SCM Dashboard

Monitoring Code Velocity at the Product / Project / Branch level

Prakash Ranade

vmware®

2011 PERFORCE USER CONFERENCE

AGENDA

- What is SCM Dashboard?
- Why is SCM Dashboard needed?
- Where is it used?
- How does it look?
- Challenges in building SCM Dashboard
- Goals in designing SCM Dashboard
- Technology in building SCM Dashboard
- Conclusion

What is SCM Dashboard?

- A framework for organizing, automating, and analyzing software configuration methodologies, metrics, processes, and systems that drive product release performance.

- The Dashboard gathers, organizes, and stores information from various internal data sources and displays metrics that are the result of simple or complex calculations with minimal processing time.

- Decision support system that provides historical data and current trends in its portlet region, showing metrics/reports side-by-side on the same web page.

Why is SCM Dashboard needed?

You are not able to manage what you can not measure.

- The Dashboard is an easy way to enhance visibility on the product releases, such as showing how you do compared to previous performances, goals and benchmarks.

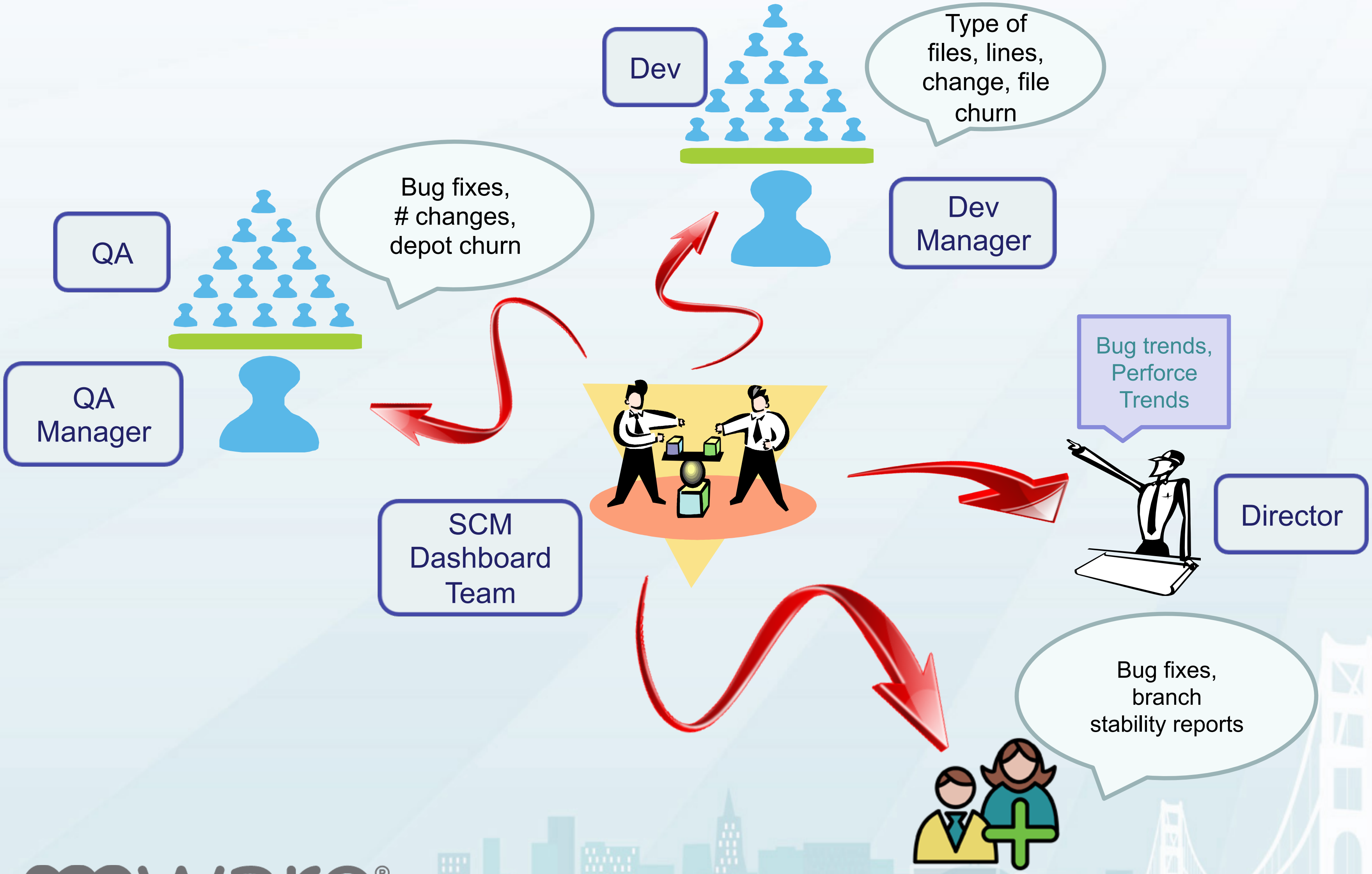
What gets watched, will get done.

