МОСКВА
11-12 мая 2012

Application Developer Days
/*Программисты всех платформ, общайтесь!*/

Riak — простая, предсказуемая и масштабируемая БД. Преимущества и недостатки.

Илья Богунов
Профессионалы.ру
twitter.com/tech_mind

О чем это ?

- Исторический экскурс БД от mainframe`ов 80х к вебу 2000-ных и текущим временам.
- Что такое Riak. Как он выглядит для разработчика\админа.
- Кто его и зачем использует.
- Дополнительные плюшки, кроме чистого kv.
- Не silver bullet - wtf`ы, проблемы и моменты, которые надо учитывать.
- Код и коммьюнити.
- «Кто, где и когда» ? Зачем и где его использовать.

http://www.flickr.com/photos/lofink/4501610335/

DB: mainframe`ы

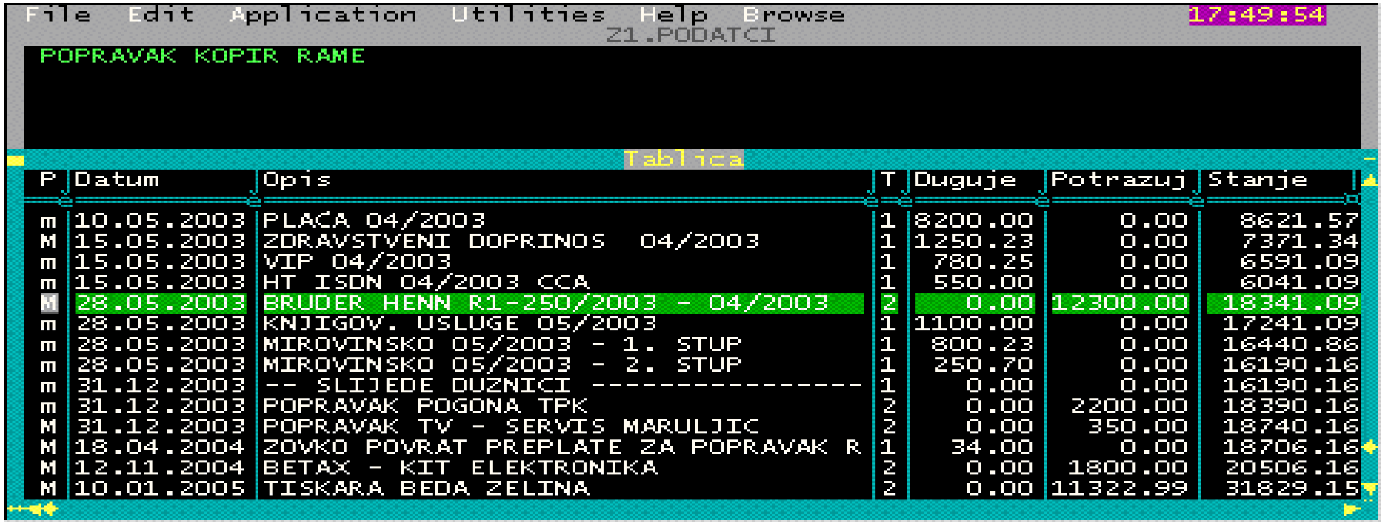
As an OLTP system benchmark, TPC-C simulates a *complete environment where a population of terminal operators* executes transactions against a database.

The benchmark is centered around the principal activities (transactions) of an order-entry environment.

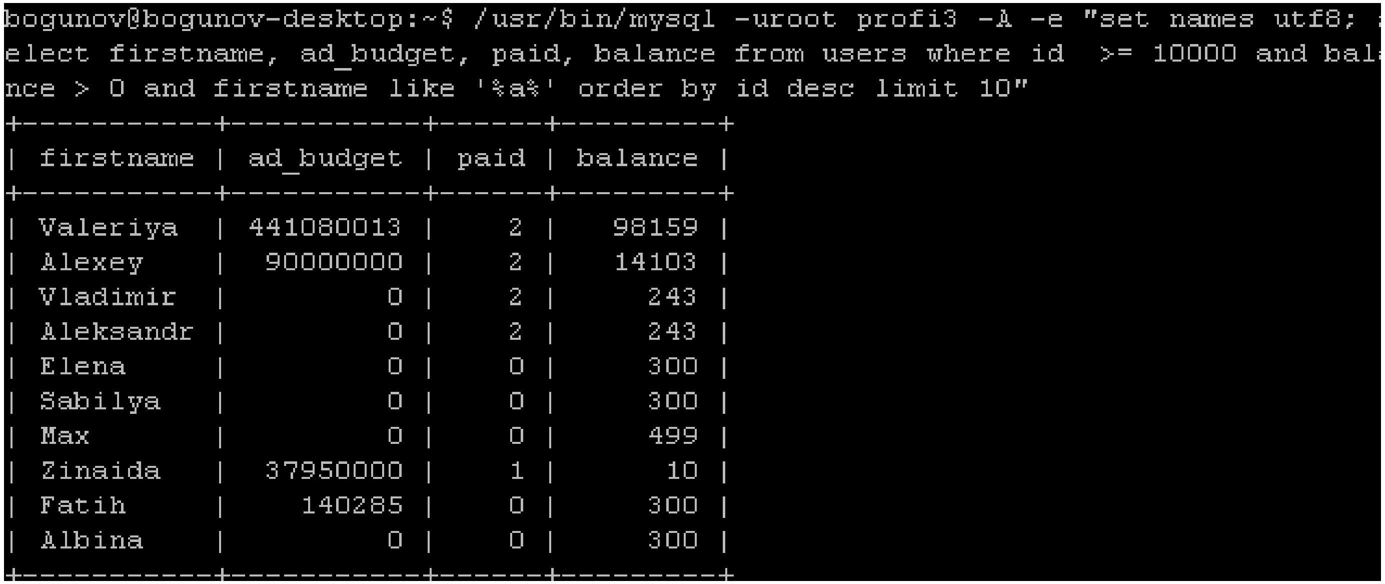
These transactions include entering and delivering

orders (**insert, update**),
recording payments (**insert, update**),
checking the status of orders (**select**), and
monitoring the level of stock at the warehouses (**group by, sum, etc**).

http://www.tpc.org/tpcc/detail.asp



```
 File   Edit   Application   Utilities   Help   Browse                  17:49:54
                                      Z1.PODATCI
 POPRAVAK KOPIR RAME




                                       Tablica
 P |Datum     |Opis                                    |T |Duguje   |Potrazuj |Stanje

 m |10.05.2003|PLACA 04/2003                           |1 |8200.00  |    0.00 |  8621.57
 M |15.05.2003|ZDRAVSTVENI DOPRINOS  04/2003           |1 |1250.23  |    0.00 |  7371.34
 m |15.05.2003|VIP 04/2003                             |1 | 780.25  |    0.00 |  6591.09
 m |15.05.2003|HT ISDN 04/2003 CCA                     |1 | 550.00  |    0.00 |  6041.09
 M |28.05.2003|BRUDER HENN R1-250/2003 - 04/2003       |2 |   0.00  |12300.00 | 18341.09
 m |28.05.2003|KNJIGOV. USLUGE 05/2003                 |1 |1100.00  |    0.00 | 17241.09
 m |28.05.2003|MIROVINSKO 05/2003 - 1. STUP            |1 | 800.23  |    0.00 | 16440.86
 m |28.05.2003|MIROVINSKO 05/2003 - 2. STUP            |1 | 250.70  |    0.00 | 16190.16
 m |31.12.2003|-- SLIJEDE DUZNICI -----------------    |1 |   0.00  |    0.00 | 16190.16
 m |31.12.2003|POPRAVAK POGONA TPK                     |2 |   0.00  | 2200.00 | 18390.16
 M |31.12.2003|POPRAVAK TV - SERVIS MARULJIC           |2 |   0.00  |  350.00 | 18740.16
 M |18.04.2004|ZOVKO POVRAT PREPLATE ZA POPRAVAK R     |1 |  34.00  |    0.00 | 18706.16
 M |12.11.2004|BETAX - KIT ELEKTRONIKA                 |2 |   0.00  | 1800.00 | 20506.16
 M |10.01.2005|TISKARA BEDA ZELINA                     |2 |   0.00  |11322.99 | 31829.15
```



```
bogunov@bogunov-desktop:~$ /usr/bin/mysql -uroot profi3 -A -e "set names utf8; s
elect firstname, ad_budget, paid, balance from users where id  >= 10000 and bala
nce > 0 and firstname like '%a%' order by id desc limit 10"
+-----------+-----------+------+---------+
| firstname | ad_budget | paid | balance |
+-----------+-----------+------+---------+
| Valeriya  | 441080013 |    2 |   98159 |
| Alexey    |  90000000 |    2 |   14103 |
| Vladimir  |         0 |    2 |     243 |
| Aleksandr |         0 |    2 |     243 |
| Elena     |         0 |    0 |     300 |
| Sabilya   |         0 |    0 |     300 |
| Max       |         0 |    0 |     499 |
| Zinaida   |  37950000 |    1 |      10 |
| Fatih     |    140285 |    0 |     300 |
| Albina    |         0 |    0 |     300 |
+-----------+-----------+------+---------+
```

DB/APP 80-х: интерфейсы
(найдите 10 отличий)

DB: MAINFRAME -> WEB

60-70e

+

= SQL

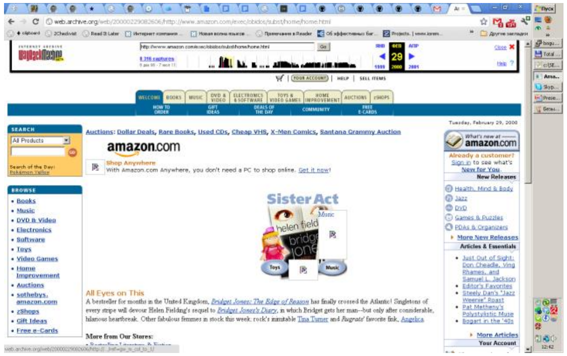
80e

= SQL

DB: MAINFRAME -> WEB

90e-2000e

+



web.archive.org/web/20000229082606/http://www.amazon.com/exec/obidos/subst/home/home.html

http://www.amazon.com/exec/obidos/subst/home/home.html

YOUR ACCOUNT | HELP | SELL ITEMS

WELCOME | BOOKS | MUSIC | DVD & VIDEO | ELECTRONICS & SOFTWARE | TOYS & VIDEO GAMES | HOME IMPROVEMENT | AUCTIONS | zSHOPS

HOW TO ORDER | GIFT IDEAS | DEALS OF THE DAY | COMMUNITY | FREE E-CARDS

Tuesday, February 29, 2000

SEARCH

All Products

Search of the Day!
Pokemon Yellow

Auctions: Dollar Deals, Rare Books, Used CDs, Cheap VHS, X-Men Comics, Santana Grammy Auction

amazon.com

Shop Anywhere
With Amazon.com Anywhere, you don't need a PC to shop online. Get it now!

Sister Act

All Eyes on This
A bestseller for months in the United Kingdom, *Bridget Jones: The Edge of Reason* has finally crossed the Atlantic! Singletons of every stripe will devour Helen Fielding's sequel to *Bridget Jones's Diary*, in which Bridget gets her man--but only after considerable, hilarious heartbreak. Other fabulous femmes in stock this week: rock's inimitable Tina Turner and *Rugrats*' favorite fink, Angelica.

More from Our Stores:

BROWSE
- Books
- Music
- DVD & Video
- Electronics
- Software
- Toys
- Video Games
- Home Improvement
- Auctions
- sothebys.amazon.com
- zShops
- Gift Ideas
- Free e-Cards

What's new at amazon.com

Already a customer?
Sign in to see what's New for You.

New Releases
- Health, Mind & Body
- Jazz
- DVD
- Games & Puzzles
- PDAs & Organizers
- More New Releases

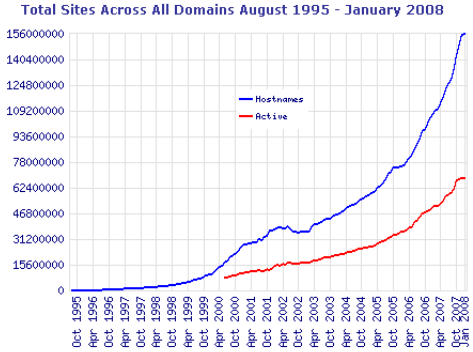
Articles & Essentials
- Just Out of Sight: Don Cheadle, Ving Rhames, and Samuel L. Jackson
- Editor's Favorites
- Steely Dan's "Jazz Weenie" Roast
- Pat Metheny's Polystylistic Muse
- Bogart in the '40s
- More Articles

Your Account

= (My)SQL

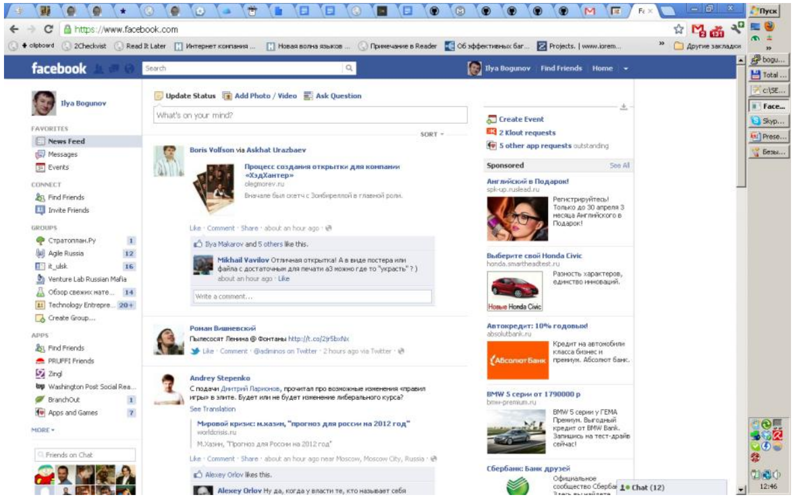
2000++

+



Total Sites Across All Domains August 1995 - January 2008

+



https://www.facebook.com

facebook | Ilya Bogunov | Find Friends | Home

Ilya Bogunov

FAVORITES
- News Feed
- Messages
- Events

CONNECT
- Find Friends
- Invite Friends

GROUPS
- Стратоплан.Ру 1
- Agile Russia 12
- it_ulsk 16
- Venture Lab Russian Mafia
- Обзор свежих мате... 14
- Technology Entrepre... 20+
- Create Group...

