authenticating will make no difference.

What you get: Cool Stuff

	A	B	C
1	StatusCode	hits	Percentage
2	200	109340431	54.409745
3	304	85985190	42.787761
4	301	3139890	1.562465
5	302	1086009	540417
6	404	1071996	3444
7	206	17917	916
8	499	125	62432
9	504		0.00963
10	503		0.002154
11	403		0.001839
12	416	36	0.000466
13	400		0.000186
14	408	221	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

408 Request Timeout

The server timed out waiting for the request. According to W3 HTTP specifications: "The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time."

What you get: Cool Stuff

	A	B	C
1	StatusCode	hits	Percentage
2	200	109340431	54.409745
3	304	85985190	42.787761
4	301	3139890	1.562465
5	302	1086009	0.540417
6	404	1071996	.533444
7	206	179174	08916
8	499	125462	432
9	504	19	0963
10	503		.002154
11	403		0.001839
12	416		0.000466
13	400	73	0.000186
14	408	1	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

What you get: Cool Stuff

	A	B	C
1	StatusCode	hits	Percentage
2	200	109340431	54.409745
3	304	85985190	42.787761
4	301	3139890	1.562465
5	302	1086009	0.540417
6	404	1071996	0.533444
7	206	179174	08916
8	499	125462	432
9	504	19	0963
10	503		.002154
11	403		0.001839
12	416		0.000466
13	400	73	0.000186
14	408	1	0.00011
15	405	206	0.000103
16	502	89	4.4E−005
17	413	86	4.3E−005
18	500	2	1E−006
19			

405 Method Not Allowed

A request was made of a resource using a request method not supported by that resource; for example, using GET on a form which requires data to be presented via POST, or using PUT on a read-only resource.

Let’s look at those 405s…

RemoteHost	DateTime	Request	St	Byte	User-Agent
118.96.132.212	6/13/11 18:49	PUT /showthread.php?t=30284	200	26679	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212	6/14/11 1:18	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212	6/19/11 5:53	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212	7/28/11 4:26	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212	9/5/11 14:55	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212	10/4/11 2:39	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139	6/30/11 20:59	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139	7/17/11 18:35	PUT /showthread.php?t=44038	200	62963	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139	7/29/11 8:18	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139	8/18/11 5:15	PUT /forumdisplay.php?f=66/i	200	164048	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	6/11/11 3:10	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	6/14/11 23:18	PUT /showthread.php?t=10888	200	18086	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	6/30/11 3:25	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	6/30/11 3:25	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	8/11/11 14:48	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	9/16/11 14:42	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	10/8/11 1:21	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.83	8/19/11 16:37	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.83	9/2/11 14:26	PUT /indonesia.htm HTTP/1.0	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.90	6/14/11 17:18	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.90	7/9/11 9:20	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.90	9/17/11 8:12	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
124.82.33.83	9/16/11 10:52	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.5
180.241.133.53	8/20/11 2:03	PUT /indonesia.txt HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET4.0

Let’s look at those 405s…

RemoteHost	DateTime	Request	St	Byte	User-Agent
118.96.132.212	6/13/11 18:49	PUT /showthread.php?t=30284	200	26679	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212	6/14/11 1:18	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.132.212			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139	6		405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139	7		200	62963	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.139			200	164048	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	6		200	18086	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	8		405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63	9		405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.63			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.83	8		405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.83			405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.90	6		405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.90	7/9/11 9:20	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
118.96.133.90	9/17/11 8:12	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2)
124.82.33.83	9/16/11 10:52	PUT /indonesia.htm HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.5
180.241.133.53	8/20/11 2:03	PUT /indonesia.txt HTTP/1.1	405	533	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET4.0

118.96.132.212
118.96.133.139 6
118.96.133.139 7
118.96.133.139
118.96.133.139
118.96.133.63
118.96.133.63 6

Let’s look at those 405s…

Subnets of: 118.96.0.0 - 118.97.255.255

IP-Range	Netname	Orgname	C
118.96.128.0 - 118.96.191.255	TLKM_D2_BB_SPEEDY_JKT	PT TELKOM INDON	
118.96.192.0 - 118.96.239.255	TLKM_D2_BB_SPEEDY_CRB	PT TEL INDON	
118.97.14.0 - 118.97.14.255	TLKM_D4_AST_CUSTOMER	PT Telk Indone custon	
118.97.20.0 - 118.97.20.255	TLKM_D4_AST_CUSTOMER	PT Telk Indone custon	
118.97.105.0 - 118.97.205.255	TLKM_D4_AST_CUSTOMER	PT Telk Indone custon	
118.97.224.0 - 118.97.255.255	TLKM_D5_AST_CUSTOMER	PT Telk Indonesia's customer.	ID