- Ability to make more informed decisions based on multiple reports.

Not only for the executives, but for all levels of engineering.

- Release Manager, Director
- Development, QA Manager,
- Developer, QA

Who needs metrics?



Dev

Type of files, lines, change, file churn

Dev Manager

QA

Bug fixes, # changes, depot churn

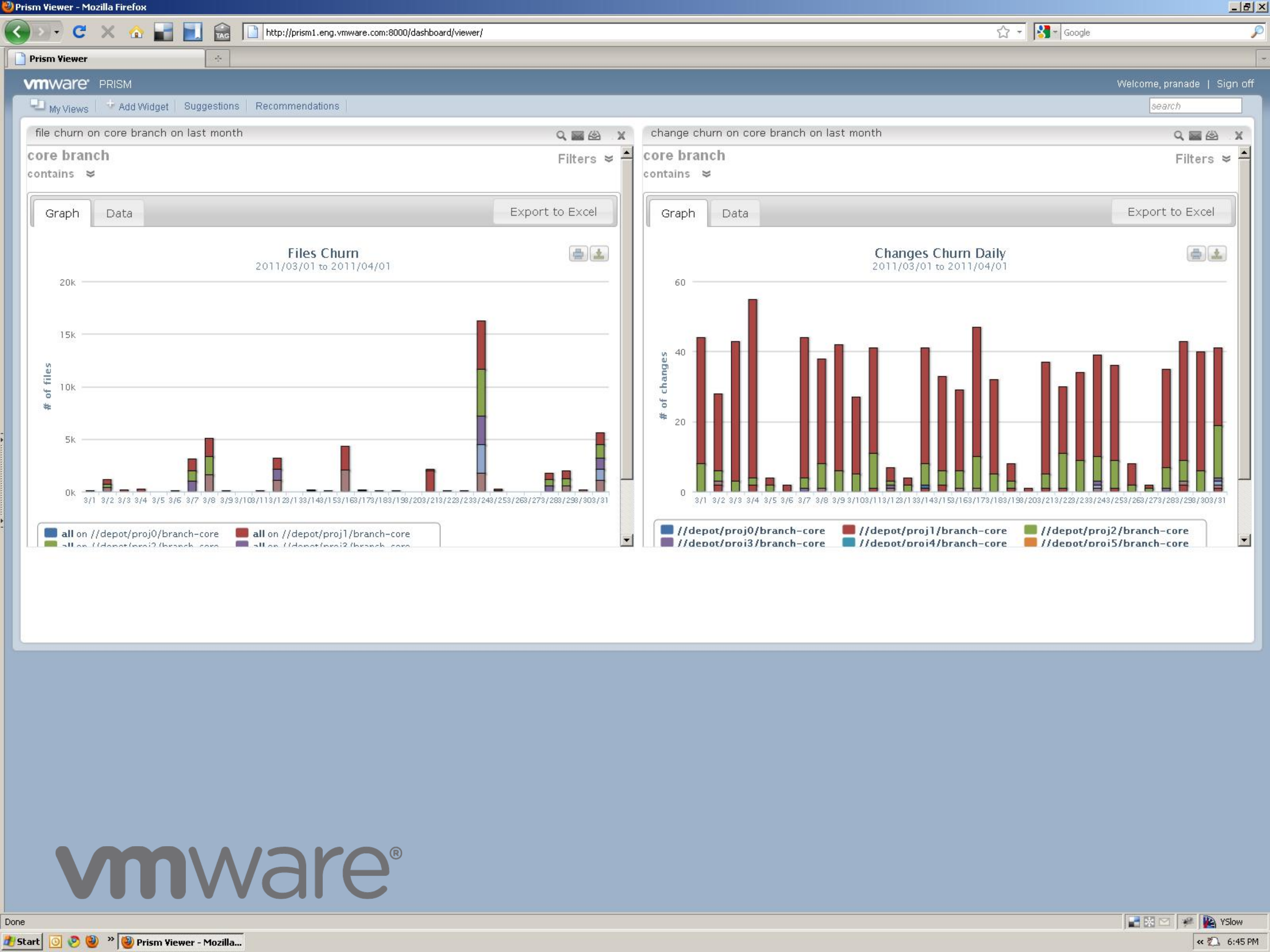
QA Manager

Bug trends, Perforce Trends

Director

SCM Dashboard Team

Bug fixes, branch stability reports



file churn on core branch on last month

core branch

contains

Files Churn

2011/03/01 to 2011/04/01

🟦 **all** on //depot/proj0/branch-core 🟥 **all** on //depot/proj1/branch-core
🟩 all on //depot/proj2/branch-core 🟪 all on //depot/proj3/branch-core