APPS
- Find Friends
- PRUFFI Friends
- Zingl
- Washington Post Social Rea...
- BranchOut 1
- Apps and Games 7

MORE

Update Status | Add Photo / Video | Ask Question

What's on your mind?

Boris Volfson via Askhat Urazbaev
Процесс создания открытки для компании «ХэдХантер»
olegmorev.ru
Вначале был скетч с Зомбирелпой в главной роли.
Like · Comment · Share · about an hour ago
Ilya Makarov and 5 others like this.
Mikhail Vavilov Отличная открытка! А в виде постера или файла с достаточным для печати а3 можно где то "украсть" ?)
about an hour ago · Like
Write a comment...

Роман Вишневский
Пылесосят Ленина @ Фонтаны http://t.co/2jr5bxNx
Like · Comment · @adminos on Twitter · 2 hours ago via Twitter

Andrey Stepenko
С подачи Дмитрий Ларионов, прочитал про возможные изменения «правил игры» в элите. Будет или не будет изменение либерального курса?
See Translation
Мировой кризис: м.хазин, "прогноз для россии на 2012 год"
worldcrisis.ru
М.Хазин, "Прогноз для России на 2012 год"
Like · Comment · Share · about an hour ago near Moscow, Moscow City, Russia
Alexey Orlov likes this.
Alexey Orlov Ну да, когда у власти те, кто называет себя

Create Event
2 Klout requests
5 other app requests outstanding

Sponsored

Английский в Подарок!
spk-up.ruslead.ru
Регистрируйтесь! Только до 30 апреля 3 месяца Английского в Подарок!

Выберите свой Honda Civic
honda.smartheadtest.ru
Разность характеров, единство инноваций.

Автокредит: 10% годовых!
absolutbank.ru
Кредит на автомобили класса бизнес и премиум. Абсолют банк.

BMW 5 серии от 1790000 р
bmw-premium.ru
BMW 5 серии у ГЕМА Премиум. Выгодный кредит от BMW Bank. Запишись на тест-драйв сейчас!

Сбербанк: Банк друзей
Официальное сообщество Сбербанка

Chat (12)

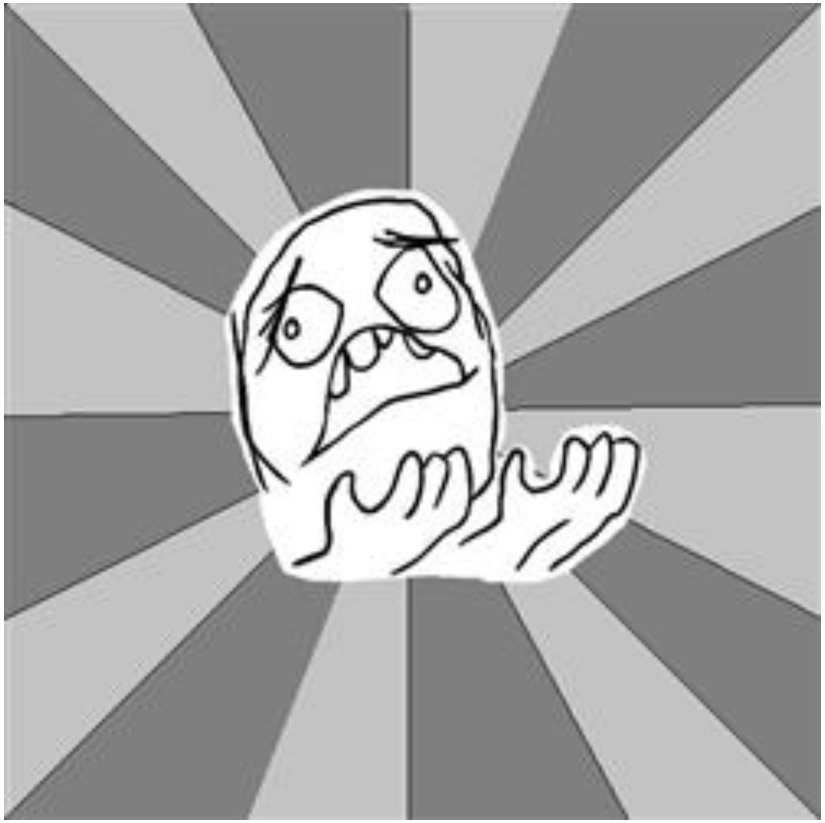
= SQL ?

DB: MAINFRAME -> WEB

MySql

- Replica (v 3.23.15, 2000)
- memcached (2003)
- batch insert/queue (mainframe`s)
- c/java daemon (custom binary file format)
- sharding (vertical/horizontal) + hand-made sharding framework

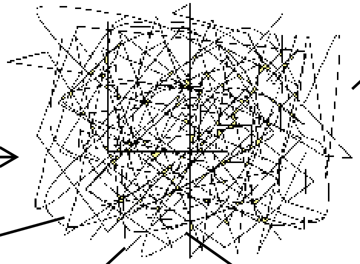
DB: MAINFRAME -> WEB



SHARDING

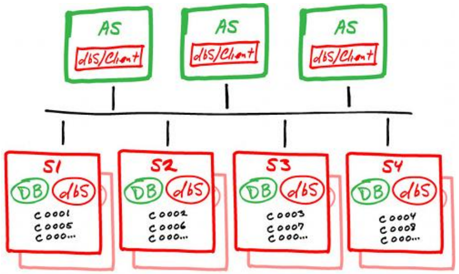
DB: MAINFRAME -> WEB

SHARDING



Middle layer

http://www.dbshards.com/dbshards/

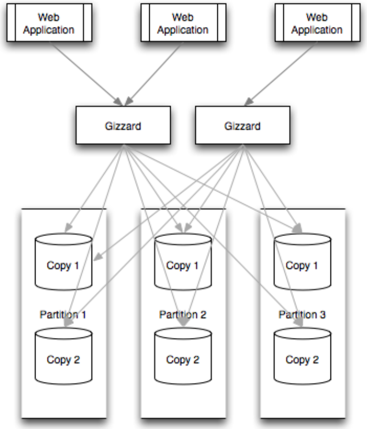


EAV

http://backchannel.org/blog/friendfeed-schemaless-mysql

Sharding framework

https://github.com/twitter/gizzard

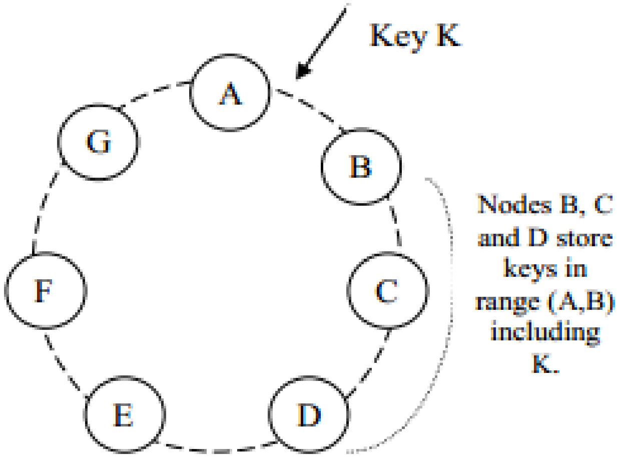


mysql patching

https://www.facebook.com/note.php?note_id=332335025932

DB: MAINFRAME -> WEB

Dynamo: Amazon’s Highly Available Key-value Store (2007), http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

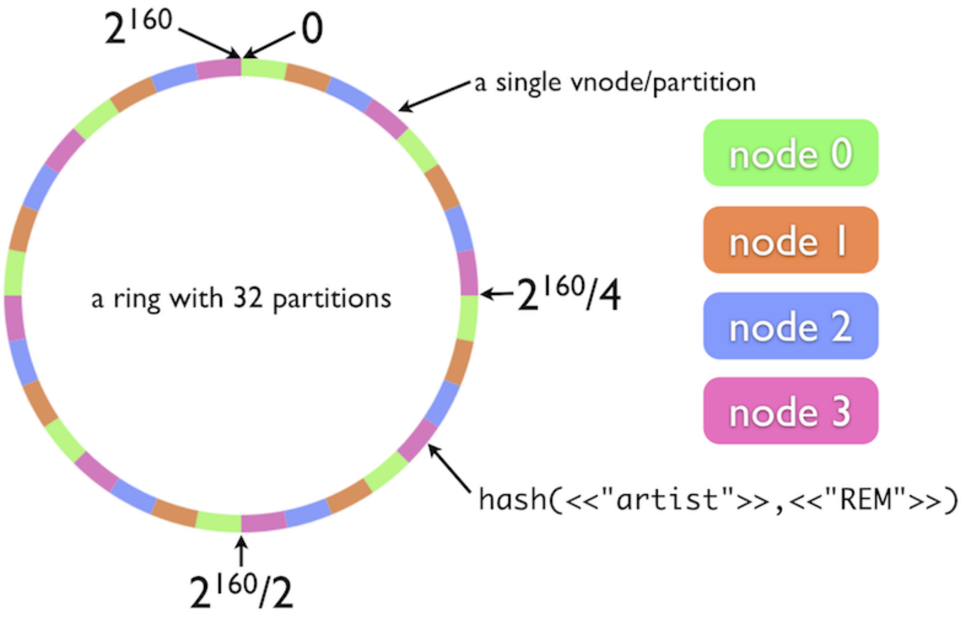


DB: MAINFRAME -> WEB

Amazon Dynamo Clones:

- Cassandra (2008)
- Voldemort (2009)
- Riak (2009)

DHT



RIAK:FOR DEVELOPER

RIAK(KV)

- PUT/GET/DELETE

  Client:get(<<“bucket”>>, <<“key”>>)

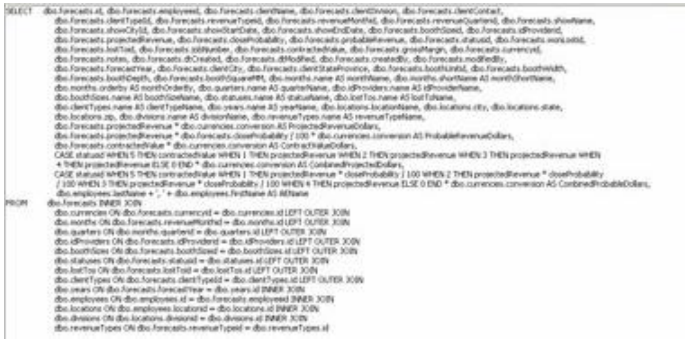
- A (req/seconds)

  × B (% file cache miss)

