

Low Fat Grid

A RESTful Grid Service for Non-Programmers

Jason Brazile

Remote Sensing Labs, University of Zurich / Netcetera

180

JAZOON07

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 24 - 28, 2007 ZURICH

netcetera

Quality
Software
Engineering



Problem: Repetitious; pesky IP; tool complexity

- > **Workload: repetitive program runs**
 - Permuting 6 variables, each with 5 values = 15,625 runs
 - Field: 3000 scan lines @350 wavelengths = 1,050,000 runs
- > Sometimes ***IP restricted*** software or observations (i.e. input data)
- > **End-users**, not professional software developers
 - “Programming? Sure, I know Fortran ***and*** BASIC”

Trend: CPU:User ratios going up

	#hosts on network	#active users	host/user ratio
Medium Sized IT Firm	1356	~100	~14
University Department (Natural Sciences)	387	~150	~3

What's Out There?

- > Condor
 - Prolific, Widely-used
 - Setup more complicated than need be
 - Not internet (WAN) friendly
- > Apple's Xgrid
 - Nice GUI
 - Apple only
 - WAN-friendliness requires port 4111 to not be blocked
- > WSRF::Lite
 - First(?) grid as web service
 - Unfortunately WS-* instead of REST (i.e. more complex)

DIY Goals: Simple, Distributed, Cycle Scavenging

Simple...

- > Use only what “everyman” has access to
- > Require no professional programming skills
- > Require no admin rights or network gurus

Distributed...

- > 3 PhDs share my weird problem but are in 3 other countries

Cycle Scavenging...

- > My wife’s laptop is faster than mine
- > My office mates are on spring vacation for two weeks
- > My colleague's research cluster is free this weekend

Design Motivations

Simple...

- > Most workstations can browse the web (proxy?), but no other ports
- > Commodity ISP hosting offers CGI, but not servlets/app servers
- > Hence: http as transport, scheduling via CGI, simple mimetypes

Distributed...

- > http is the one and only transport
- > Web service (Mashup-able)
- > Scheduling via pull from clients (self-balancing)

Cycle Scavenging...

- > Friends/colleagues download a single jar file
- > Maybe run an instance for each CPU (core)

Web service: Why REST instead of WS-* ?

- > Easy to program
- > No external library dependencies (for deployment)
- > Easy to test (`curl -X PUT -T data.file http://isp.com/lfg`)
- > Easy to mashup (*mapping* instead of *wrapping*)
- > Don't have to use XML
 - Some payloads are `text/html`, others `application/x-tar`
- > “Leverages” standard web stuff (caches, security, etc)

On a tangent: What REST Email would look like

- > Resources (URIs) and actions instead of RCP calls
 - POST content on sender's outgoing URI
 - POST to recipient's incoming_message URI as notification
 - recipient GET's content from sender's newly created message URI
- > Mapping instead of wrapping
 - Actions are limited to GET/PUT/POST/DELETE (+LOCK/MOVE)
 - Think: Unix's create/read/write/open/close + everything's a file
- > Use appropriate `mimetype` instead of always XML

(see Paul Precod's complete example <http://www.precod.net/rest/restmail>)

Example: REST Email delivery

Intention	Request	Response
Create new message	POST sender.com/outgoing Content-type: text/plain	
		202 Accepted Location: sender.com/outgoing/19682 A new resource has been allocated
Alert Recipient	POST recipient.com/incoming Content-type: text/xml+rss <notification> <item><desc>new msg</desc> <url> sender.com/outgonig/19682 </url></item> </notification>	
		202 Accepted Location: recipeient.com/incoming/78979 Notification stored until retrieved

Summary of REST benefits

- > Re-use of standard server-side tools like apache, squid, search, mgmt
 - website mgmt tools become mailbox mgmt tools “for free”
 - every web browser becomes mailbox browser “for free”
- > email namespace == web namespace
 - web pages can refer to messages/mailbox & vice-versa
 - Individuals wouldn't need separate mailto:/home page URLs
 - HTTP features available to mail “for free” (security, caching, reliability)
- > Mail User Agent only uses one protocol, not 3 (SMTP/POP/IMAP)

(see Paul Precod's complete example <http://www.prescod.net/rest/restmail>)