change churn on core branch on last month

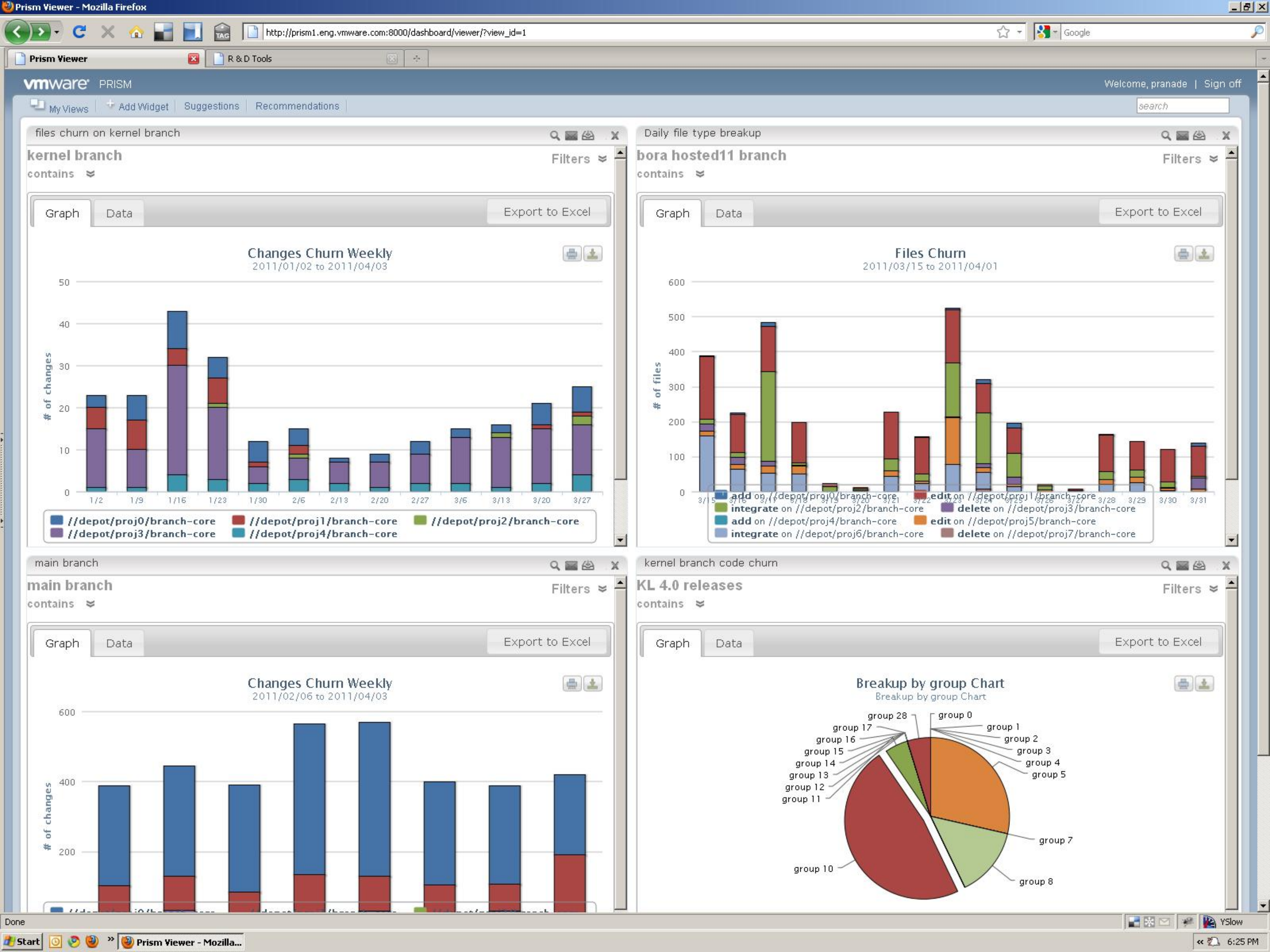
core branch

contains

Changes Churn Daily

2011/03/01 to 2011/04/01

🟦 //depot/proj0/branch-core 🟥 //depot/proj1/branch-core 🟩 //depot/proj2/branch-core
🟪 //depot/proj3/branch-core 🟦 //depot/proj4/branch-core 🟧 //depot/proj5/branch-core



Prism Viewer - Mozilla Firefox
http://prism1.eng.vmware.com:8000/dashboard/viewer/?view_id=1
Google
Prism Viewer
R & D Tools
vmware PRISM
Welcome, pranade | Sign off
My Views
Add Widget
Suggestions
Recommendations
search

files churn on kernel branch

kernel branch

contains

Filters

Graph | Data | Export to Excel

Changes Churn Weekly

2011/01/02 to 2011/04/03

- 🟦 //depot/proj0/branch-core
- 🟥 //depot/proj1/branch-core
- 🟩 //depot/proj2/branch-core
- 🟪 //depot/proj3/branch-core
- 🟦 //depot/proj4/branch-core

Daily file type breakup

bora hosted11 branch

contains

Filters

Graph | Data | Export to Excel

Files Churn

2011/03/15 to 2011/04/01

- 🟦 **add** on //depot/proj0/branch-core
- 🟥 **edit** on //depot/proj1/branch-core
- 🟩 **integrate** on //depot/proj2/branch-core
- 🟪 **delete** on //depot/proj3/branch-core
- 🟦 **add** on //depot/proj4/branch-core
- 🟧 **edit** on //depot/proj5/branch-core
- 🟦 **integrate** on //depot/proj6/branch-core
- 🟫 **delete** on //depot/proj7/branch-core