  ≈ C iops

SQL

- INSERT/CREATE/UPDATE/DELETE/, CREATE/ALTER/DROP, GRANT/REVOKE/DENY, MERGE/JOIN/GROUP BY/HAVING/UNION …



```
SELECT   dbo.forecasts.id, dbo.forecasts.employeeId, dbo.forecasts.clientName, dbo.forecasts.clientDivision, dbo.forecasts.clientContact,
           dbo.forecasts.clientTypeId, dbo.forecasts.revenueTypeId, dbo.forecasts.revenueMonthId, dbo.forecasts.revenueQuarterId, dbo.forecasts.showName,
           dbo.forecasts.showCityId, dbo.forecasts.showStartDate, dbo.forecasts.showEndDate, dbo.forecasts.boothSizeId, dbo.forecasts.idProviderId,
           dbo.forecasts.projectedRevenue, dbo.forecasts.closeProbability, dbo.forecasts.probableRevenue, dbo.forecasts.statusId, dbo.forecasts.wonLostId,
           dbo.forecasts.lostToId, dbo.forecasts.jobNumber, dbo.forecasts.contractedValue, dbo.forecasts.grossMargin, dbo.forecasts.currencyId,
           dbo.forecasts.notes, dbo.forecasts.dtCreated, dbo.forecasts.dtModified, dbo.forecasts.createdBy, dbo.forecasts.modifiedBy,
           dbo.forecasts.forecastYear, dbo.forecasts.clientCity, dbo.forecasts.clientStateProvince, dbo.forecasts.boothUnitId, dbo.forecasts.boothWidth,
           dbo.forecasts.boothDepth, dbo.forecasts.boothSquareMM, dbo.months.name AS monthName, dbo.months.shortName AS monthShortName,
           dbo.months.orderby AS monthOrderBy, dbo.quarters.name AS quarterName, dbo.idProviders.name AS idProviderName,
           dbo.boothSizes.name AS boothSizeName, dbo.statuses.name AS statusName, dbo.lostTos.name AS lostToName,
           dbo.clientTypes.name AS clientTypeName, dbo.years.name AS yearName, dbo.locations.locationName, dbo.locations.city, dbo.locations.state,
           dbo.locations.zip, dbo.divisions.name AS divisionName, dbo.revenueTypes.name AS revenueTypeName,
           dbo.forecasts.projectedRevenue * dbo.currencies.conversion AS ProjectedRevenueDollars,
           dbo.forecasts.projectedRevenue * dbo.forecasts.closeProbability / 100 * dbo.currencies.conversion AS ProbableRevenueDollars,
           dbo.forecasts.contractedValue * dbo.currencies.conversion AS ContractValueDollars,
           CASE statusId WHEN 5 THEN contractedValue WHEN 1 THEN projectedRevenue WHEN 2 THEN projectedRevenue WHEN 3 THEN projectedRevenue WHEN
            4 THEN projectedRevenue ELSE 0 END * dbo.currencies.conversion AS CombinedProjectedDollars,
           CASE statusId WHEN 5 THEN contractedValue WHEN 1 THEN projectedRevenue * closeProbability / 100 WHEN 2 THEN projectedRevenue * closeProbability
            / 100 WHEN 3 THEN projectedRevenue * closeProbability / 100 WHEN 4 THEN projectedRevenue ELSE 0 END * dbo.currencies.conversion AS CombinedProbableDollars,
            dbo.employees.lastName + ', ' + dbo.employees.firstName AS AEName
FROM     dbo.forecasts INNER JOIN
           dbo.currencies ON dbo.forecasts.currencyId = dbo.currencies.id LEFT OUTER JOIN
           dbo.months ON dbo.forecasts.revenueMonthId = dbo.months.id LEFT OUTER JOIN
           dbo.quarters ON dbo.months.quarterId = dbo.quarters.id LEFT OUTER JOIN
           dbo.idProviders ON dbo.forecasts.idProviderId = dbo.idProviders.id LEFT OUTER JOIN
           dbo.boothSizes ON dbo.forecasts.boothSizeId = dbo.boothSizes.id LEFT OUTER JOIN
           dbo.statuses ON dbo.forecasts.statusId = dbo.statuses.id LEFT OUTER JOIN
           dbo.lostTos ON dbo.forecasts.lostToId = dbo.lostTos.id LEFT OUTER JOIN
           dbo.clientTypes ON dbo.forecasts.clientTypeId = dbo.clientTypes.id LEFT OUTER JOIN
           dbo.years ON dbo.forecasts.forecastYear = dbo.years.id INNER JOIN
           dbo.employees ON dbo.employees.id = dbo.forecasts.employeeId INNER JOIN
           dbo.locations ON dbo.employees.locationId = dbo.locations.id INNER JOIN
           dbo.divisions ON dbo.locations.divisionId = dbo.divisions.id INNER JOIN
           dbo.revenueTypes ON dbo.forecasts.revenueTypeId = dbo.revenueTypes.id
```

- X (req/seconds)

  × [index, locks, read-ahead, random-reads, optimizator index selection, cache, buffer sizes]

  = WTF ???

RIAK:FOR DEVELOPER

RIAK(KV) vs SQL

Client:mapred(List, fun () -> doX() end) = length(List) io reqs
GROUP BY/COUNT = ? io (random read + filesort) reqs

~~pessimistic/optimistic locking,~~ WRITE ALWAYS, FIX ON READ
START TRANSACTION/COMMIT/ROLLBACK

`cat /etc/riak/app.config | grep "\s*{" | grep -v "%%" | wc -l` = 78
`mysql -u root -e "show variables" | wc -l` = 301

~~ALTER~~
ALTER DOWNTIME\CUSTOM MIGRATION SCRIPT

RIAK: FOR ADMIN/OPS

- Read repair, failover, hinted handoff.
- горизонтальное масштабирование
- легкость установки и добавления ноды:

```
apt-get\pkgin\dpkg install riak
riak start
riak-admin join node@name
```

CASSANDRA

http://wiki.apache.org/cassandra/Operations#Bootstrap

- **Bootstrap**
- Adding new nodes is called "bootstrapping."
- To bootstrap a node, turn AutoBootstrap on in the configuration file, and start it.
- If you explicitly specify an InitialToken in the configuration, the new node will bootstrap to that position on the ring. Otherwise, it will pick a Token that will give it half the keys from the node with the most disk space used, that does not already have another node bootstrapping into its Range.
- Important things to note:
- You should wait long enough for all the nodes in your cluster to become aware of the bootstrapping node via gossip before starting another bootstrap. The new node will log "Bootstrapping" when this is safe, 2 minutes after starting. (90s to make sure it has accurate load information, and 30s waiting for other nodes to start sending it inserts happening in its to-be-assumed part of the token ring.)
- Relating to point 1, one can only bootstrap N nodes at a time with automatic token picking, where N is the size of the existing cluster. If you need to more than double the size of your cluster, you have to wait for the first N nodes to finish until your cluster is size 2N before bootstrapping more nodes. So if your current cluster is 5 nodes and you want add 7 nodes, bootstrap 5 and let those finish before bootstrapping the last two.
- As a safety measure, Cassandra does not automatically remove data from nodes that "lose" part of their Token Range to a newly added node. Run nodetool cleanup on the source node(s) (neighboring nodes that shared the same subrange) when you are satisfied the new node is up and working. If you do not do this the old data will still be counted against the load on that node and future bootstrap attempts at choosing a location will be thrown off.
- When bootstrapping a new node, existing nodes have to divide the key space before beginning replication. This can take awhile, so be patient.
- During bootstrap, a node will drop the Thrift port and will not be accessible from nodetool.
- Bootstrap can take many hours when a lot of data is involved. See Streaming for how to monitor progress.
- Cassandra is smart enough to transfer data from the nearest source node(s), if your EndpointSnitch is configured correctly. So, the new node doesn't need to be in the same datacenter as the primary replica for the Range it is bootstrapping into, as long as another replica is in the datacenter with the new one.
- Bootstrap progress can be monitored using nodetool with the netstats argument (0.7 and later) or streams (Cassandra 0.6).
- During bootstrap nodetool may report that the new node is not receiving nor sending any streams, this is because the sending node will copy out locally the data they will send to the receiving one, which can be seen in the sending node through the the "AntiCompacting... AntiCompacted" log messages.
- **Moving or Removing nodes**
- **Removing nodes entirely**
- You can take a node out of the cluster with nodetool decommission to a live node, or nodetool removetoken (to any other machine) to remove a dead one. This will assign the ranges the old node was responsible for to other nodes, and replicate the appropriate data there. If decommission is used, the data will stream from the decommissioned node. Ifremovetoken is used, the data will stream from the remaining replicas.
- No data is removed automatically from the node being decommissioned, so if you want to put the node back into service at a different token on the ring, it should be removed manually.
- **Moving nodes**
- nodetool move: move the target node to a given Token. Moving is both a convenience over and more efficient than decommission + bootstrap. After moving a node, nodetool cleanup should be run to remove any unnecessary data.
- As with bootstrap, see Streaming for how to monitor progress.
- **Load balancing**
- If you add nodes to your cluster your ring will be unbalanced and only way to get perfect balance is to compute new tokens for every node and assign them to each node manually by using nodetool move command.
- Here's a python program which can be used to calculate new tokens for the nodes. There's more info on the subject at Ben Black's presentation at Cassandra Summit 2010. http://www.datastax.com/blog/slides-and-videos-cassandra-summit-2010
- def tokens(nodes):
  - for x in xrange(nodes):
    - print 2 ** 127 / nodes * x
- In versions of Cassandra 0.7.* and lower, there's also nodetool loadbalance: essentially a convenience over decommission + bootstrap, only instead of telling the target node where to move on the ring it will choose its location based on the same heuristic as Token selection on bootstrap. You should not use this as it doesn't rebalance the entire ring.
- The status of move and balancing operations can be monitored using nodetool with the netstat argument. (Cassandra 0.6.* and lower use the streams argument).
- **Replacing a Dead Node**
- Since Cassandra 1.0 we can replace a dead node with a new one using the property "cassandra.replace_token=<Token>", This property can be set using -D option while starting cassandra demon process.
- (Note:This property will be taken into effect only when the node doesn't have any data in it, You might want to empty the data dir if you want to force the node replace.)
- You must use this property when replacing a dead node (If tried to replace an existing live node, the bootstrapping node will throw a Exception). The token used via this property must be part of the ring and the node have died due to various reasons.
- Once this Property is enabled the node starts in a hibernate state, during which all the other nodes will see this node to be down. The new node will now start to bootstrap the data from the rest of the nodes in the cluster (Main difference between normal bootstrapping of a new node is that this new node will not accept any writes during this phase). Once the bootstrapping is complete the node will be marked "UP", we rely on the hinted handoff's for making this node consistent (Since we don't accept writes since the start of the bootstrap).
- Note: We Strongly suggest to repair the node once the bootstrap is completed, because Hinted handoff is a "best effort and not a guarantee".