Grid Service Resources and Actions

Resource	Action	Behavior
<code>batches</code>	GET	Retrieve list of batches with status
	POST	Request URI creation for next runnable job (any batch)
<code>batches.tar</code>	GET	Download archive of all batches
	POST	Request URI creation for new batch
<code>batches/NNNNN</code>	GET	Retrieve list of jobs for batch NNNNN with status
	POST	Request URI creation for next runnable job (for batch NNNNN)
<code>batches/NNNNN.tar</code>	GET	Download archive of batch NNNNN
<code>batches/NNNNN/MMMMM</code>	GET	<code>getJobStatus()</code>
<code>batches/NNNNN/MMMMM.tar</code>	GET	Download archive of job MMMMM
	PUT	Upload results for job MMMMM

Example Message Exchange: I

Intention	Request	Response
Upload new batch	POST batches.tar Content-type: application/x-tar	
		202 Accepted Location: batches/19682 A new batch has been allocated
Alternative discovery	GET batches	
		200 OK Content-Type: text/html ... Batch Name

Example Message Exchange: II

Intention	Request	Response
Request next job	POST batches/19682 Content-type: text/plain ...	
		202 Accepted Location: batches/19682/00001 A new job has been allocated
Get it	GET batches/19682/00001.tar	
		200 OK Content-Type: application/x-tar ...

Example Message Exchange: III

Intention	Request	Response
Upload processing results	POST batches/19682/00001.tar Content-type: application/x-tar ...	
		202 Accepted Location: batches/19682/00001.tar Job has been committed
Get batch	GET batches/19682.tar	
		200 OK Content-Type: application/x-tar ...