main branch

main branch

contains

Filters

Graph | Data | Export to Excel

Changes Churn Weekly

2011/02/06 to 2011/04/03

kernel branch code churn

KL 4.0 releases

contains

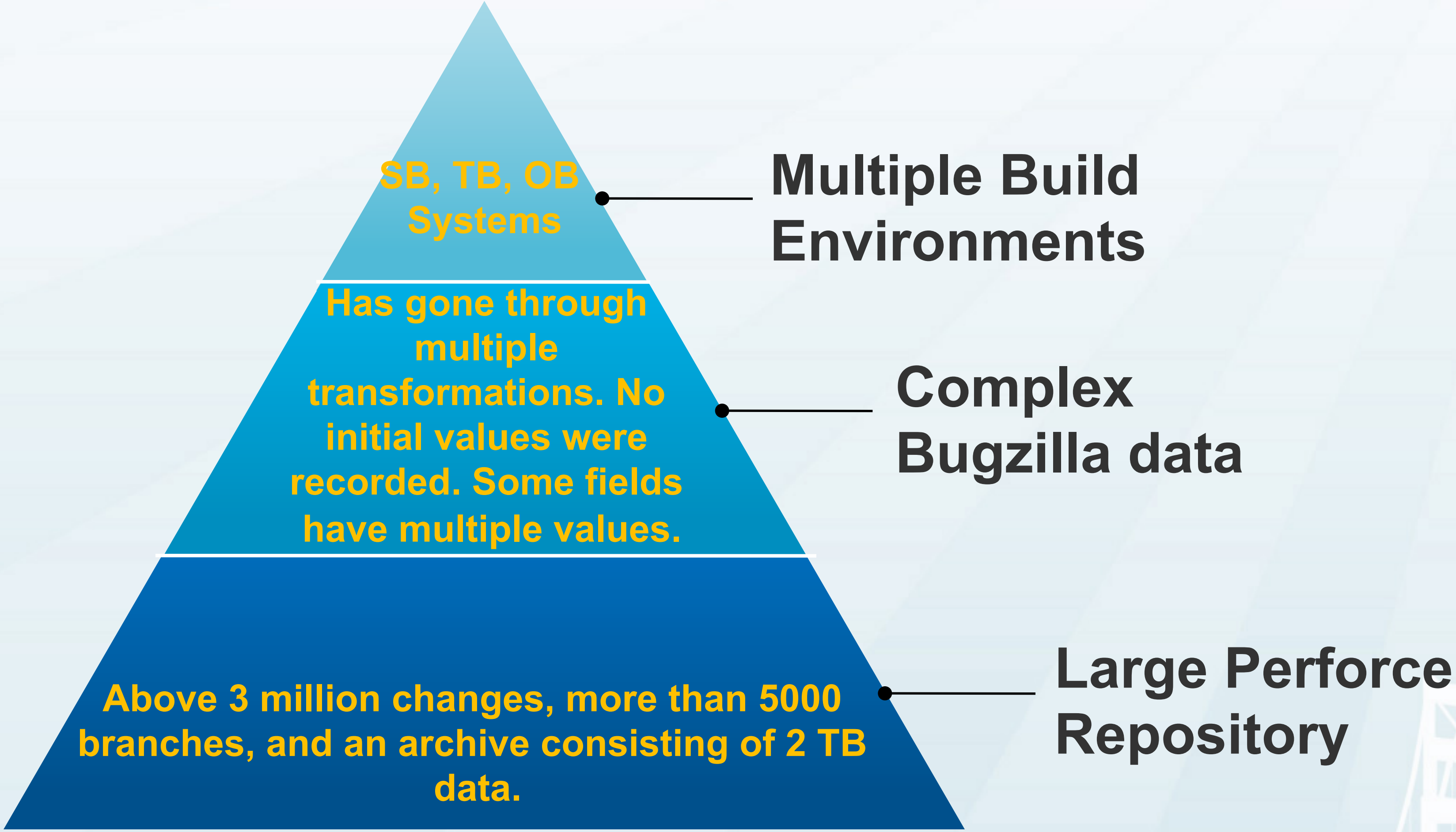
Filters

Graph | Data | Export to Excel

Breakup by group Chart

Breakup by group Chart

Data challenges



SB, TB, OB Systems

Multiple Build Environments

Has gone through multiple transformations. No initial values were recorded. Some fields have multiple values.

Complex Bugzilla data

Above 3 million changes, more than 5000 branches, and an archive consisting of 2 TB data.

Large Perforce Repository

Dashboard Goals

Speed

- Max. 5 seconds response time for the requests
- Provides frequent, or at least daily, updates
- Bases project status on incremental data updates

Sharing

- Social Engineering
- Easy to share charts and reports among team members
- Easy to make project dashboards

Portal

- Ability to configure multiple metrics on a single page.
- Ability to fine tune settings and filters on charts and reports.
- Ability to drill downs and form aggregations.

Building blocks

python™

jQuery
write less, do more.

django

hadoop

mongoDB

vmware®

An Architecture based on Hadoop and MongoDB

- Hadoop is a open-source software used for breaking a big job into smaller tasks, performing each task and collecting the results.
- MapReduce is a programming model for data processing, working by breaking the processing into two phases, a map phase and a reduce phase.
- Hadoop streaming is a utility that comes with the distribution, allowing you to create and run MapReduce jobs in Python.
- The HDFS is a filesystem that stores large files across multiple machines and achieves reliability by replicating the data across multiple hosts.
- MongoDB is a document based database system. Each document can be thought of as a large hash object. There are keys(columns) with values which can be anything such as hashes, arrays, numbers, serialized objects, etc.

Perforce Branch:

Our Perforce branch exists on multiple perforce servers. Our branch specification looks like this.

