

# Programming the Cloud: the Internet as Platform



Gregor Hohpe  
Software Engineer  
[www.EnterpriseIntegrationPatterns.com](http://www.EnterpriseIntegrationPatterns.com)



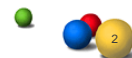
© 2008 Google, Inc. All rights reserved.

## Who's Gregor?

- Distributed systems, asynchronous messaging, service-oriented architectures
- MQ, MSMQ, JMS, ESB's
- Software engineer at Google
- Book: *Enterprise Integration Patterns*
- Site: [www.eaipatterns.com](http://www.eaipatterns.com)
- Write code every day. Share knowledge through patterns.
- "Starbucks does not use 2-phase commit" featured in Joel Spolsky's *Best Software Writing*.



© 2008 Google, Inc. All rights reserved.



## Internet as a Platform: The Good



Falling cost of  
storage and  
computing  
power



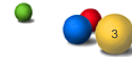
Ubiquitous  
broadband  
connectivity



Democratized  
tools of  
production



© 2008 Google, Inc. All rights reserved.



## Internet as a Platform: The Challenges

### Architect's Dream



- Loosely coupled
- Extensible
- Standards-based
- Fault tolerant
- Unlimited computing power
- Ubiquitous

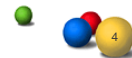
### Developer's Nightmare



- NO Call Stack
- NO Transactions
- NO Promises
- NO Certainty
- NO Ordering Constraints



© 2008 Google, Inc. All rights reserved.



## Isn't This What Distributed Transactions Are For?

- Require coordinator
- Even 2 Phase Commit has windows of uncertainty
- Not practical for long running interactions
  - Locks not practical / economical
  - Isolation not possible / practical
- Usually not supported
- Don't scale

"Life Beyond Distributed Transactions –  
an Apostate's Opinion"  
--Pat Helland



© 2008 Google, Inc. All rights reserved.

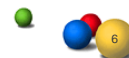


## Still An Issue With HTTP

- Hardware failure
- Network failure
- Time-outs
- Partial response



© 2008 Google, Inc. All rights reserved.



## New Game Rules

### ACID (before)

- Atomic
- Consistent
- Isolated
- Durable

Predictive  
Accurate

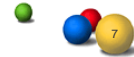
### ACID (today)

- Associative
- Commutative
- Idempotent
- Distributed

Flexible  
Redundant



© 2008 Google, Inc. All rights reserved.



## Starbucks Does not Use 2-Phase Commit Either

- Start making coffee before customer pays
- Reduces latency
- What happens if...

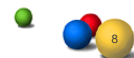
Customer rejects drink → Remake drink  
Retry

Coffee maker breaks → Refund money  
Compensation

Customer cannot pay → Discard beverage  
Write-off



© 2008 Google, Inc. All rights reserved.

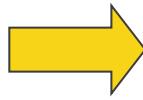


## Programming the Cloud – The Google Way

- Fault tolerant distributed storage: Google File System
- Distributed shared memory: Bigtable
- New programming abstractions: MapReduce
- Domain Specific Languages: Sawzall



Google.stanford.edu (Circa 1997)



Current Rack Design

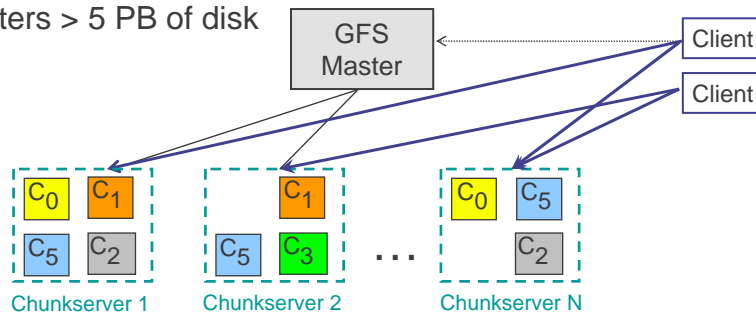


© 2008 Google, Inc. All rights reserved.



## Fault Tolerant Distributed Disk Storage: GFS

- Data replicated 3 times. Upon failure, software re-replicates.
- Master: Manages file metadata. Chunk size 64 MB.
- Optimized for high-bandwidth sequential read / writes
- Clusters > 5 PB of disk



<http://research.google.com/archive/gfs-sosp2003.pdf>



© 2008 Google, Inc. All rights reserved.