PT TELKOM INDONESIA

PT TELKOM INDONESIA

PT Telkom Indonesia's customer.

http://oucsace.cs.ohiou.edu/~tysko/scanattack.2010.09

```
against us for Sep. 2010

me to keep the list up all the time
ng scans/attacks, or 1 day samples are here

ame, if any) attack/scan/notes

cked by Hmei7" via "PUT /indonesia.htm HTTP/1.0" against132.235
U for ports 8088, 8085, 8080, 808, 80, 3124, 3129, 3128, 3127
te force attace login=webohiou, ohiouweb
an net for telnet port
cked by Hmei7" via "PUT /indonesia.htm HTTP/1.0" against 132.23
Hacked by Hmei7" via "PUT /indonesia.htm HTTP/1.0" against 132
ning our net for ports 808,80,8888,8080,3128,2301,8000
p brute force passwd attack on 132.235.1.2 user=ohiou, ohiouedu,ace
p brute force passwd attack on 132.235.1.2 user=ohiou,ohiouedu
9 probe ports 80,8080,3128 on 132.235.1.53
```

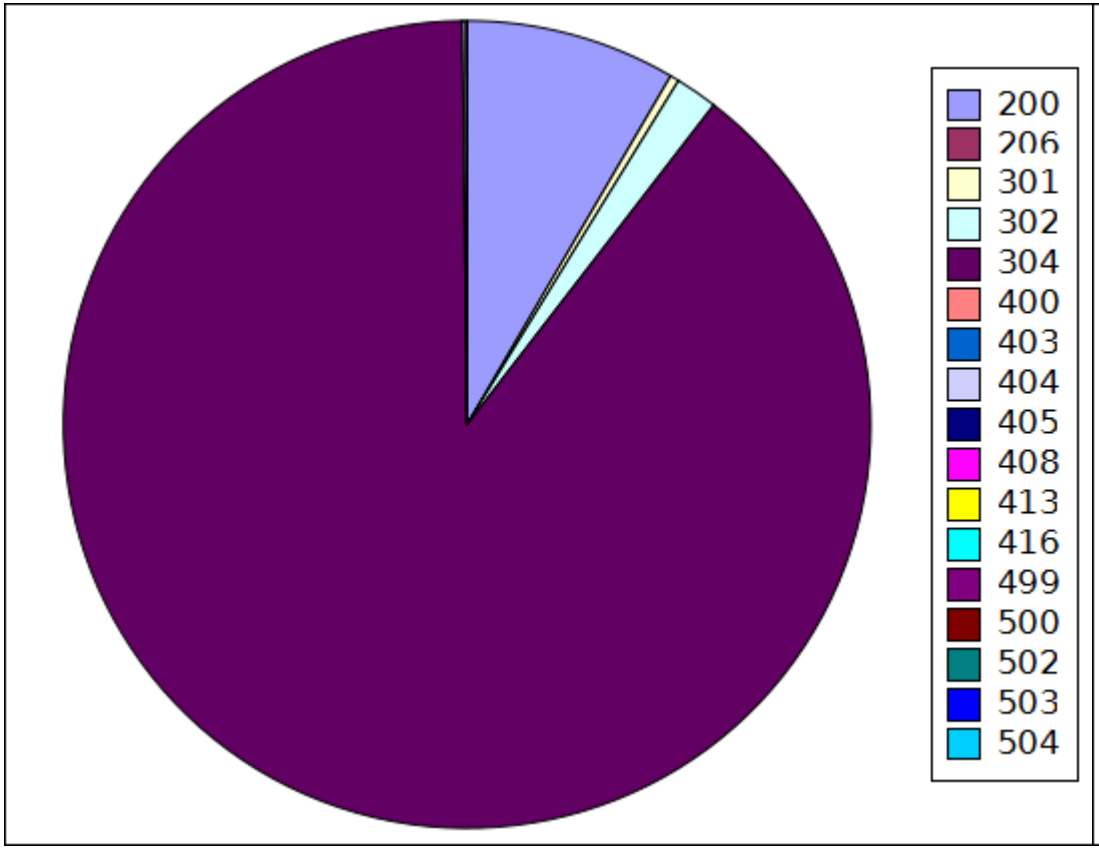
"Hacked by Hmei7" via "PUT /indonesia.htm HTTP/1.0"

