

PostgreSQL Directions



***8.1
and
Beyond***

PostgreSQL 8.1 Announced

November 8th, 2005

Open Database Conference, Frankfurt, Germany

Peter Eisentraut announces PostgreSQL 8.1

NEWS

08.11.2005 11:02

PostgreSQL 8.1 ist fertig

Seit dem heutigen Dienstag ist die Version 8.1 freigegeben; sie steht auf den [ftp-Servern](#). Die Neuerungen bei der Benutzerverwaltung, den Tabellen. Benutzer können nun bestimmte Rollen zusammenfassen. Dieses Modell bieten auch kor

PostgreSQL community releases version 8.1

9th November 2005

By CBR Staff Writer

The PostgreSQL development community has released version 8.1 of the open source database introducing advanced features and performance improvements for what will be the most advanced open source database management system.

09 Nov 2005

PostgreSQL 8.1正式リリース！

[\[はてなブックマークに追加\]](#) [\[del.icio.usに追加\]](#)

Category: [\[お知らせ\]](#)

PostgreSQL 8.1が正式にリリースされました！

今回のバージョンアップもいろいろと改善されていますので、ぜひお試しください。...

[Top Story](#)

PostgreSQL 8.1 released

Tuesday November 08, 2005 (03:00 PM GMT)

By: [Joe Brockmeier](#)