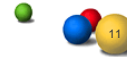
## Distributed Shared Memory: Bigtable

- Sparse, distributed, persistent, multidimensional, sorted
- Not a relational database (RDBMS): no schema, no joins, no foreign key constraints, no multi-row transactions
- Each row can have any number of columns, similar to a dictionary data structure for each row.
- Basic data types: string, counter, byte array
- Accessed by row key, column name, timestamp
- Data split into tablets for replication
- Largest cells are > 700TB

<http://research.google.com/archive/bigtable-osdi06.pdf>



© 2008 Google, Inc. All rights reserved.

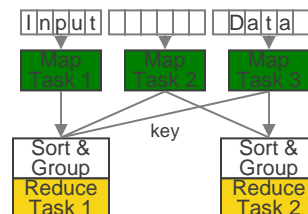


## Programming Abstraction: MapReduce

- Represent problems as Map and Reduce step (inspired by functional programming)
- Distribute data among many machines, execute same computation at each machine on its dataset
- Infrastructure manages parallel execution
- Open source implementation: Hadoop

```
map(i n_key, data)
  → list(key, value)

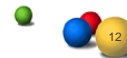
reduce(key, list(values))
  → list(out_data)
```



<http://research.google.com/archive/mapreduce.html>

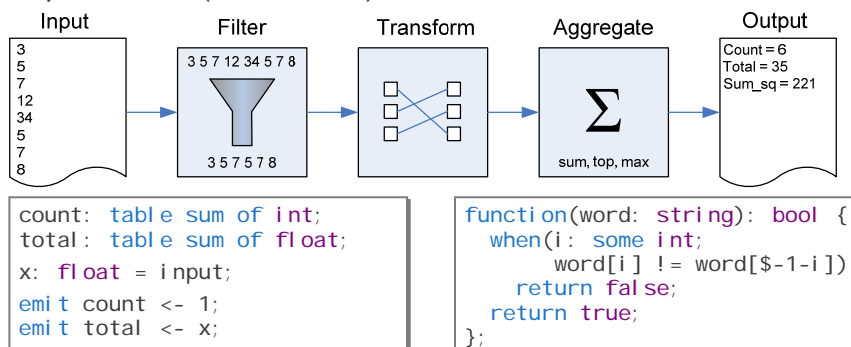


© 2008 Google, Inc. All rights reserved.



## Language for Parallel Log Processing: Sawzall

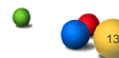
- Commutative and associative operations allow parallel execution and aggregation
- Language avoids specifying order by replacing loops with quantifiers (constraints)



<http://labs.google.com/papers/sawzall.html>



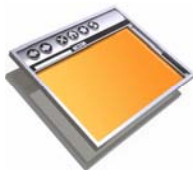
© 2008 Google, Inc. All rights reserved.



## Google, the Cloud, and You!



Make the cloud more **accessible**



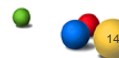
Make the client more **powerful**



Keep connectivity **pervasive**



© 2008 Google, Inc. All rights reserved.



# Google Data API's



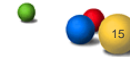
- Standard protocol for reading and writing data on the web
- Based on Atom 1.0 and RSS 2.0 syndication formats, Google Data extensions
- Atom Publishing Protocol
- Optimistic concurrency based on version numbers: no locks
- AuthSub authentication scheme: no stored passwords

- Google Apps
- Google Base
- Blogger
- Google Calendar
- Google Code Search
- Google Contacts
- Google Health
- Google Notebook
- Google Spreadsheets
- Picasa Web Albums
- Google Documents
- YouTube
- ...

<http://code.google.com/apis>



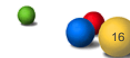
© 2008 Google, Inc. All rights reserved.



# Simple Example: Google Calendar Feed



© 2008 Google, Inc. All rights reserved.





## Calendar Feed



```
<entry>
<id>http://www.google.com/calendar/feeds/...</id>
<published>2007-08-19T19:29:25.000Z</published>
<updated>2007-09-28T17:56:20.000Z</updated>
<name>Gregor's Conferences</name>
<gd:comments>
  <gd:feedLink href='http://www.google.com/calendar/feeds/...'/>
</gd:comments>
<gd:eventStatus value='http://schemas.google.com/g/2005#event.confirmed'/>
<gd:transparency value='http://schemas.google.com/g/2005#event.transparent'/>
<gd:when startTime='2007-10-23' endTime='2007-10-27'/>
<gd:who rel='http://schemas.google.com/g/2005#event.organizer'
  valueString='Gregor&apos;s Conferences'
  email='...@group.calendar.google.com'/>
<gd:where valueString='Keystone Resort, Colorado'/>
</entry>
```