- **server1:1666**

  //depot/<component>/<old-branch>/… //depot/<component>/<new-branch>/…

- **server2:1666**

  //depot/<component2>/<old-branch>/… //depot/<component2>/<new-branch>/…

  //depot/<component3>/<old-branch>/… //depot/<component3>/<new-branch>/…

- **server3:1666**

  //depot/<component4>/<old-branch>/… //depot/<component4>/<new-branch>/…

Branch policies

- Branch Manager identifies and lists new feature/bugs, improvements in Bugzilla and Perforce BMPS, and then sets the check-in policies on the branch and change specification forms.

```
Change 1359870 by pranade@pranade-prism1 on 2011/04/27 17:31:36
        Implement Prism View...
        QA Notes:
        Testing Done: Perforce Create, Update, delete view
        Bug Number: 703648, 703649
        Approved by: daf
        Reviewed by: gaddamk, akalaveshi
        Review URL: https://reviewboard.eng.vmware.com/r/227466/
        #You may set automerge requests to YES|NO|MANUAL below,
        #with at most one being set to YES.
        Merge to: MAIN: YES
        Merge to: Release: NO

Affected files ...

... //depot/component-1/branch-1/views.py#12 edit
... //depot/component-1/branch-1/templates/vcs/perforce.html#15 edit
... //depot/component-1/branch-1/tests.py#1 add
... //depot/component-1/branch-1/utils.py#14 delete

Differences ...
```

Perforce Data collection

- **"p4 describe" displays the details of the changeset, as follows:**

  The changelist number
  The changelist creator name and workspace name
  The date when the changelist created
  The changelist's description
  The submitted file lists and the code diffs

- **We have a Perforce data dumper script which connect to perforce servers and dumps the "p4 describe" output of the submitted changelist.**

- **The Perforce data dumper script dumps output in 64 MB file chunks, which are then copied to HDFS.**

MapReduce

• We have a Perforce data dumper script which connect to perforce servers and dumps the “p4 describe” output of the submitted changelist. Each MapReduce script scans all the information from a “p4 describe” output. The following reports can be created by writing different MapReduce scripts:

Number of submitted changes per depot path

File information like add, edit, integrate, branch, delete

File types such as “c”, “py”, “pl”, “java”, etc.