PRINT



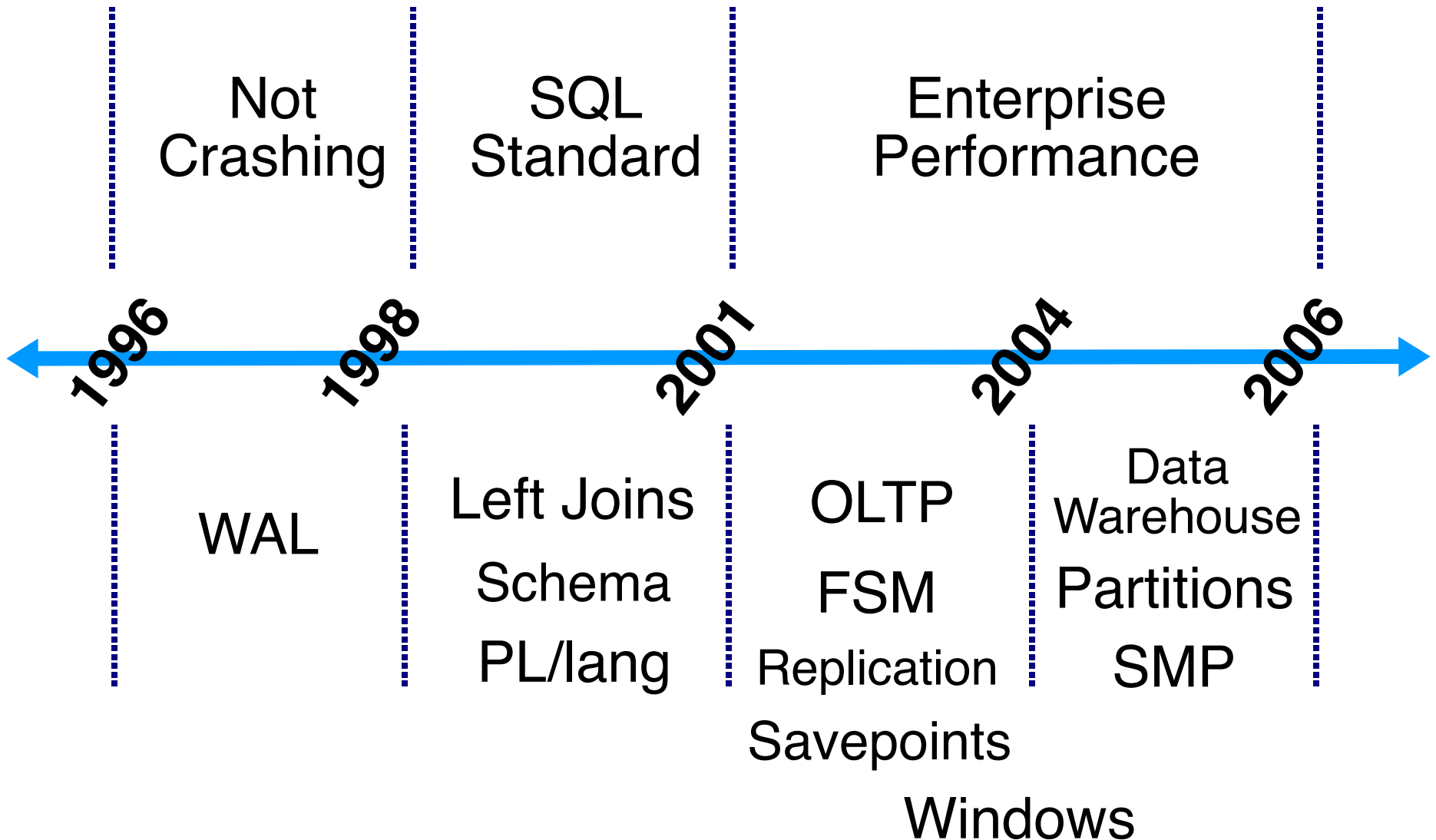
EMAIL



DISCUSS

The PostgreSQL Global Development Group (PGDG) is expected to announce the release of PostgreSQL 8.1 at the [Open Source Database Conference](#) in Frankfurt, Germany. The release is based on a press release from the PostgreSQL Project, including new and improved features and performance improvements, as well as a number of changes in query syntax and PostgreSQL utility commands.

Years And Features



Features of 8.1

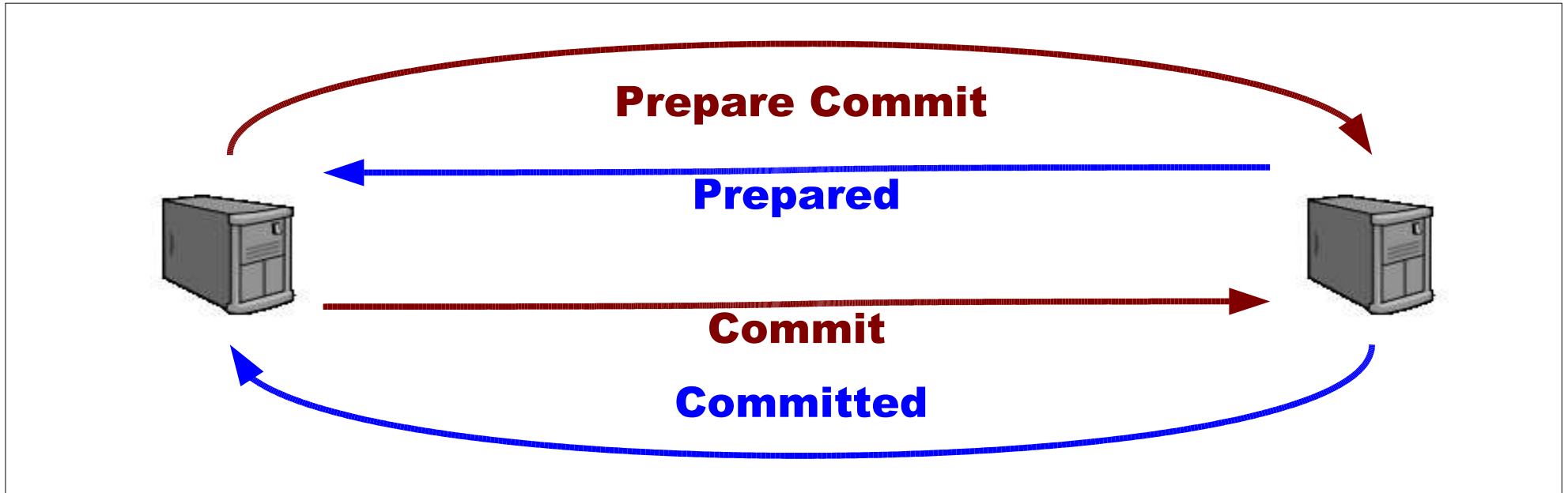
- SQL Features
 - ➔ Two Phase Commit
 - ➔ User ROLES
- Large Database
 - ➔ Index Bitmap Scan
 - ➔ Table Partitioning
- Faster Transactions
 - ➔ SMP Performance
 - ➔ Shared Row Locks
 - ➔ Faster GIST