VOLDEMORT

https://github.com/voldemort/voldemort/wiki/Voldemort-Rebalancing

- **What are the actual steps performed during rebalancing?**
- Following are the steps that we go through to rebalance successfully. The controller initiates the rebalancing and then waits for the completion.
- Input
  - Either (a) current cluster xml, current stores xml, target cluster xml (b) url, target cluster xml
  - batch size – Number of primary partitions to move together. There is a trade-off, more primary partitions movements = more redirections.
  - max parallel rebalancing – Number of units (stealer + donor node tuples) to move together
- Get the latest state from the cluster
  Compare the targetCluster.xml provided and add all new nodes to currentCluster.xml
- Verify that
  - We are not in rebalancing state already ("./bin/voldemort-admin-tool.sh —get-metadata server.state —url [url]" all returns NORMAL_SERVER and "./bin/voldemort-admin-tool.sh —get-metadata rebalancing.steal.info.key —url [url]" all returns "[]" i.e. no rebalancing state)
  - RO stores (if they exist) are all using the latest storage format ("./bin/voldemort-admin-tool.sh —ro-metadata storage-format —url [url]" returns all stores with "ro2" format)
- Get a list of every primary partition to be moved
- For every "batch" of primary partitions to move
  - Create a transition cluster metadata which contains movement of "batch size" number of primary partitions
  - Create a rebalancing plan based on this transition cluster.xml and the current state of the cluster. The plan generated is a map of stealer node to list of donor node + partitions to be moved.
  - State change step
    - has_read-only_stores AND has_read-write_stores => Change the rebalancing state [with RO stores information] change on all stealer nodes
    - has_read-only_stores => Change the rebalancing state change on all stealer nodes
  - Start multiple units of migration [unit => stealer + donor node movement] of all RO Stores data [No changes to the routing layer and clients]. At the end of each migration delete the rebalancing state => Done with parallelism (max parallel rebalancing)
  - State change step [Changes in rebalancing state will kick in redirecting stores which will start redirecting requests to the old nodes]
    - hasROStore AND hasRWStore => Change the cluster metadata + Swap on all nodes AND Change rebalancing state [with RW stores information] on all stealer nodes
    - hasROStore AND !hasRWStore => Change the cluster metadata + Swap on all nodes
    - !hasROStore AND hasRWStore => Change the cluster metadata on all nodes AND Change rebalancing state [with RW stores information] on all stealer nodes
  - Start multiple units of migration of all RW Stores data [With redirecting happening]. At the end of migration delete the rebalancing state => Done with parallelism (max parallel rebalancing)
- **What about the failure scenarios?**
- Extra precaution has been taken and every step (5 [c,d,e,f]) has a rollback strategy
- Rollback strategy for
  5 (c , e) => If any failure takes place during the 'State change step', the following rollback strategy is run on every node that was completed successfully
- Swap ROChange cluster metadataChange rebalance stateOrderFTTremove the rebalance state change → change back clusterFFTremove from rebalance state changeTTFchange back cluster metadata → swapTTTremove from rebalance state change → change back cluster → swap5 (d, f) => Similarly during migration of partitions
- Has RO storesHas RW storesFinished RO storesRollback Action [in the correct order]TTTrollback cluster change + swapTTFnothing to do since "rebalance state change" should have removed everythingTFTwon't be triggered since hasRW is falseTFFnothing to do since "rebalance state change" should have removed everythingFTTrollback cluster changeFTFwon't be triggeredFFTwon't be triggeredFFFwon't be triggered**What happens on the stealer node side during 5 (d, f)?**
- The stealer node on receiving a "unit of migration" [unit => single stealer + single donor node migration] and does the following
- Check if the rebalancing state change was already done [i.e. 5 (c, e) was successfully completed]
- Acquire a lock for the donor node [Fail if donor node was already rebalancing]
- Start migration of the store partitions from the donor node => PARALLEL [setMaxParallelStoresRebalancing]. At the end of every store migration remove it from the list rebalance state change [so as to stop redirecting stores]
- **What about the donor side?**
- The donor node has no knowledge for rebalancing at all and keeps behaving normally.
- **What about my data consistency during rebalancing?**
- Rebalancing process has to maintain data consistency guarantees during rebalancing, We are doing it through a proxy based design. Once rebalancing starts the stealer node is the new master for all rebalancing partitions, All the clients talk directly to stealer node for all the requests for these partitions. Stealer node internally make proxy calls to the original donor node to return correct data back to the client.The process steps are
- Client request stealer node for key 'k1' belonging to partition 'p1' which is currently being migrated/rebalanced.
- Stealer node looks at the key and understands that this key is part of a rebalancing partition
- Stealer node makes a proxy call to donor node and gets the list of values as returned by the donor node.
- Stealer node does local put for all (key,value) pairs **ignoring all ObsoleteVersionException**
- Stealer node now should have all the versions from the original node and now does normal local get/put/ getAll operations.
- **And how do my clients know the changes?**
- Voldemort client currently bootstrap from the bootstrap URL at the start time and use the returned cluster/stores metadata for all subsequent operation. Rebalancing results in the cluster metadata change and so we need a mechanism to tell clients that they should rebootstrap if they have old metadata.
- Client Side routing bootstrapping: Since the client will be using the old metadata during rebalancing the server now throws an**InvalidMetadataException** if it sees a request for a partition which does not belong to it. On seeing this special exception the client is re-bootstraps from the bootstrap url and will hence pick up the correct cluster metadata.
- Server side routing bootstrapping: The other method of routing i.e. make calls to any server with enable_routing flag set to true and with re-routing to the correct location taking place on the server side (i.e. 2 hops). In this approach we've added a **RebootstrappingStore** which picks up the new metadata in case of change.
- **How to start rebalancing?**
- Step 1: Make sure cluster is not doing rebalancing.
  - The rebalancing steal info should be null
    ./bin/voldemort-admin-tool.sh —get-metadata rebalancing.steal.info.key —url [url]
  - The servers should be NORMAL_STATE
    ./bin/voldemort-admin-tool.sh —get-metadata server.state —url [url]
- To check whether the keys were moved correctly we need to save some keys and later check if they have been migrated to their new locations
  ./bin/voldemort-rebalance.sh —current-cluster [current_cluster] —current-stores [current_stores_path] —entropy false —output-dir [directory where we'll store the keys for later use. Keys are stored on a per store basis]
- Generate the new cluster xml. This can be done either by hand or if you want to do it automatically here is a tool to do to
  ./bin/voldemort-rebalance.sh —current-cluster [current_cluster_path] —target-cluster [should be the same as current-cluster but with new nodes put in with empty partitions] —current-stores [current_stores_path] —generate
- Run the new metadata through key-distribution generator to get an idea of skew if any. Make sure your standard deviation is close to 0.
  ./bin/run-class.sh voldemort.utils.KeyDistributionGenerator —cluster-xml [new_cluster_xml_generated_above] —stores-xml [stores_metadata]
- Use the new_cluster_xml and run the real rebalancing BUT first with —show-plan (to check what we're going to move)
  ./bin/voldemort-rebalance.sh —url [url] —target-cluster [new_cluster_metadata_generated] —show-plan
- Run the real deal
  ./bin/voldemort-rebalance.sh —url [url] —target-cluster [new_cluster_metadata_generated]
- Monitor by checking the async jobs
  ./bin/voldemort-admin-tool.sh —async get —url [url]
- The following takes place when we are running the real rebalancing (i) Pick batch of partitions to move (ii) Generate transition plan (iii) Execute the plan as a series of 'stealer-donor' node copying steps. By default we do only one 'stealer-donor' tuple movement at once. You can increase this by setting —parallelism option.
- If anything fails we can rollback easily (as long as —delete was not used while running voldemort-rebalance.sh)
- To stop an async process
  ./bin/voldemort-admin-tool.sh —async stop —async-id [comma separated list of async jobs] —url [url] —node [node on which the async job is running]
- To clear the rebalancing information on a particular node
  ./bin/voldemort-admin-tool.sh —clear-rebalancing-metadata —url [url] —node [node-id]
- Following are some of the configurations that will help you on the server side
- Parameter on serverDefaultWhat it doesenable.rebalancingtrueShould be true so as to run rebalancingmax.rebalancing.attempts3Once a stealer node receives the plan to copy from a donor node, it will attempt this many times to copy the data (in case of failure)rebalancing.timeout.seconds10 * 24 * 60 * 60Time we give for the server side rebalancing to finish copying data from a donor nodemax.parallel.stores.rebalancing3Stores to rebalance in parallelrebalancing.optimizationtrueSome times we have data stored without being partition aware (Example : BDB). In this scenario we can run an optimization phase which ignores copying data over if a replica already exists**What is left to make this process better?**
- a) Execute tasks should be smarter and choose tasks to execute so as to avoid two disk sweeps happening on the same node.
- b) Fix deletes! – Make it run at the end instead of in the middle [Even though I'll never run this in production]
- c) Logging – Currently we propagate the message at the lowest level all the way to the top. Instead we should try to make a better progress bar ("Number of stores completed – 5 / 10") and push that upwards.
- d) Currently the stealer node goes into REBALANCING_MASTER state and doesn't allow any disk sweeps (like slop pusher job, etc) from not taking place. But what about the poor donor node

HAND-MADE SHARDING

```
apt-get install x-sql
./configure
```

Мигрируем

- **легкий способ**:
  1. стопнуть сервис
  2. запустить скрипты миграции
  3. запустить сервис
- сложный способ:
  1. дописать код работающий с 2мя партициями одновременно
  2. научить приложение понимать, когда идет миграция и что надо читать\писать в ОБА места
  3. следить чтоб при переезде не разбалансировался кластер
  4. ощутить после переезда всю прелесть неучтенных многопоточных ситуаций\неконсистентных данных

RIAK: USES

- http://wiki.basho.com/Who-is-Using-Riak.html
- **Mozilla** http://blog.mozilla.com/data/2010/05/18/riak-and-cassandra-and-hbase-oh-my/
- **Unitypark** http://muchdifferent.com/
- **Yammer** http://dl.dropbox.com/u/2744222/2011-03-22_Riak-At-Yammer.pdf
- **Voxer** http://speakerd.s3.amazonaws.com/presentations/4f229a9421e6f8002201fa9f/ScaleConf-HealthyDistributedSystems.pdf
- **Unison** http://www.unison.com/
- **Mochimedia** http://www.mochimedia.com/
- **Seomoz** http://devblog.seomoz.org/2011/10/using-riak-for-ranking-collection/
- **Trifork** http://www.erlang-solutions.com/upload/docs/116/Basho%20and%20Trifork.pdf
- **clipboard.com** http://blog.clipboard.com/2012/03/18/0-Milking-Performance-From-Riak-Search
- **Echo** http://aboutecho.com/
- Другие =)

RIAK: Когда\Где ?

1. мне критична высоконадежность и высокодоступность
2. у меня миллионы юзеров\приложений\страниц\топиков\файлов\обьектов\лент новостей
3. они между собой не связаны (нет real-time GROUP BY/JOIN)
4. можно для аналитики типа time-series data
5. можно слепить ежа с ужом и прикрутить Riak core например к Lucene

Come to the dark side. We have cookies... -V
Come to the dark side. We have cookies... -V

Riak: дополнительные плюшки

Индексы

```
Key = <<"X">>, Value = <<"X">>, IndexInt = 1.

Robj = riak_object:new(<<"X">>, Key, Value),

MetaData = dict:from_list([{<<"index">>, [{<<"index_bin">>,
<<IndexInt:32/big-unsigned-integer>>}]}]),

Robj2 = riak_object:update_metadata(Robj, MetaData),

Client:put(Robj2, 1).

Client:get_index(<<"X">>, {range, <<"index_bin">>, <<1:32/big-
unsigned-integer>>, <<2:32/big-unsigned-integer>>}).

{ok,[<<"X">>]}
```

Riak: дополнительные плюшки

Полнотекстовый поиск

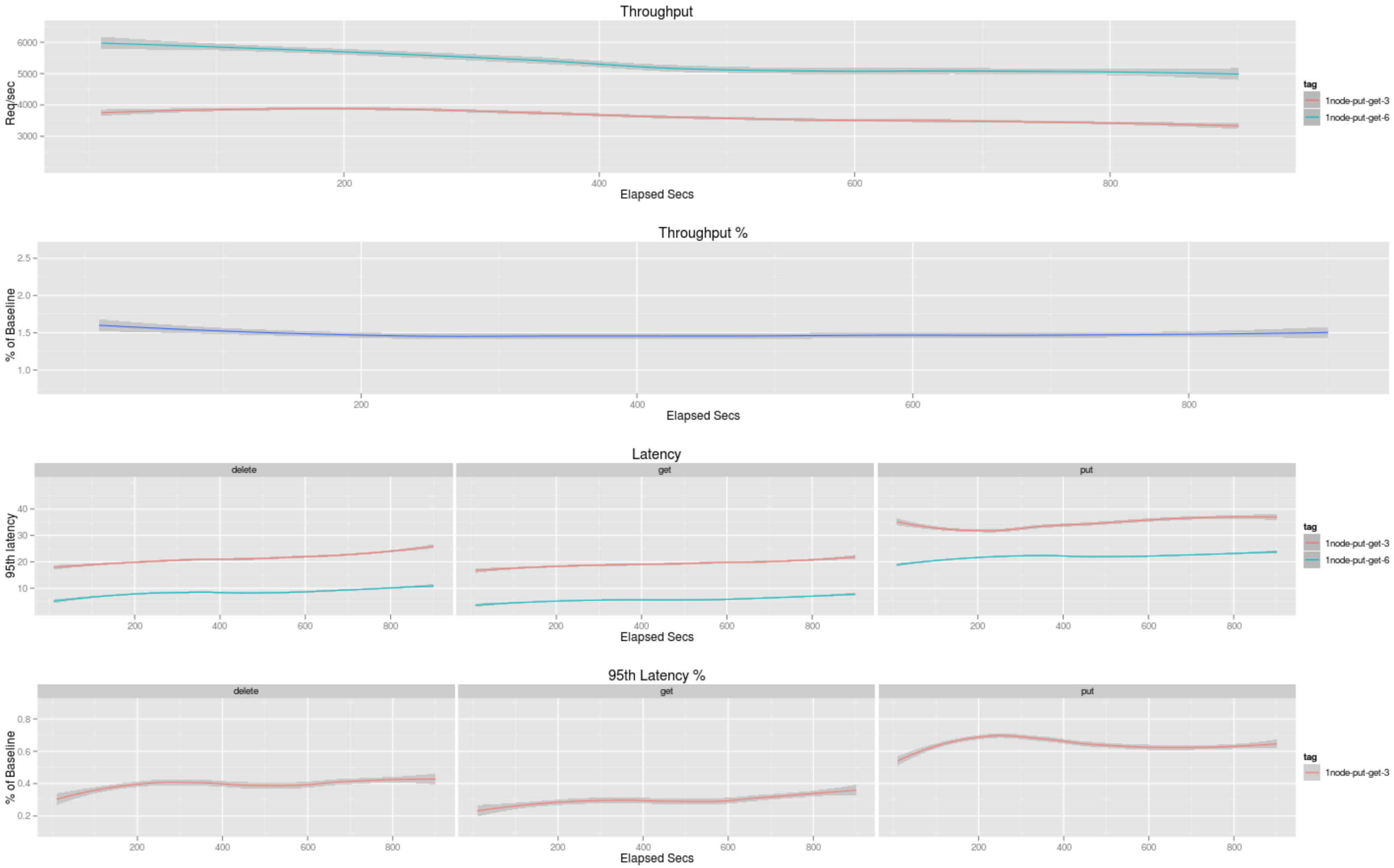
```
search:index_doc(<<"users">>, <<"test">>,
[{<<"title">>, <<"The Test">>}, {<<"content">>, <<"The
Content">>}]).

search:search(<<"users">>, <<"title:Test">>).

{1,
 [{<<"users">>,<<"test">>,
   [{p,[1]},{score,0.35355339059327373}]}]}
```

Riak: дополнительные плюшки

- Post\pre-commit хуки:
  - Поиск – это тот же пре-коммит хук
  - Интеграция с RabbitMq https://github.com/jbrisbin/riak-rabbitmq-commit-hooks
- Map-reduce.
  - На эрланге =)
  - На javascript`e.
  - По заданному списку ключей!

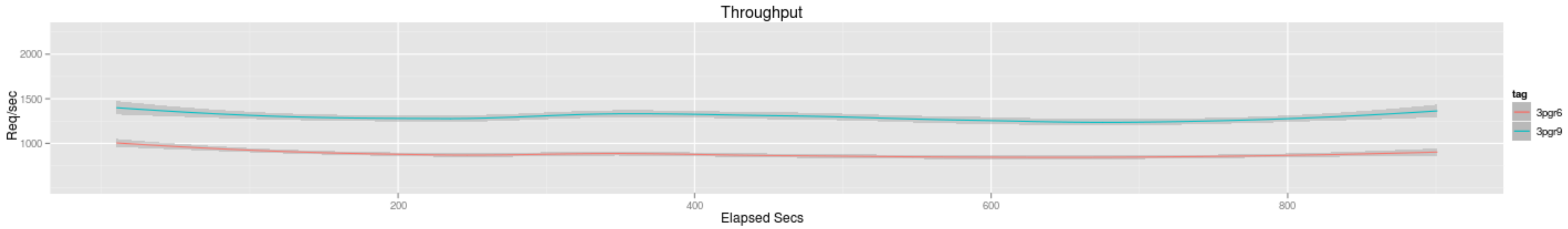
RIAK: а работает ?