Number of lines added, removed, modified

Most revised files and least revised files

Bug number and bug status

Reviewers and test case information

Change submitter names and group mapping

Depot path and branch spec mapping

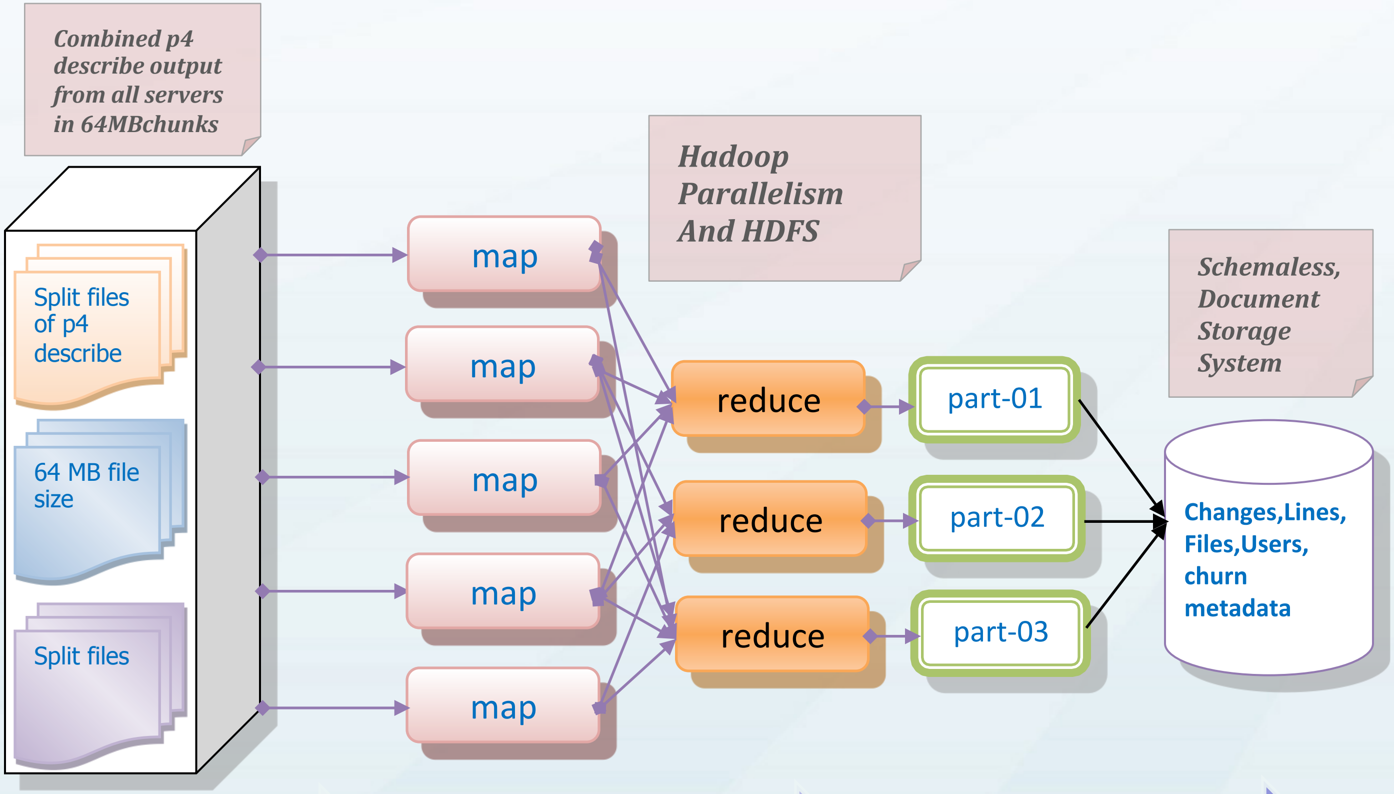
Python MapReduce

- MapReduce programs are much easier to develop in a scripting language using the Streaming API tool. Hadoop MapReduce provides automatic parallelization and distribution, fault-tolerance, and status and monitoring tools.

- Hadoop Streaming interacts with programs that use the Unix streaming paradigm. Inputs come in through STDIN and outputs go to STDOUT. The data has to be text based and each line is considered a record. The overall data flow in Hadoop streaming is like a pipe where data streams in through the mapper and the sorted output streams out through the reducer. In pseudo-code using Unix's command line notation, it comes up as the following:

```
cat [input_file] | [mapper] | sort | [reducer] > [output_file]
```

Process



Combined p4 describe output from all servers in 64MBchunks

Split files of p4 describe

64 MB file size

Split files

map

map

map

map

map

Hadoop Parallelism And HDFS

reduce