Other TODOs

Expanded Functions
Integrated AutoVacuum

SQL Features

Two-Phase Commit



```
BEGIN WORK
```

```
INSERT INTO account VALUES  
( 50753, 'debit', 59.95 );
```

```
PREPARE TRANSACTION 'tns50753';
```

```
COMMIT PREPARED 'tns50753';
```

```
BEGIN WORK
```

```
INSERT INTO account VALUES  
( 50753, 'credit', 59.95 );
```

```
PREPARE TRANSACTION 'tns50753';
```

```
COMMIT PREPARED 'tns50753';
```

SQL Features

■ ROLES

How it WAS

- Users belong to Groups
- Groups do not belong to other Groups
- Groups cannot own Objects
- Like Filesystem Permissions
- Not SQL Standard

How it IS NOW

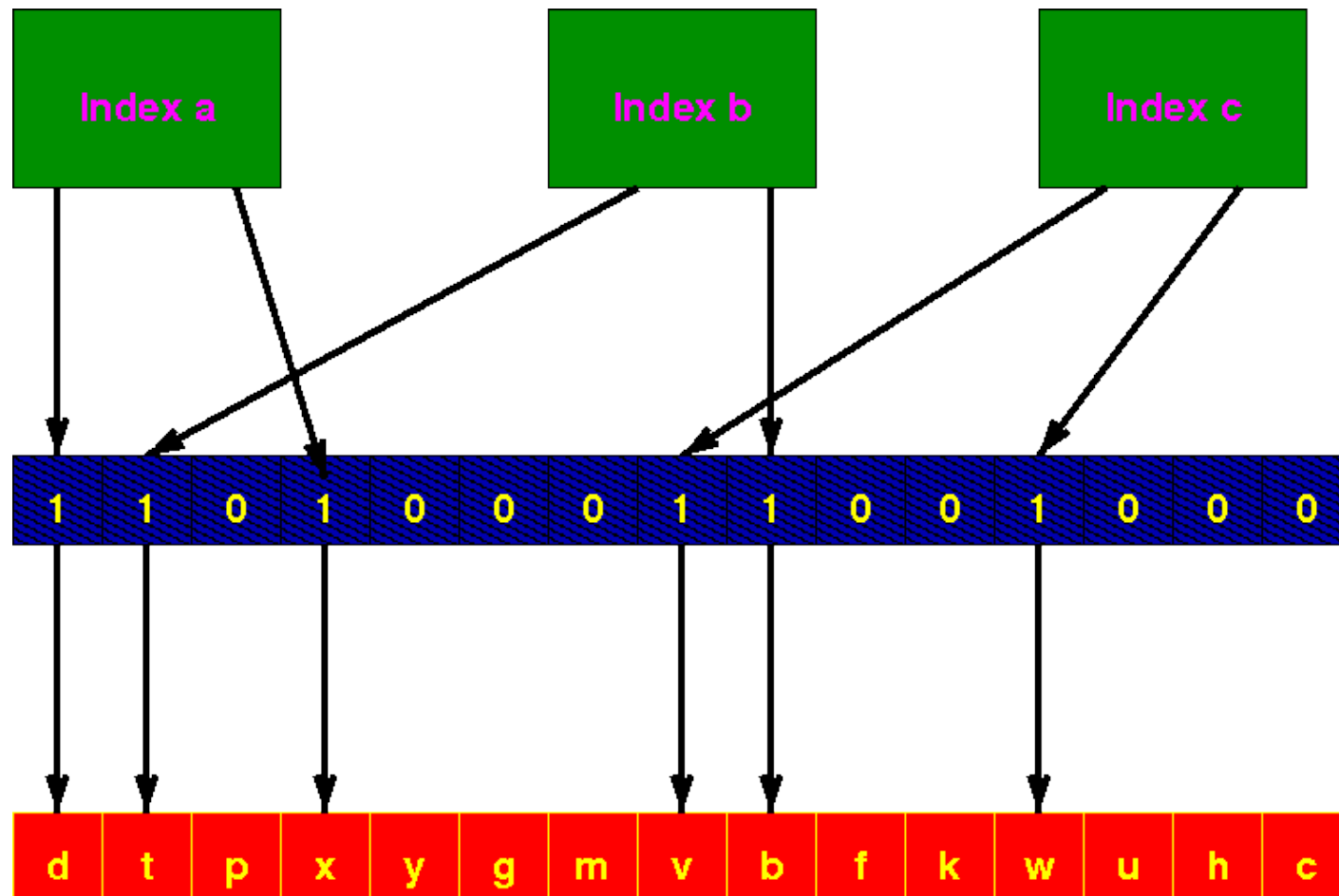
- Users and Groups are Roles
- Roles can belong to other Roles
- Roles own Objects
- SET ROLE and INHERITS
- SQL Standard