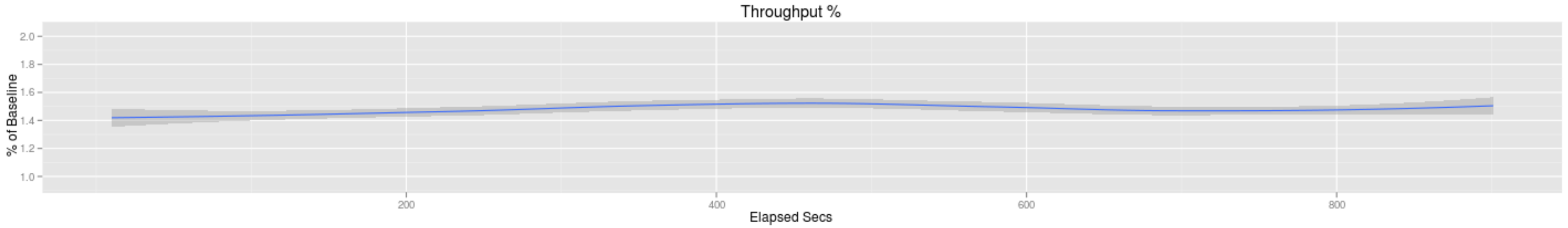
RIAK: (put\get\delete, uniform, 3vs6)

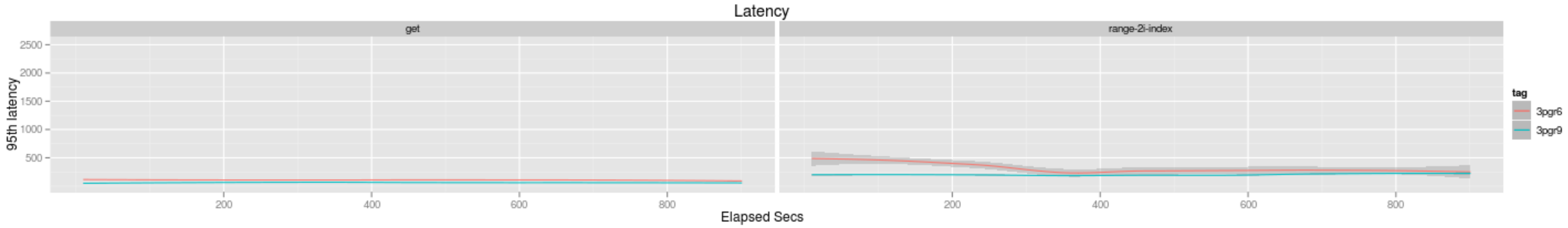


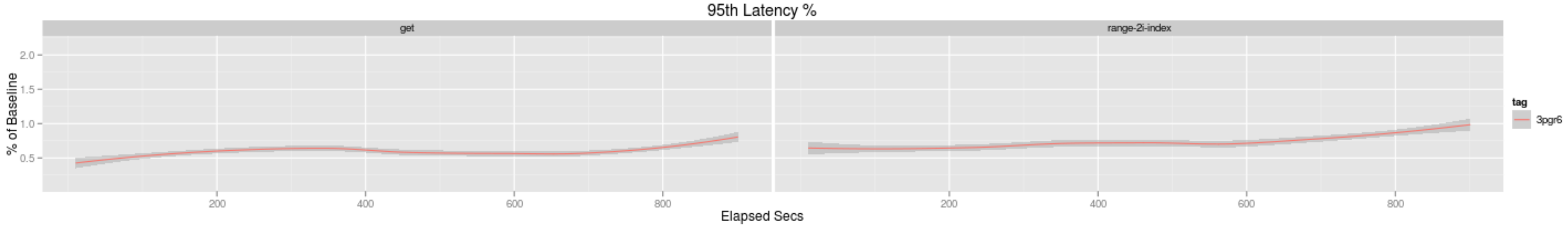
http://joyeur.com/2010/10/31/riak-smartmachine-benchmark-the-technical-details/

RIAK: (get\range_index, uniform, 6vs9)

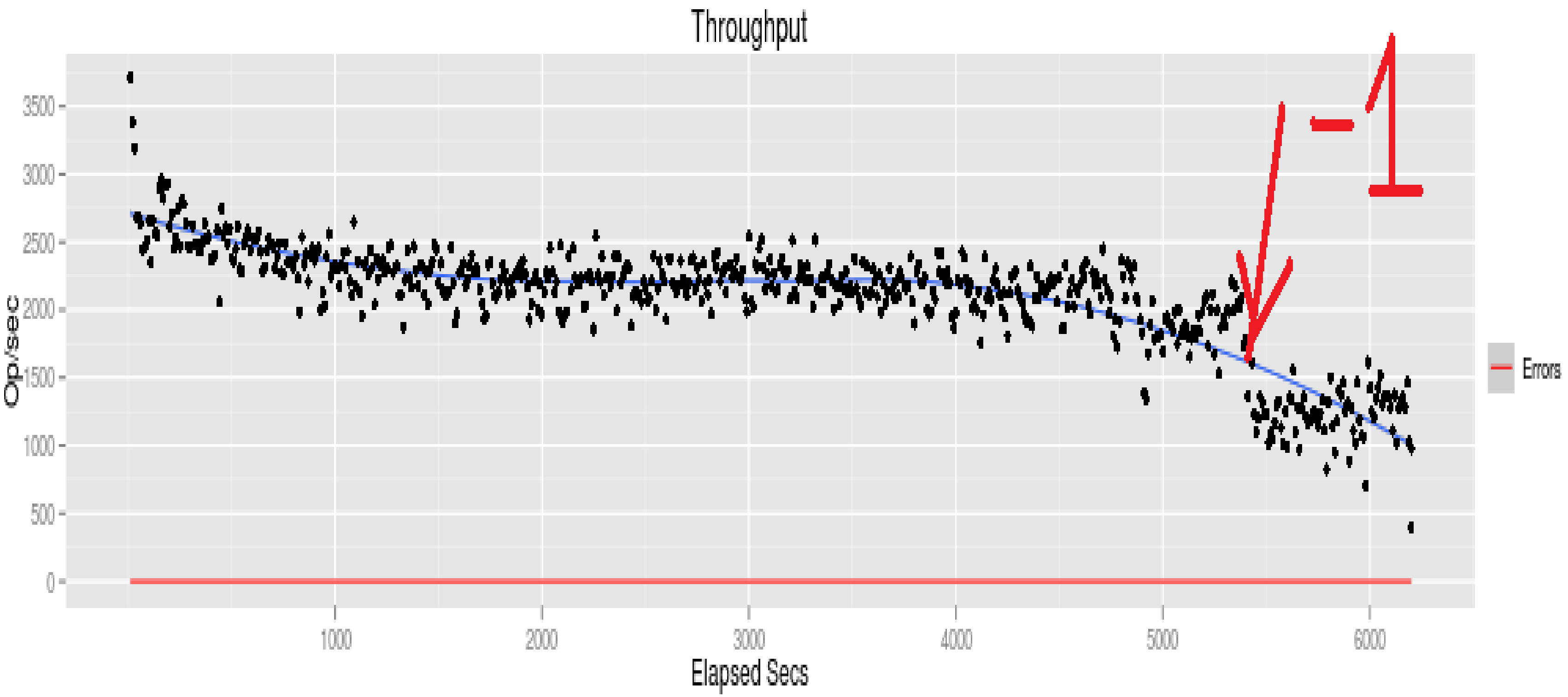




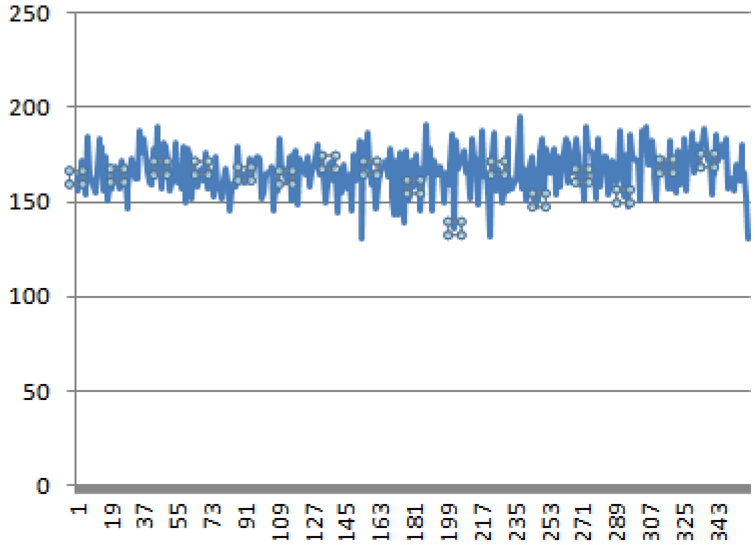




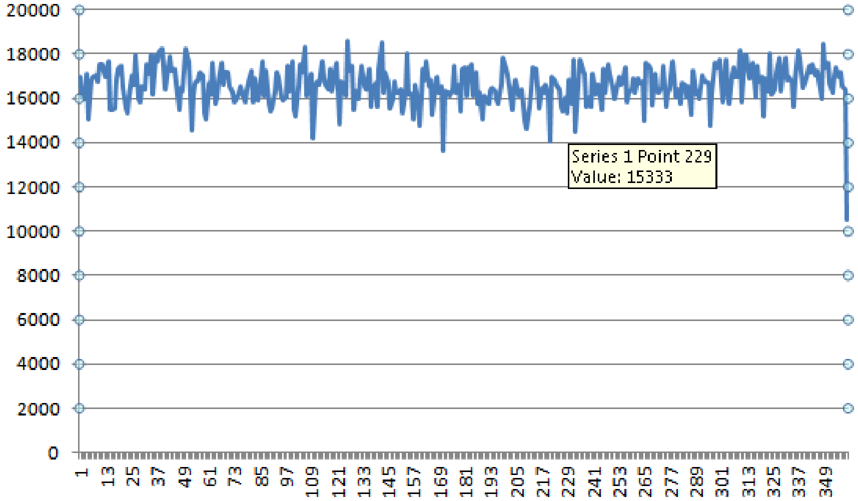
RIAK: (puts, 3 nodes, node failure)



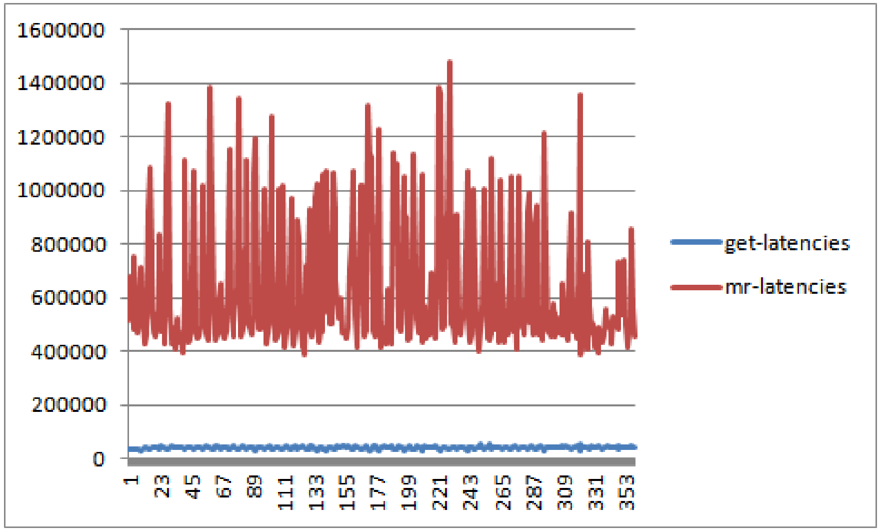
RIAK: GET vs MR-GET



mr-get(x100)



Get(x1)



RIAK: Неудобства и WTF`S

Неудобства:

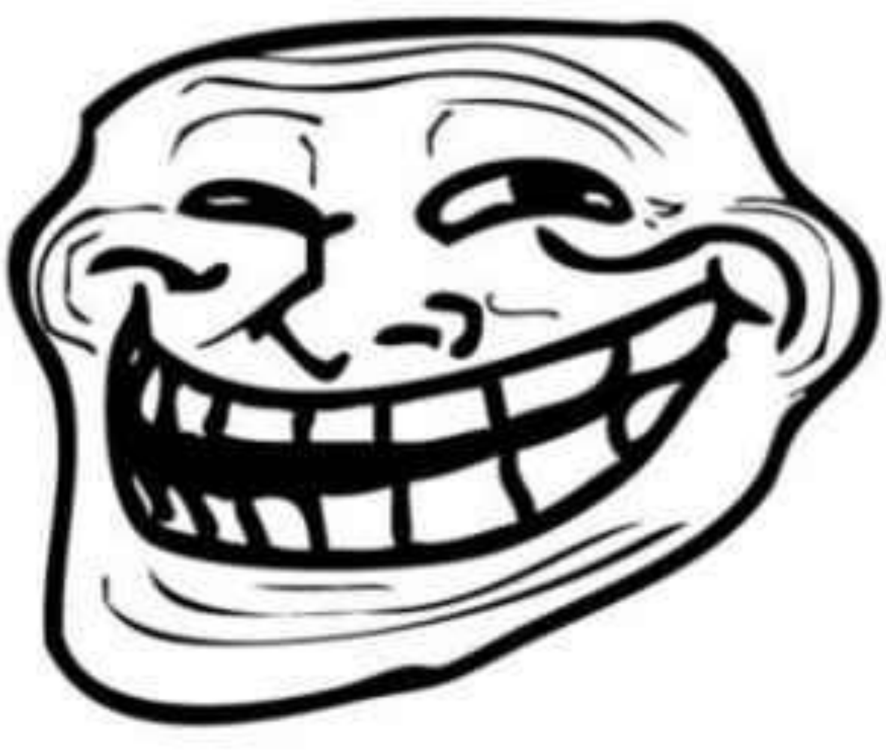
- Adhoc запросы невозможны, нужно ЗАРАНЕЕ подумать о своих паттернах доступа
- COUNT\GROUP BY, только пред-агрегацией (map-reduce`ом довольно не удобно), либо хранить счетчики в Redis`e
- Eventual consistency, планируем свои данные чтоб уметь мерджить
- Нет паджинации и сортировки - либо редис, либо что-то вроде linked-list`a

RIAK: Неудобства и WTF`S

WTF’S:

- BUCKET = PREFIX
- «урезанный» map-reduce
- Восстановление данных только по read-repair на чтении, данные сами не починятся (без EDS)
- RiakSearch – нет анти-энтропии
- Специфичные проблемы, каждого из бекэндов

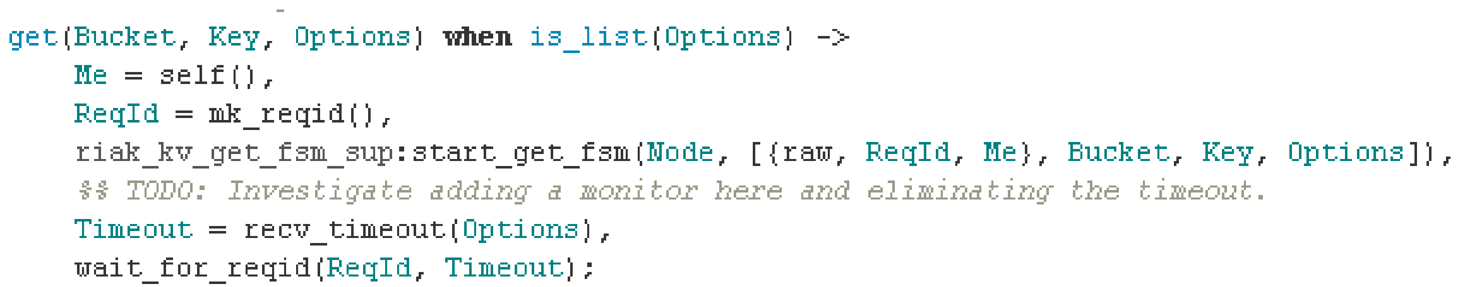
RIAK: WTF`S



Вывод - читаем mailing list и код =)

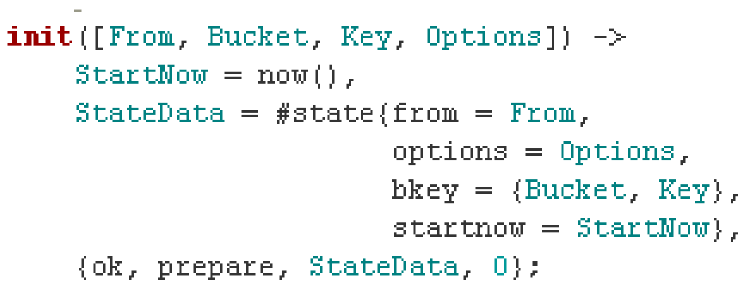
RIAK: CODE & COMMUNITY

https://github.com/basho/riak_kv/blob/master/src/riak_client.erl

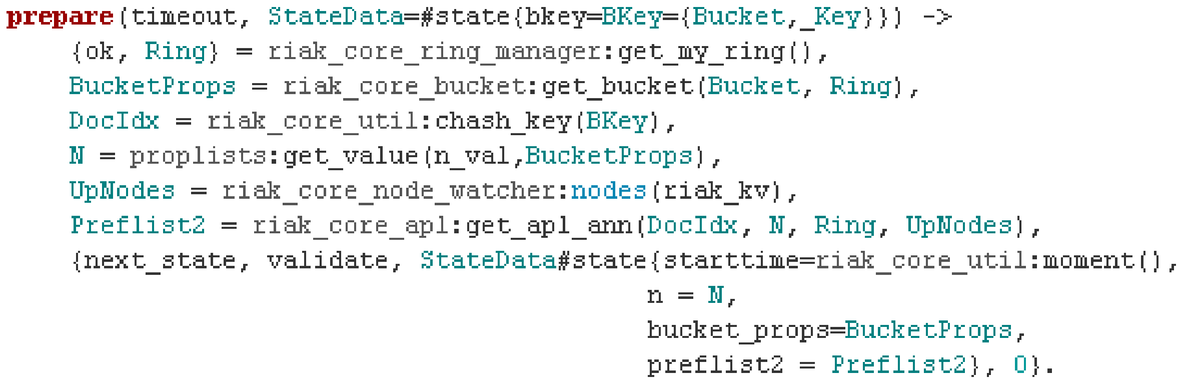


```
get(Bucket, Key, Options) when is_list(Options) ->
    Me = self(),
    ReqId = mk_reqid(),
    riak_kv_get_fsm_sup:start_get_fsm(Node, [{raw, ReqId, Me}, Bucket, Key, Options]),
    %% TODO: Investigate adding a monitor here and eliminating the timeout.
    Timeout = recv_timeout(Options),
    wait_for_reqid(ReqId, Timeout);
```

https://github.com/basho/riak_kv/blob/master/src/riak_kv_get_fsm.erl

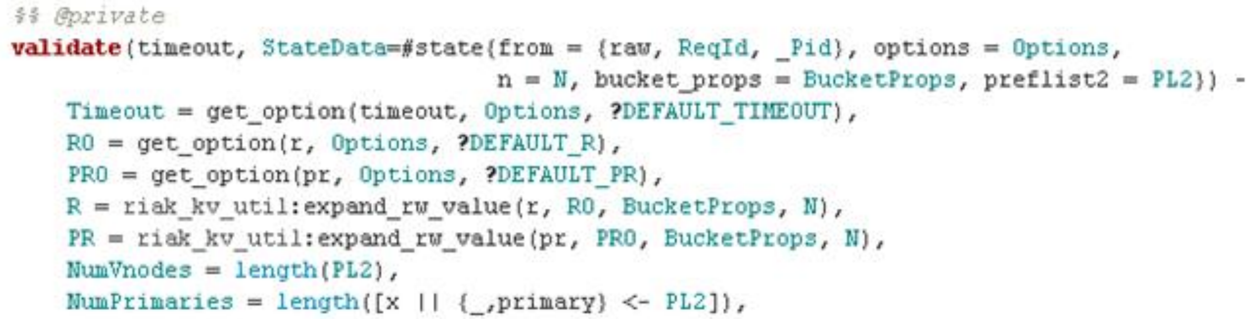


```
init([From, Bucket, Key, Options]) ->
    StartNow = now(),
    StateData = #state{from = From,
                       options = Options,
                       bkey = {Bucket, Key},
                       startnow = StartNow},
    {ok, prepare, StateData, 0};
```



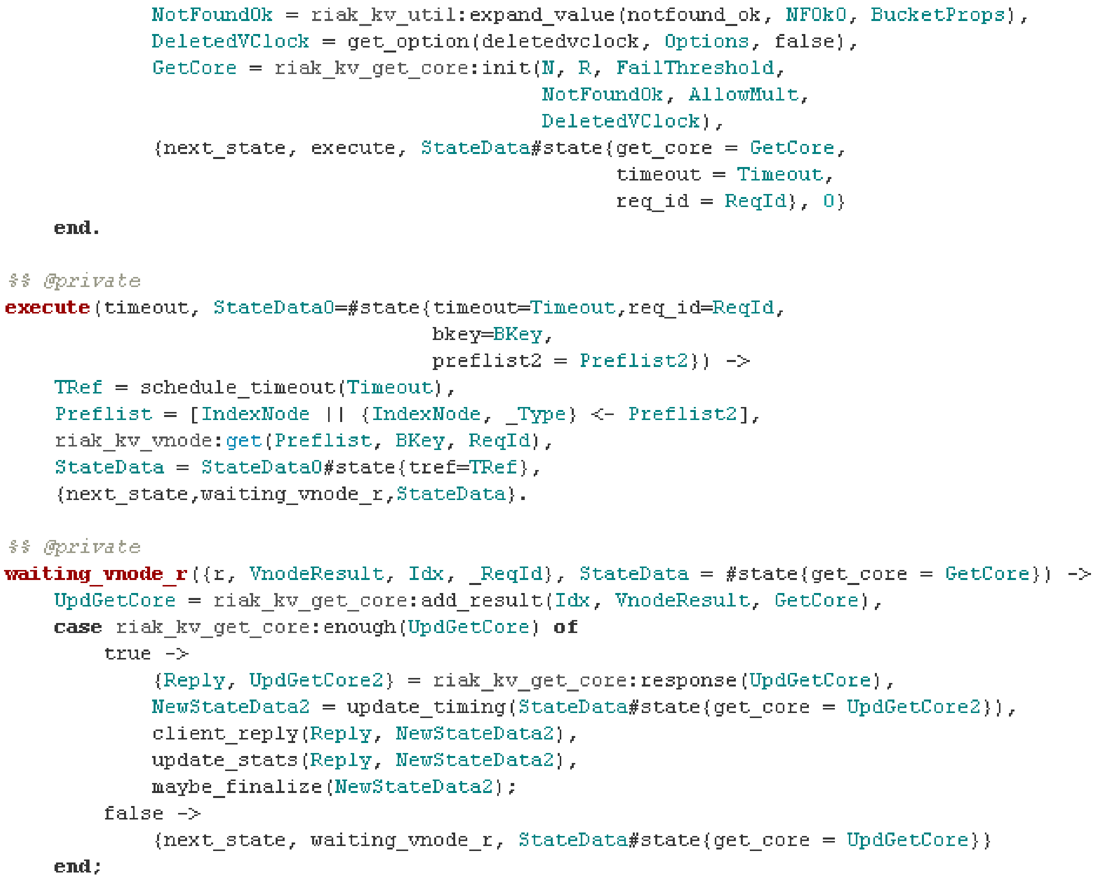
```
prepare(timeout, StateData=#state{bkey=BKey={Bucket,_Key}}) ->
    {ok, Ring} = riak_core_ring_manager:get_my_ring(),
    BucketProps = riak_core_bucket:get_bucket(Bucket, Ring),
    DocIdx = riak_core_util:chash_key(BKey),
    N = proplists:get_value(n_val,BucketProps),
    UpNodes = riak_core_node_watcher:nodes(riak_kv),
    Preflist2 = riak_core_apl:get_apl_ann(DocIdx, N, Ring, UpNodes),
    {next_state, validate, StateData#state{starttime=riak_core_util:moment(),
                                           n = N,
                                           bucket_props=BucketProps,
                                           preflist2 = Preflist2}, 0}.
```

RIAK: CODE & COMMUNITY



```
%% @private
validate(timeout, StateData=#state{from = {raw, ReqId, _Pid}, options = Options,
                                   n = N, bucket_props = BucketProps, preflist2 = PL2}) -
    Timeout = get_option(timeout, Options, ?DEFAULT_TIMEOUT),
    R0 = get_option(r, Options, ?DEFAULT_R),
    PR0 = get_option(pr, Options, ?DEFAULT_PR),
    R = riak_kv_util:expand_rw_value(r, R0, BucketProps, N),
    PR = riak_kv_util:expand_rw_value(pr, PR0, BucketProps, N),
    NumVnodes = length(PL2),
    NumPrimaries = length([x || {_,primary} <- PL2]),
```