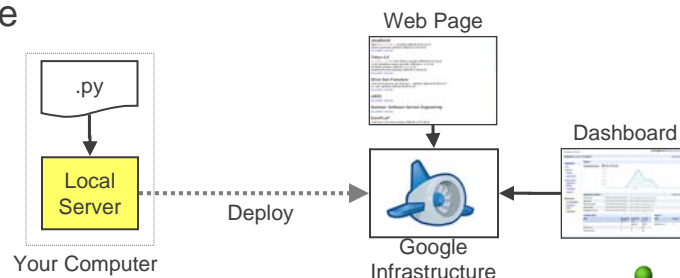
© 2008 Google, Inc. All rights reserved.



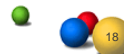
## Google App Engine – Easy to Start, Easy to Scale



- Your code on Google infrastructure
- Python source code and run-time
- Develop locally, deploy to Cloud
- Write once, scale automatically
- Free quota of 5M pageviews / month and 500MB storage



© 2008 Google, Inc. All rights reserved.



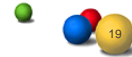
## Programming & Run-time Model



- Responds to HTTP requests
- A programming platform, not “raw iron”
- API support for
  - User login and identity
  - Persistent state (on top of Bigtable, not RDBMS)
  - memcache
  - Mail, Images, URL Fetch
  - Python libraries (not native code)
  - Django Templates
- Automatic scaling



© 2008 Google, Inc. All rights reserved.



## Google App Engine Success Stories

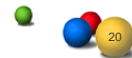


“We got a prototype of our new ‘Pix Chat’ OpenSocial app running in App Engine and the Hi5 sandbox this morning. It took about 3 hours to get the app serving and our db code converted.”

[PixVerse](#) (now acquired by Hi5)



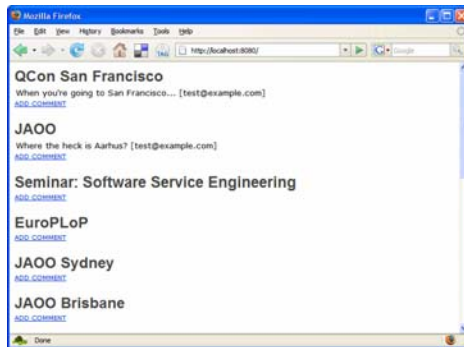
© 2008 Google, Inc. All rights reserved.



## App Engine Example: Calendar Feedback



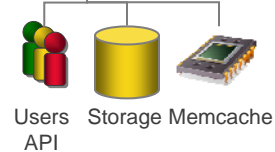
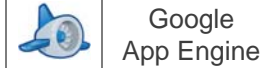
Browser



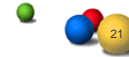
<http://gregortravel.appspot.com/>



Atom (XML)  
Over http



© 2008 Google, Inc. All rights reserved.



## Data Access



- Models and Entities declared in code

```
class Comment(db.Model):  
    author = db.UserProperty()  
    eventKey = db.StringProperty()  
    comment = db.StringProperty(multiline=True)  
    date = db.DateTimeProperty(auto_now_add=True)
```

Declaration

```
comments = Comment.all()  
comments.filter("eventKey =", id)  
comments.order("-date")  
for c in comments:  
    self.response.out.write(c.comment)
```

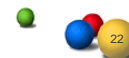
Query

```
comment = Comment()  
comment.author = users.get_current_user()  
comment.comment = self.request.get('content')  
comment.eventKey = self.request.get('id')  
comment.put()
```

Insert /  
Update



© 2008 Google, Inc. All rights reserved.

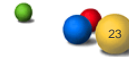


## Programming the Cloud

- Programming the cloud is exciting, but uses different programming and run-time models
- Parallel execution, constraint-based programming instead of linear loops
- Highly distributed data storage instead of RDBMS
- Live with the uncertainty: retry, compensation, tentative operations
- Tools and API's can take make your live a lot easier, but you have to do your part



© 2008 Google, Inc. All rights reserved.



## Google and the Cloud

- Google Data API's
- Google App Engine
- Google Mashup Editor
- Academic Cloud Computing Initiative (IBM & Google)
  - <http://code.google.com/edu/parallel>
- Developer community
  - <http://code.google.com/apis>
- Open Source
  - <http://code.google.com/opensource/>



© 2008 Google, Inc. All rights reserved.