Large Databases

Bitmap Scan on Indexes

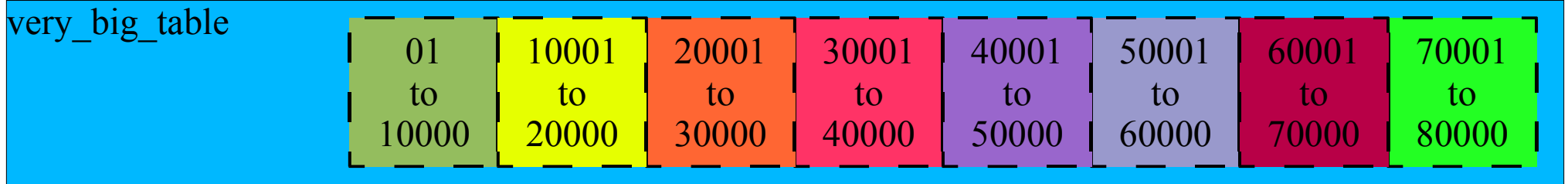
➔ Faster index scans

➔ Merge indexes



Large Databases

■ Table Partitioning



```
CREATE TABLE very_big_master (  
    id SERIAL PRIMARY KEY  
    name TEXT NOT NULL ... );
```

```
CREATE TABLE very_big_60001 INHERITS (very_big_master)
```

```
ALTER TABLE very_big_60001 ADD CONSTRAINT vb_60001_range  
CHECK ( id >= 60001 AND id <= 70000 );
```