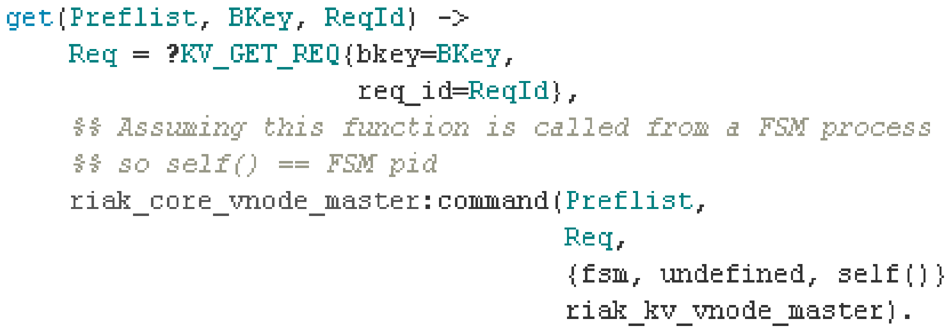
. . . .



```
            NotFoundOk = riak_kv_util:expand_value(notfound_ok, NFOk0, BucketProps),
            DeletedVClock = get_option(deletedvclock, Options, false),
            GetCore = riak_kv_get_core:init(N, R, FailThreshold,
                                            NotFoundOk, AllowMult,
                                            DeletedVClock),
            {next_state, execute, StateData#state{get_core = GetCore,
                                                  timeout = Timeout,
                                                  req_id = ReqId}, 0}
    end.

%% @private
execute(timeout, StateData0=#state{timeout=Timeout,req_id=ReqId,
                                   bkey=BKey,
                                   preflist2 = Preflist2}) ->
    TRef = schedule_timeout(Timeout),
    Preflist = [IndexNode || {IndexNode, _Type} <- Preflist2],
    riak_kv_vnode:get(Preflist, BKey, ReqId),
    StateData = StateData0#state{tref=TRef},
    {next_state,waiting_vnode_r,StateData}.

%% @private
waiting_vnode_r({r, VnodeResult, Idx, _ReqId}, StateData = #state{get_core = GetCore}) ->
    UpdGetCore = riak_kv_get_core:add_result(Idx, VnodeResult, GetCore),
    case riak_kv_get_core:enough(UpdGetCore) of
        true ->
            {Reply, UpdGetCore2} = riak_kv_get_core:response(UpdGetCore),
            NewStateData2 = update_timing(StateData#state{get_core = UpdGetCore2}),
            client_reply(Reply, NewStateData2),
            update_stats(Reply, NewStateData2),
            maybe_finalize(NewStateData2);
        false ->
            {next_state, waiting_vnode_r, StateData#state{get_core = UpdGetCore}}
    end;
```

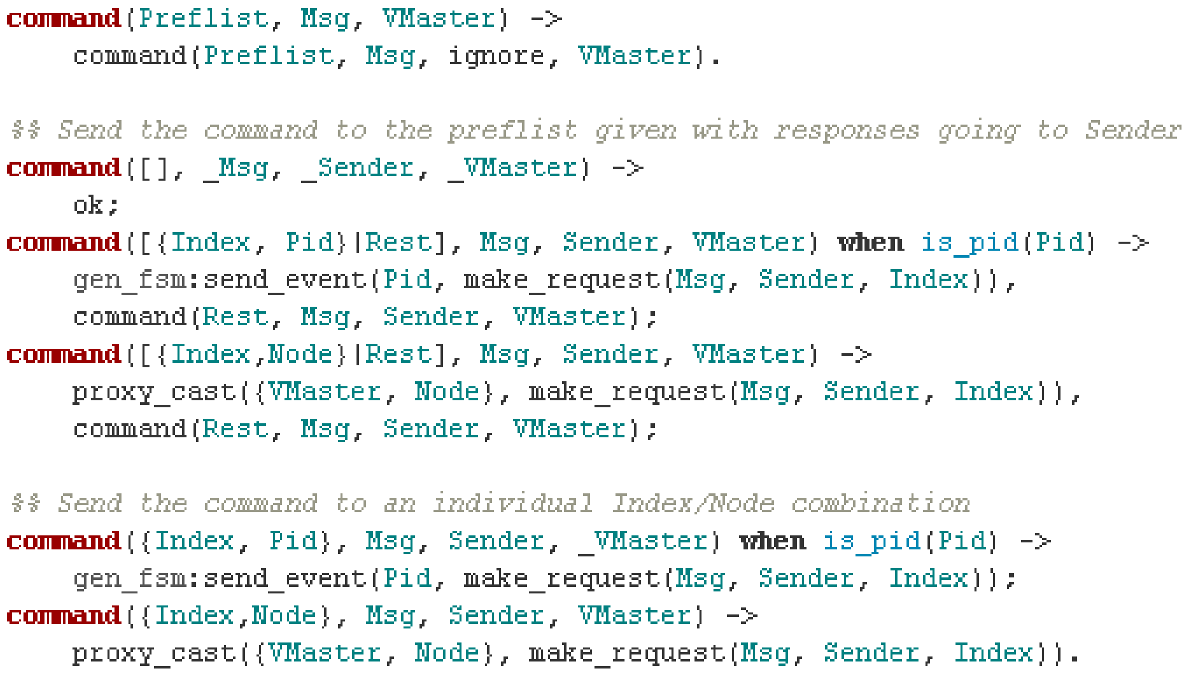
RIAK: CODE & COMMUNITY

https://github.com/basho/riak_kv/blob/master/src/riak_kv_vnode.erl



```
get(Preflist, BKey, ReqId) ->
    Req = ?KV_GET_REQ{bkey=BKey,
                      req_id=ReqId},
    %% Assuming this function is called from a FSM process
    %% so self() == FSM pid
    riak_core_vnode_master:command(Preflist,
                                   Req,
                                   {fsm, undefined, self()},
                                   riak_kv_vnode_master).
```

https://github.com/basho/riak_core/blob/master/src/riak_core_vnode_master.erl



```
command(Preflist, Msg, VMaster) ->
    command(Preflist, Msg, ignore, VMaster).

%% Send the command to the preflist given with responses going to Sender
command([], _Msg, _Sender, _VMaster) ->
    ok;
command([{Index, Pid}|Rest], Msg, Sender, VMaster) when is_pid(Pid) ->
    gen_fsm:send_event(Pid, make_request(Msg, Sender, Index)),
    command(Rest, Msg, Sender, VMaster);
command([{Index,Node}|Rest], Msg, Sender, VMaster) ->
    proxy_cast({VMaster, Node}, make_request(Msg, Sender, Index)),
    command(Rest, Msg, Sender, VMaster);

%% Send the command to an individual Index/Node combination
command({Index, Pid}, Msg, Sender, _VMaster) when is_pid(Pid) ->
    gen_fsm:send_event(Pid, make_request(Msg, Sender, Index));
command({Index,Node}, Msg, Sender, VMaster) ->
    proxy_cast({VMaster, Node}, make_request(Msg, Sender, Index)).
```

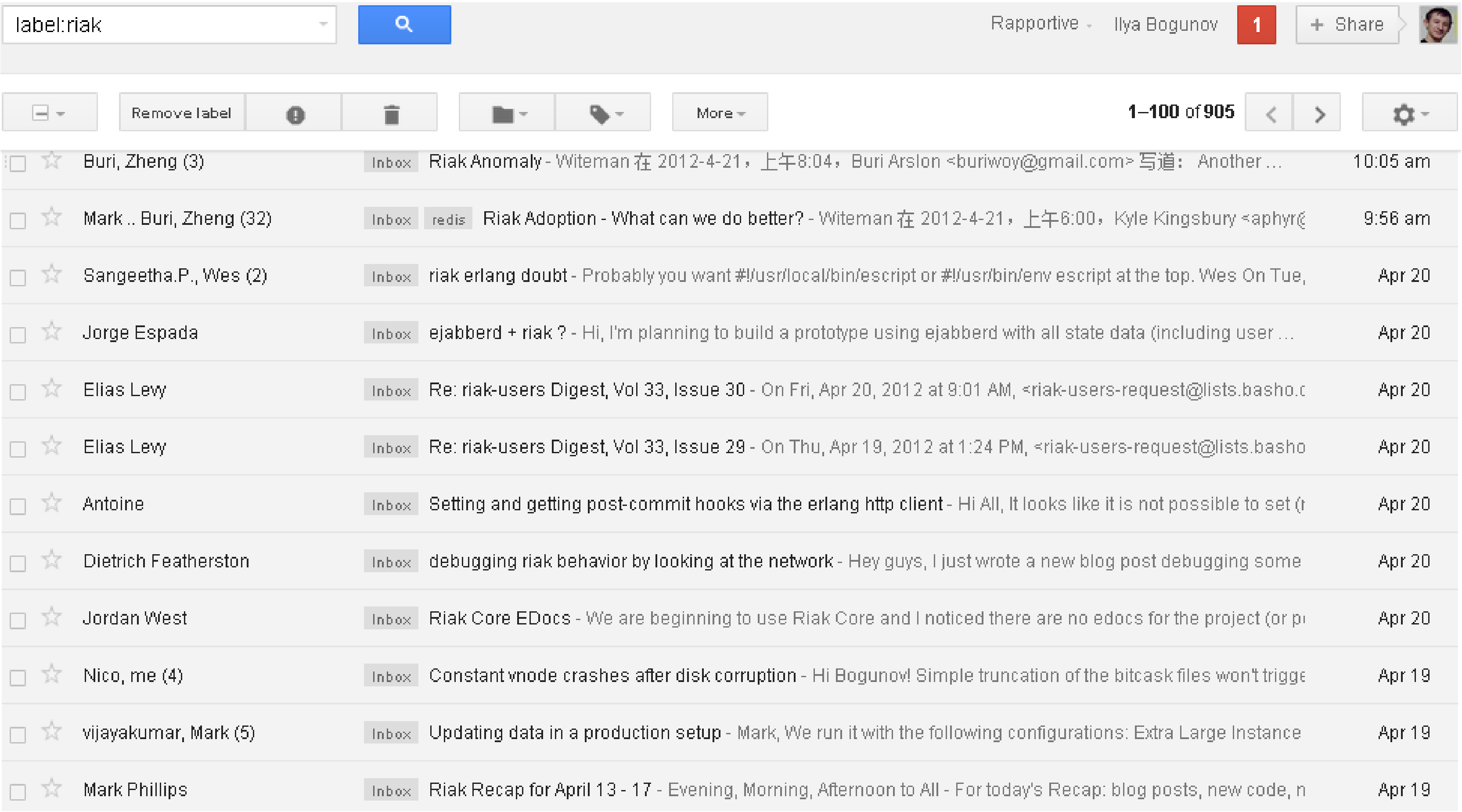
RIAK: CODE & COMMUNITY

https://github.com/basho/riak_kv/blob/master/src/riak_kv_eleveldb_backend.erl

```
get(Bucket, Key, #state{read_opts=ReadOpts,
                        ref=Ref}=State) ->
    StorageKey = to_object_key(Bucket, Key),
    case eleveldb:get(Ref, StorageKey, ReadOpts) of
        {ok, Value} ->
            {ok, Value, State};
        not_found  ->
            {error, not_found, State}
        {error, Reason} ->
            {error, Reason, State}
    end.
```

RIAK: CODE & COMMUNITY

Сообщество живое, иногда даже излишне)