reduce

reduce

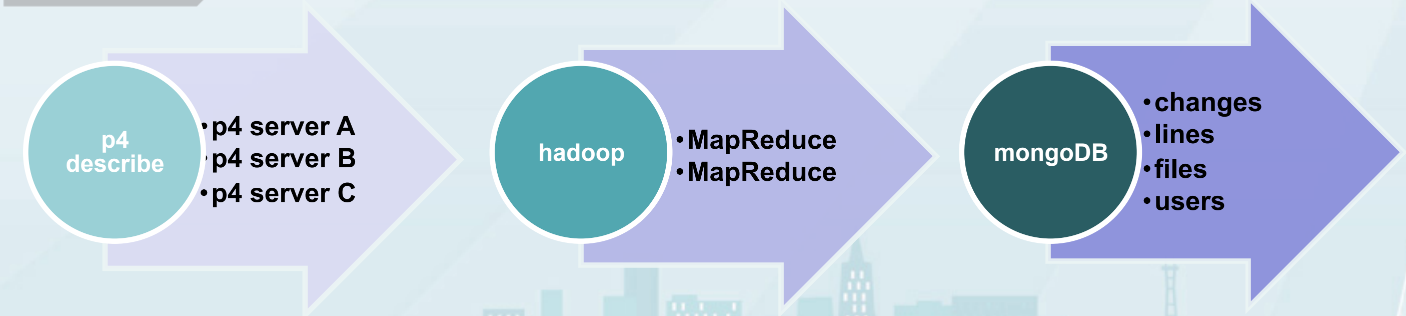
part-01

part-02

part-03

Schemaless, Document Storage System

Changes,Lines, Files,Users, churn metadata



p4 describe

- p4 server A
- p4 server B
- p4 server C

hadoop

- MapReduce
- MapReduce

mongoDB

- changes
- lines
- files
- users

```
def dump_to_reducer(srvr, chng, depotfiles):
    if srvr and depotfiles and chng:
        for filename in depotfiles:
            print "%s|%s\t%s" % (srvr, filename, str(chng))

 def main():
     chng, depot_files, l = 0, set(), os.linesep
     p4srvr = site_perforce_servers(site.perforce_servers)
     for line in sys.stdin:
        line = line.rstrip(l)
        if line and line.count('/')==80:
            srvr = match_begin_line(line, p4srvr)
            if srvr:
                chng, depot_files = 0, set()
                continue
        if line and line.count('%')==80:
            srvr = match_end_line(line, p4srvr)
            if srvr:
                dump_to_reducer(srvr, chng, depot_files)
                continue
         if line and line[0:7]=='Change ':
            chng = dtgrep(line)
            continue
         if line and line[0:6]=='... //':
            flgrep(line, depot_files)
```