ATCOR Atmospheric Correction Software



IKONOS* Raw Image



**IKONOS* Image after
ATCOR Haze Reduction**

*Data Courtesy of European Space Imaging © European Space Imaging GmbH

JAZZON07

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 24 - 28, 2007 ZURICH

netcetera

Quality
Software
Engineering

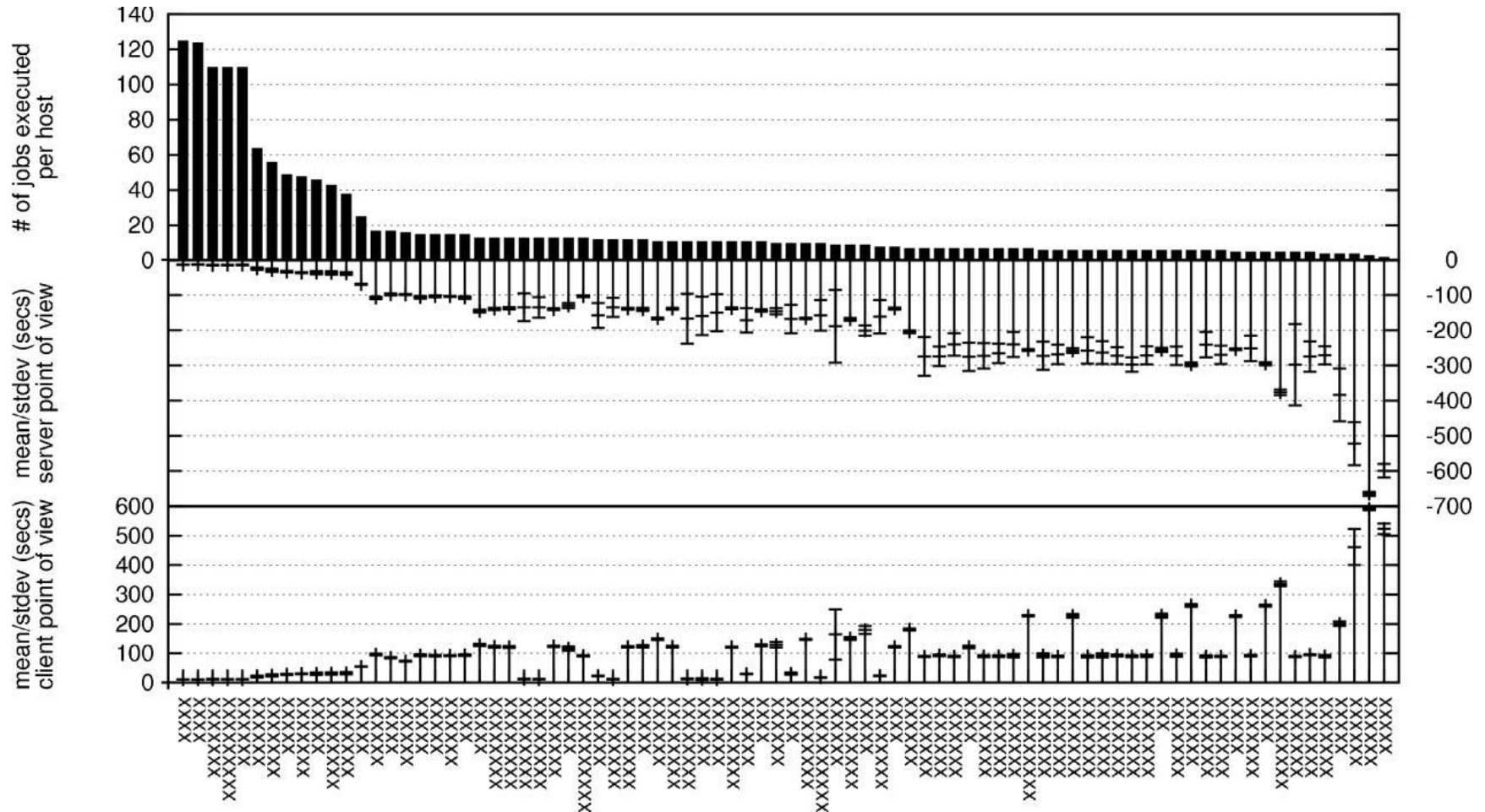


Existing Applications

- > Atmospheric Modeling
 - 1,500 -> 64,000 executions
 - Individual execution <1min on fastest machines
 - Input file size ~3K
 - Output file size ~1Mb
- > Earth Observation Satellite Mission Simulation
 - Multiple universities/research institutions, multiple countries
 - One software component was IP restricted (state secret)
 - One set of observations were IP restricted (still too much recency value)
 - However local execution allowed, results can be exported

Distributed processing not just for performance!

Reality of (lack of) heterogeneous speedup



Other Hypothetical Applications

- > Home Movie Render Farm (repetitive, CPU intensive)
 - Distributed pov-ray for single frame rendering
 - Distributed ffmpeg for MPEG encoding
- > Taxes/Bills paper digitizing/archiving (IP restrictive)
 - One friend: license for OCR -> single page MS word
 - Another friend: license for MS word -> pdf
 - Another friend: license for pdf concatenation, indexing, etc

Summary

- > Easy to use
 - create archive of subdirs, each with sh-like command file
- > Simple to deploy
 - server: single CGI file
 - client: single .jar
- > Better latency performance than Condor
- > RESTful web service
 - “leverage” standard web stuff (performance, caching, security)
 - mashup-able
- > Easy way to slow down wife’s previously much faster laptop **J**

Availability

code.google.com/p/lowfatgrid

JAZOON07

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 24 - 28, 2007 ZÜRICH

netcetera

Quality
Software
Engineering



References/Links

Apple Xgrid: <http://www.apple.com/acg/xgrid>

B. Beckles, S.C. Son, J. Kewley (2005) *Current methods for negotiating firewalls for the Condor System*, Proceedings 4th UK e-Science All Hands Meeting

J. Brazile, R. Richter, D. Schlaepfer, M.E. Schaepman, K.I. Itten (2007) *Cluster versus grid for operational generation of ATCOR's MODTRAN-based lookup tables*. Parallel Computing, in review.

S. Dangel, F. Petitcolin, J. Brazile, C. Miesch, L. Poutier, L.Jia, J. Moreno, M. Gloor, M. Kneubuehler, M.E. Schaepman, K.I. Itten (2005) *SPECTRA end-to-end simulator: Final Report* ESA/ESTEC, CR(P)-4492

R.T. Fielding (2000) *Architectural Styles and the Design of Network-based Software Architectures*, PhD Thesis, University of California Irvine.

M. Litzgow, M. Livny, M. Mutka (1988) *Condor, a hunter of idle workstations*, in The 8th International Conference on Distributed Computing Systems, pages 104-111

Jason Brazile

<http://www.geo.uzh.ch/~jbrazile>

Remote Sensing Labs, Univ. of Zurich
Netcetera

jbrazile@geo.uzh.ch

jason.brazile@netcetera.ch

JAZOON07

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 24 - 28, 2007 ZURICH

netcetera

Quality
Software
Engineering