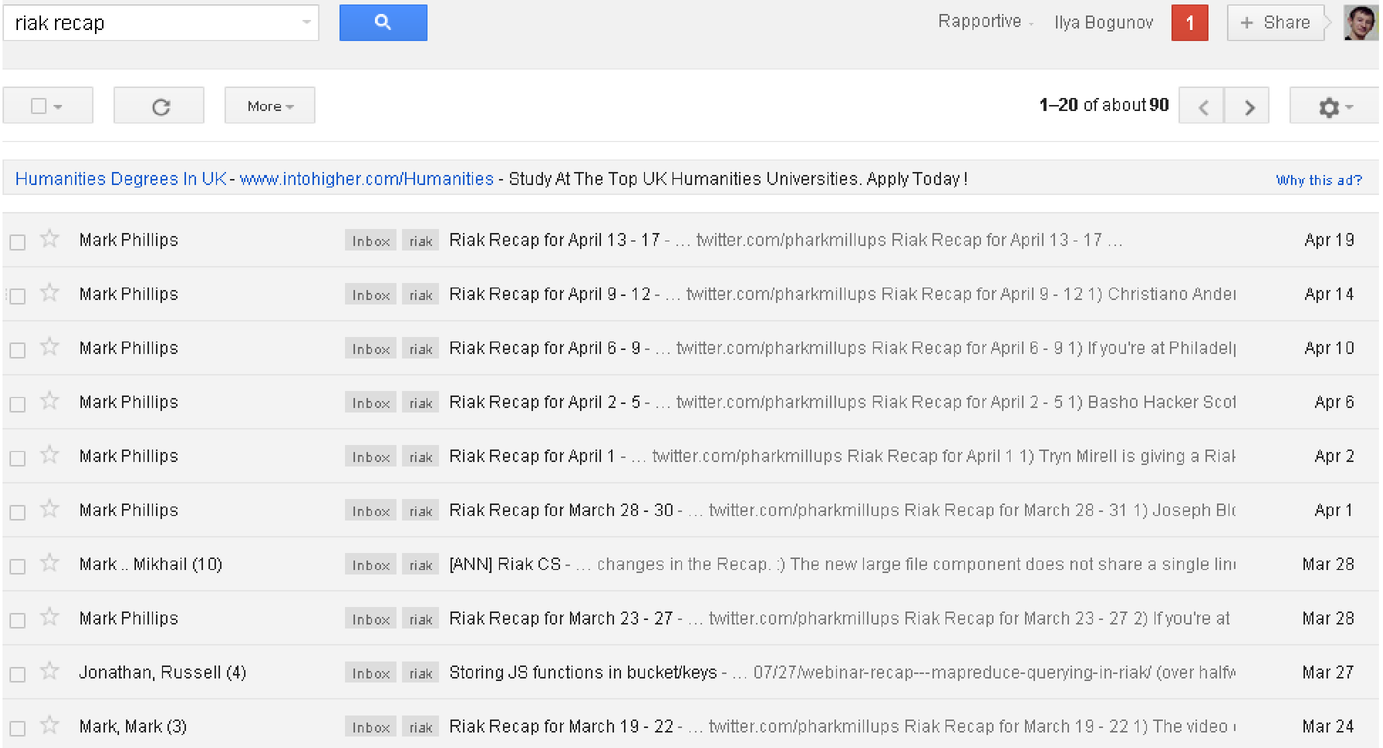


label:riak

Buri, Zheng (3)	Inbox	Riak Anomaly - Witeman 在 2012-4-21，上午8:04，Buri Arslon <buriwoy@gmail.com> 写道： Another ...	10:05 am
Mark .. Buri, Zheng (32)	Inbox redis	Riak Adoption - What can we do better? - Witeman 在 2012-4-21，上午6:00，Kyle Kingsbury <aphyr@	9:56 am
Sangeetha.P., Wes (2)	Inbox	riak erlang doubt - Probably you want #!/usr/local/bin/escript or #!/usr/bin/env escript at the top. Wes On Tue,	Apr 20
Jorge Espada	Inbox	ejabberd + riak ? - Hi, I'm planning to build a prototype using ejabberd with all state data (including user ...	Apr 20
Elias Levy	Inbox	Re: riak-users Digest, Vol 33, Issue 30 - On Fri, Apr 20, 2012 at 9:01 AM, <riak-users-request@lists.basho.c	Apr 20
Elias Levy	Inbox	Re: riak-users Digest, Vol 33, Issue 29 - On Thu, Apr 19, 2012 at 1:24 PM, <riak-users-request@lists.basho	Apr 20
Antoine	Inbox	Setting and getting post-commit hooks via the erlang http client - Hi All, It looks like it is not possible to set (i	Apr 20
Dietrich Featherston	Inbox	debugging riak behavior by looking at the network - Hey guys, I just wrote a new blog post debugging some	Apr 20
Jordan West	Inbox	Riak Core EDocs - We are beginning to use Riak Core and I noticed there are no edocs for the project (or pi	Apr 20
Nico, me (4)	Inbox	Constant vnode crashes after disk corruption - Hi Bogunov! Simple truncation of the bitcask files won't trigge	Apr 19
vijayakumar, Mark (5)	Inbox	Updating data in a production setup - Mark, We run it with the following configurations: Extra Large Instance	Apr 19
Mark Phillips	Inbox	Riak Recap for April 13 - 17 - Evening, Morning, Afternoon to All - For today's Recap: blog posts, new code, n	Apr 19

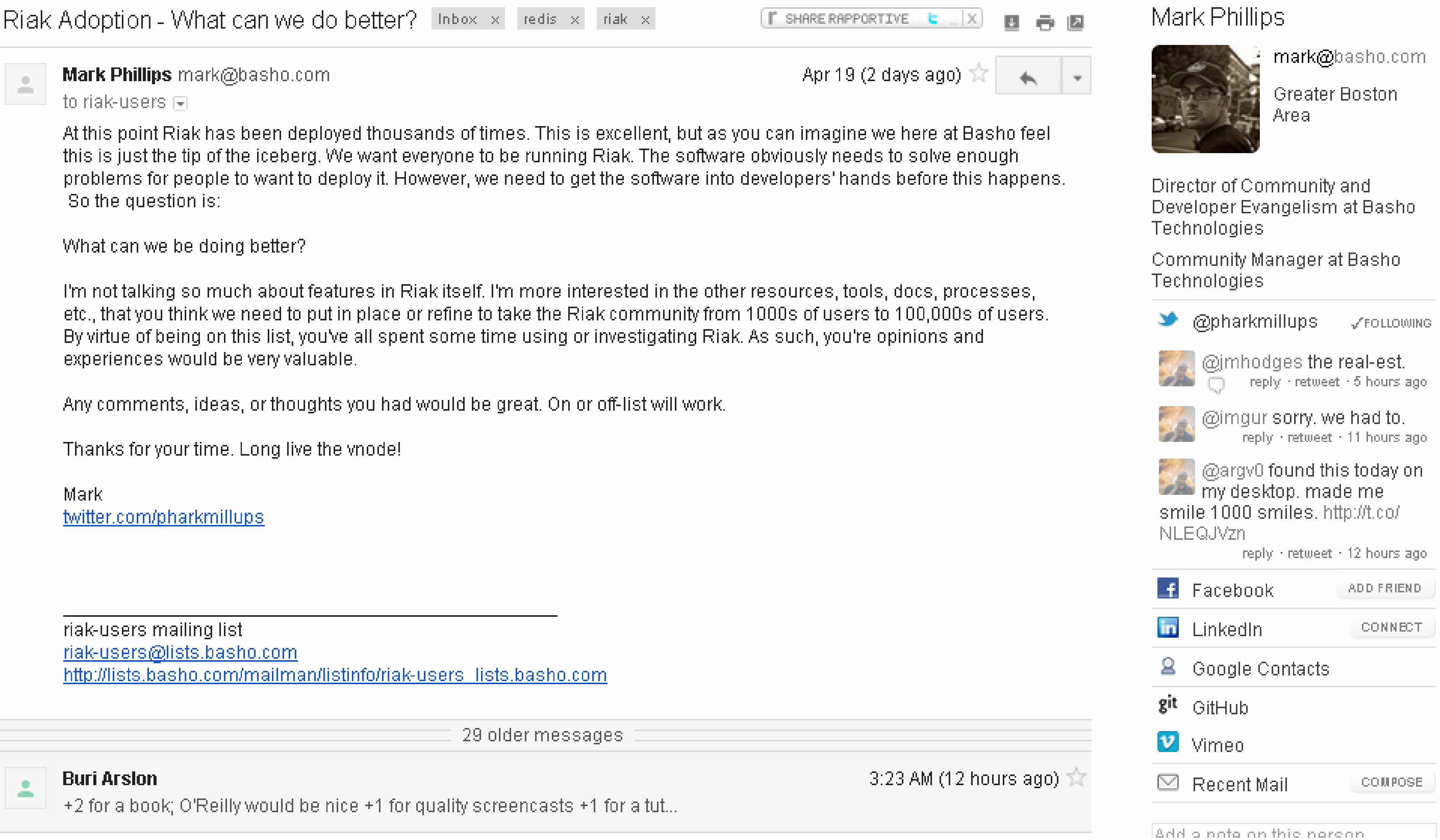
RIAK: CODE & COMMUNITY

- За ним стоит коммерческая компания (с инвестициями), и в ней Eric Brewer =)
- Еженедельные дайджесты «что нового»



□ ☆	Mark Phillips	Inbox	riak	Riak Recap for April 13 - 17 - ... twitter.com/pharkmillups Riak Recap for April 13 - 17 ...	Apr 19
□ ☆	Mark Phillips	Inbox	riak	Riak Recap for April 9 - 12 - ... twitter.com/pharkmillups Riak Recap for April 9 - 12 1) Christiano Ander	Apr 14
□ ☆	Mark Phillips	Inbox	riak	Riak Recap for April 6 - 9 - ... twitter.com/pharkmillups Riak Recap for April 6 - 9 1) If you're at Philadel	Apr 10
□ ☆	Mark Phillips	Inbox	riak	Riak Recap for April 2 - 5 - ... twitter.com/pharkmillups Riak Recap for April 2 - 5 1) Basho Hacker Scot	Apr 6
□ ☆	Mark Phillips	Inbox	riak	Riak Recap for April 1 - ... twitter.com/pharkmillups Riak Recap for April 1 1) Tryn Mirell is giving a Rial	Apr 2
□ ☆	Mark Phillips	Inbox	riak	Riak Recap for March 28 - 30 - ... twitter.com/pharkmillups Riak Recap for March 28 - 31 1) Joseph Bl	Apr 1
□ ☆	Mark .. Mikhail (10)	Inbox	riak	[ANN] Riak CS - ... changes in the Recap. :) The new large file component does not share a single lin	Mar 28
□ ☆	Mark Phillips	Inbox	riak	Riak Recap for March 23 - 27 - ... twitter.com/pharkmillups Riak Recap for March 23 - 27 2) If you're at	Mar 28
□ ☆	Jonathan, Russell (4)	Inbox	riak	Storing JS functions in bucket/keys - ... 07/27/webinar-recap---mapreduce-querying-in-riak/ (over halfv	Mar 27
□ ☆	Mark, Mark (3)	Inbox	riak	Riak Recap for March 19 - 22 - ... twitter.com/pharkmillups Riak Recap for March 19 - 22 1) The video	Mar 24

RIAK: CODE & COMMUNITY



Riak Adoption - What can we do better?

Mark Phillips mark@basho.com
to riak-users
Apr 19 (2 days ago)

At this point Riak has been deployed thousands of times. This is excellent, but as you can imagine we here at Basho feel this is just the tip of the iceberg. We want everyone to be running Riak. The software obviously needs to solve enough problems for people to want to deploy it. However, we need to get the software into developers' hands before this happens. So the question is:

What can we be doing better?

I'm not talking so much about features in Riak itself. I'm more interested in the other resources, tools, docs, processes, etc., that you think we need to put in place or refine to take the Riak community from 1000s of users to 100,000s of users. By virtue of being on this list, you've all spent some time using or investigating Riak. As such, you're opinions and experiences would be very valuable.

Any comments, ideas, or thoughts you had would be great. On or off-list will work.

Thanks for your time. Long live the vnode!

Mark
twitter.com/pharkmillups

riak-users mailing list
riak-users@lists.basho.com
http://lists.basho.com/mailman/listinfo/riak-users_lists.basho.com

29 older messages

Buri Arslon
3:23 AM (12 hours ago)
+2 for a book; O'Reilly would be nice +1 for quality screencasts +1 for a tut...

Mark Phillips

mark@basho.com

Greater Boston Area

Director of Community and Developer Evangelism at Basho Technologies

Community Manager at Basho Technologies

@pharkmillups

@jmhodges the real-est.
reply · retweet · 5 hours ago

@imgur sorry. we had to.
reply · retweet · 11 hours ago

@argv0 found this today on my desktop. made me smile 1000 smiles. http://t.co/NLEQJVzn
reply · retweet · 12 hours ago

Facebook

LinkedIn

Google Contacts

GitHub

Vimeo

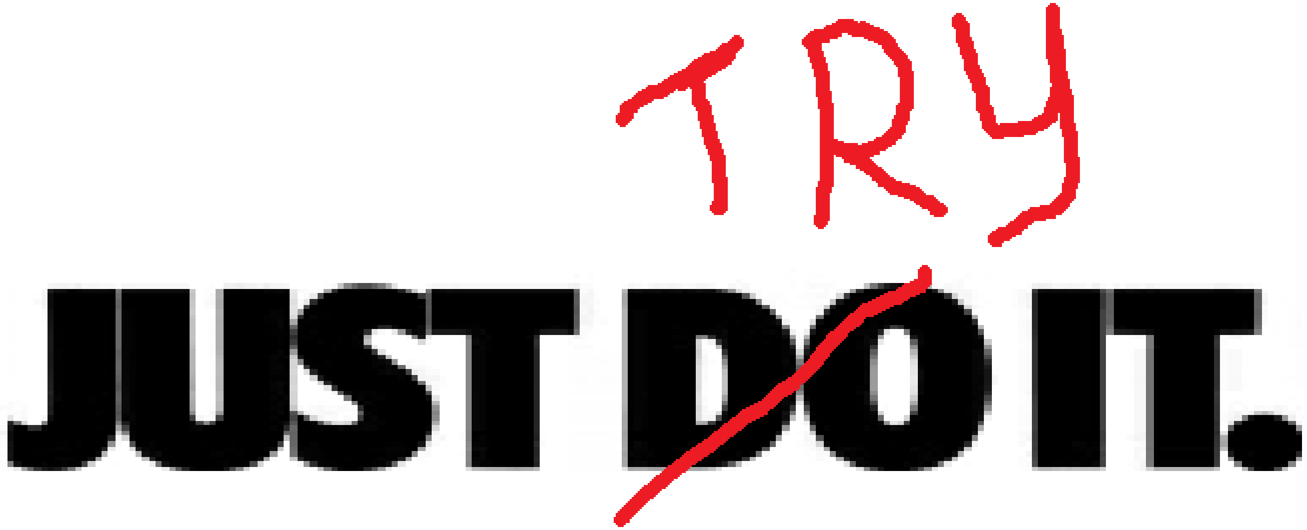
Recent Mail

Add a note on this person

RIAK: Стоит или не стоит ?

- У вас есть проблема с количеством IOPS.
- Вам важна высокодоступность и хочется минимум проблем при расширение кластера
- Вы готовы заранее полностью продумать то, как ваши данные положить на kv
- Вы знаете, как будете решать конфликты
- Вам не нужны транзакции (pessimistic\optimistic locking)
- Вы не боитесь erlang`a =) И не боитесь обратиться в community, чтобы получить ответ на свой вопрос

RIAK: Стоит или не стоит ?



Спасибо за внимание\Вопросы.

Илья Богунов

bogunov@gmail.com

twitter.com/tech_mind

Пожалуйста, поставьте оценку моему докладу.

Ваше мнение очень важно.

Спасибо!