Python
Mapper script

```
def main():
    depot2count = {}
    final_changes = {}
     for line in sys.stdin:
        try:
            p4srvr_depotpath, date_chng = line.split('\t',1)
        except:
            continue
         if (not p4srvr_depotpath) and (not date_chng):
            print >> sys.stderr, line
            continue
        dt, change = date_chng.split('.')
        change = change.rstrip(l)
        depot_hash = depot2count.setdefault
(p4srvr_depotpath,{})
        depot_hash.setdefault(dt,0)
        chng_set = depot2count[p4srvr_depotpath][dt]
        depot2count[p4srvr_depotpath][dt] = int(change)
     for (p4srvr_depotpath, dt) in depot2count.items():
        for (dt, chngset) in dt.items():
            print json.dumps
({'p4srvr_depotpath':p4srvr_depotpath, 'date': dt,
'changes': chngset})
```

Python
Reducer script

```
        mdb = mongo_utils.Vcs_Stats(collection_name="depot_churn")

        mdb.collection.create_index([('p4srvr_depotpath', pymongo.ASCENDING), ('date',
pymongo.ASCENDING)])

        for line in datafile.readlines():
                data = json.loads(line)
                p4srvr_depotpath = "%s" % data['p4srvr_depotpath']
                dstr = data['date']
                yy, mm, dd, hh, MM, ss = dstr[0:4], dstr[4:6], dstr[6:8], dstr[8:10], dstr[10:12], dstr
[12:14]
                changes = data['changes']
                new_data = []
                mongo_data = {'p4srvr_depotpath':p4srvr_depotpath,
                                        'date‘:datetime.datetime(yy,mm,dd,hh,MM,ss),
                                            'changes':changes, '_id':"%s/%s:%s"%
(p4srvr_depotpath,dstr,changes)}
                mdb.collection.insert(mongo_data)
         mdb.collection.ensure_index([('p4srvr_depotpath', pymongo.ASCENDING), ('date',
pymongo.ASCENDING)])
```

mongodb
upload script

```
/* 0 */
{
  "_id": "perforce-server1:1666|//depot/component-1/branch-1/20110204005204:1290141",
  "date": "Thu, 03 Feb 2011 16:52:04 GMT -08:00",
  "p4srvr_depotpath": "perforce-server1:1666|//depot/component-1/esx41p01-hp4/",
  "changes": 1290141,
  "user": "pranade",
  "total_dict": {
    "all": "9",
    "branch": "9"
  }
}
/* 1 */
{
  "_id": "perforce-server1:1666|//depot/component-2/branch-2/20100407144638:1029666",
  "date": "Wed, 07 Apr 2010 07:46:38 GMT -07:00",
  "p4srvr_depotpath": "perforce-server1:1666|//depot/component-2/branch-2/",
  "changes": 1029666,
  "user": "akalaveshi",
  "total_dict": {
    "edit": "3",
    "all": "3"
  }
}
/* 2 */
{
  "_id": "perforce-server1:1666|//depot/component-2/branch-2/20100106003808:976075",
  "date": "Tue, 05 Jan 2010 16:38:08 GMT -08:00",
  "p4srvr_depotpath": "perforce-server1:1666|//depot/component-2/branch-2/",
  "changes": 976075,
  "user": "pranade",
  "total_dict": {
    "integrate": "10",
    "edit": "2",
    "all": "12"
  }
}
```

mongodb data

Conclusion

- We have designed a framework called SCM Dashboard.

- “p4 describe” command contains most of the information.

- Hadoop: horizontally scalable computational solution. Streaming makes MapReduce programming easy.

- Mongodb: Document model, dynamic queries, comprehensive data models.

QUESTIONS?

vmware®