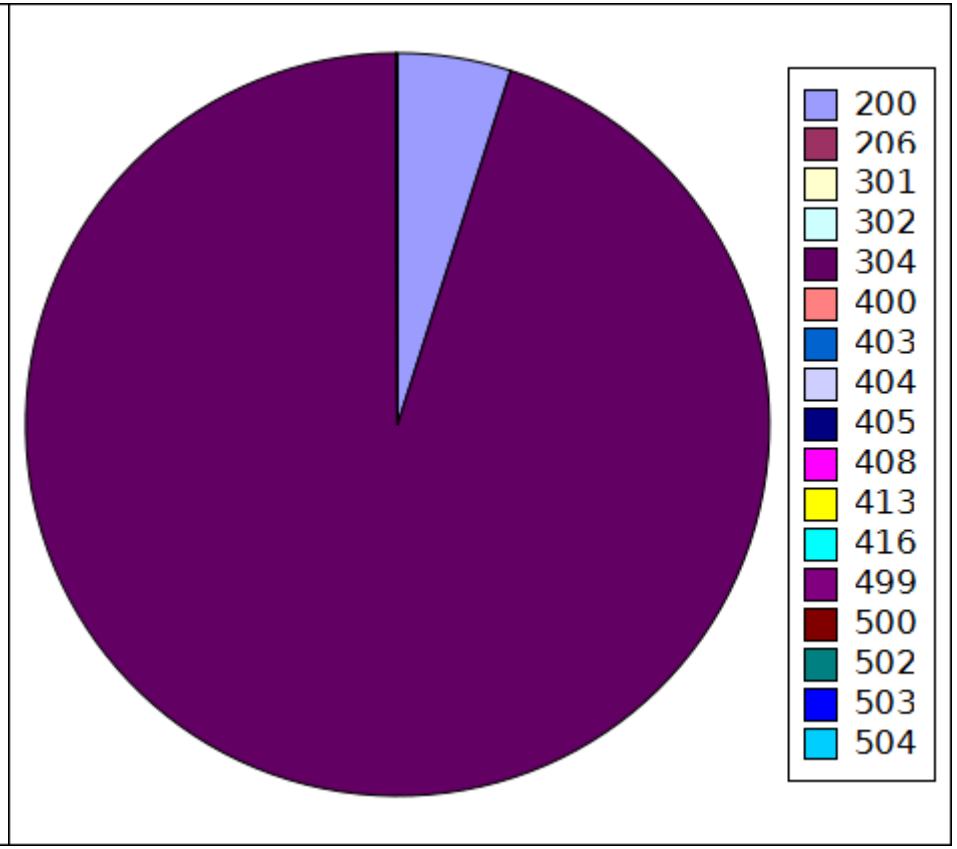
```
scan 132.235.1.165 for ports 8888,8080,80,3128,8415,443,8085
7 scan 132.235.1.249 for ports 8888,8080,80,3128,8415,443,8085
```

What you get: Cool Stuff

Status Code	Average	StDev
200	84.06655	29.836096
206	0.352751	3.98607
301	0.246852	2.972122
302	0.117694	1.299405
304	10.886761	24.349427
400	0.000357	0.155511
403	0.000959	0.092487
404	3.746868	18.036991
405	0.001214	0.322419
408	0.001991	0.406084
413	3E−006	0.001439
416	0.001405	0.285774
499	0.521032	6.503026
500	0	1.5E−005
502	0.000151	0.110485
503	0.008067	0.658489
504	0.046797	1.823632

What you get: Cool Stuff





Wrap Things Up

Back to BOINC

1. Set up a BOINC Server
2. Edit config.xml
3. Lock down the server
4. Set up a client image
5. Set up an application
6. Automate the client image
7. ???
8. Profit!

1. Patch source

   or

1. Write job.xml
2. Write input & output templates
3. update_versions
4. Create test workunits
5. Test
6. Repeat 1-6 as needed

Alternatives to BOINC

Password Cracking Only
- Browser Based using Javascript / AJAX / Web Workers
- Durandal http://durandal-project.org/
- Rick Redman of Korelogic's tool

General Architecture
- Amazon Elastic Beanstalk (Java-only)
- Amazon SQS (Write your own wrapper and uploader)
- Bash Scripts/tentakel/multixterm/cssh
- Write your own?
  Will that take more or less than time than configuring BOINC?
  I think more.

Questions?

Big Ups To:

- Brian Holyfield & Joe Hemler
- jasonp

Thanks:

- iSEC Partners
- Gotham Digital Science
- MersenneForum & jasonp

Tom Ritter

http://www.isecpartners.com/
https://github.com/tomrittervg/cloud-and-control