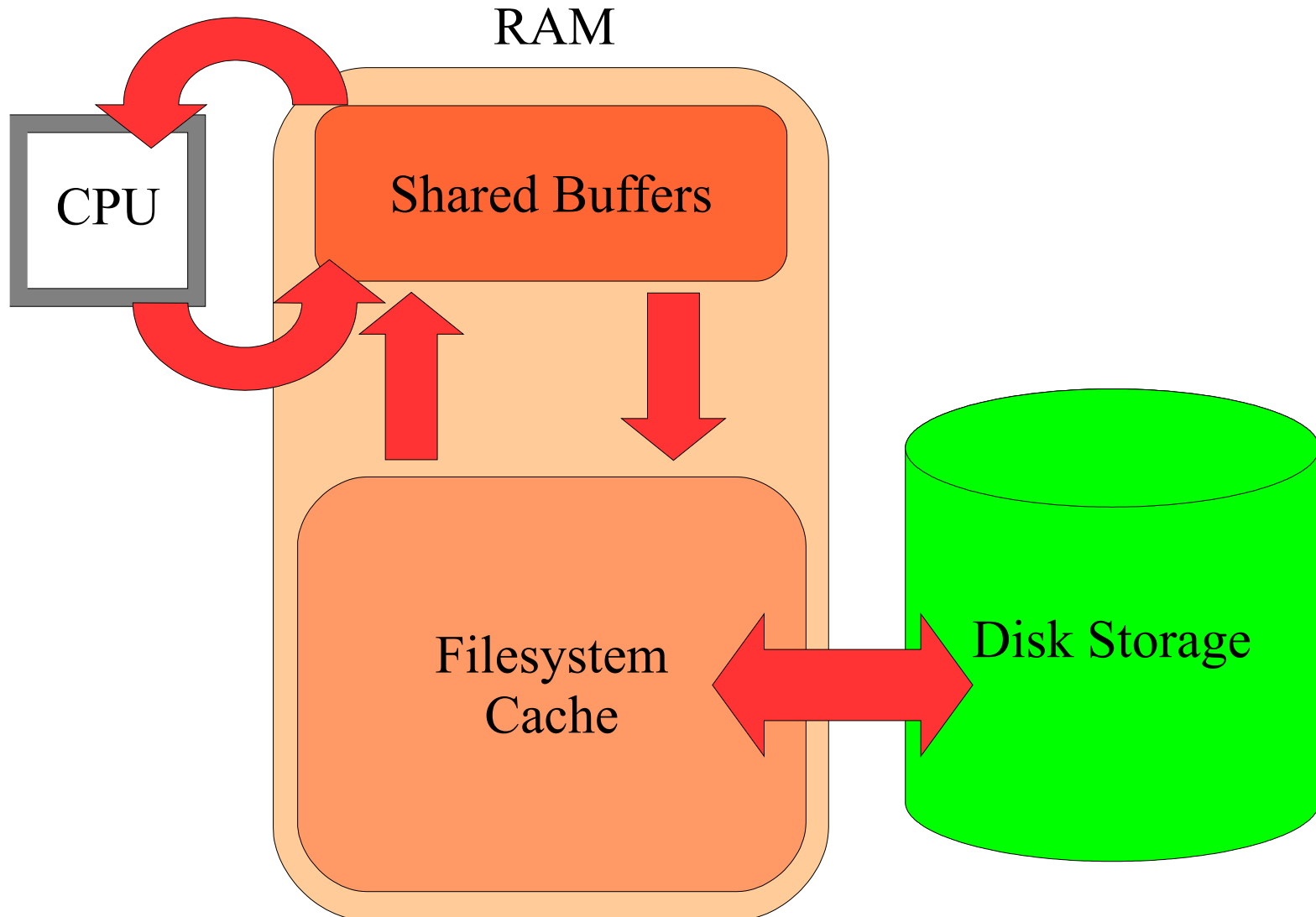
```
CREATE TABLE very_big_70001 INHERITS (very_big_master)
```

```
ALTER TABLE very_big_70001 ADD CONSTRAINT vb_70001_range  
CHECK ( id >= 70001 AND id <= 80000 );
```

```
SELECT * FROM very_big_master WHERE id = 61427;
```


Faster Transactions

- SMP Performance - Shared Buffers



Faster Transactions: SMP

LRU: Least Recently Used: Simplest.
Since Postgres95.

2Q: Two-Queues: adds frequency to recency for
caching evaluation.
PostgreSQL 8.0.2

ARC: Adaptive Resource Cache: Adaptively resizing
version of 2Q.
PostgreSQL 8.0.0

Clock-Sweep: Has a separate register for each
buffer. PostgreSQL 8.1.0

The ARC Patent Mess, or: *Why Software Patents Are Evil*

November 2002: IBM applies for a patent on the memory management algorithm Adaptive Resource Cache.

March 2003: two researchers present ARC as a paper at USENIX. The patent is not mentioned and is not public at this time.

June 2003 – December 2003: Jan Wieck implements ARC for PostgreSQL.

January 2005: PostgreSQL 8.0 is released, including ARC memory management. A developer on the pgsql-hackers mailing list points out IBM's pending patent on ARC.

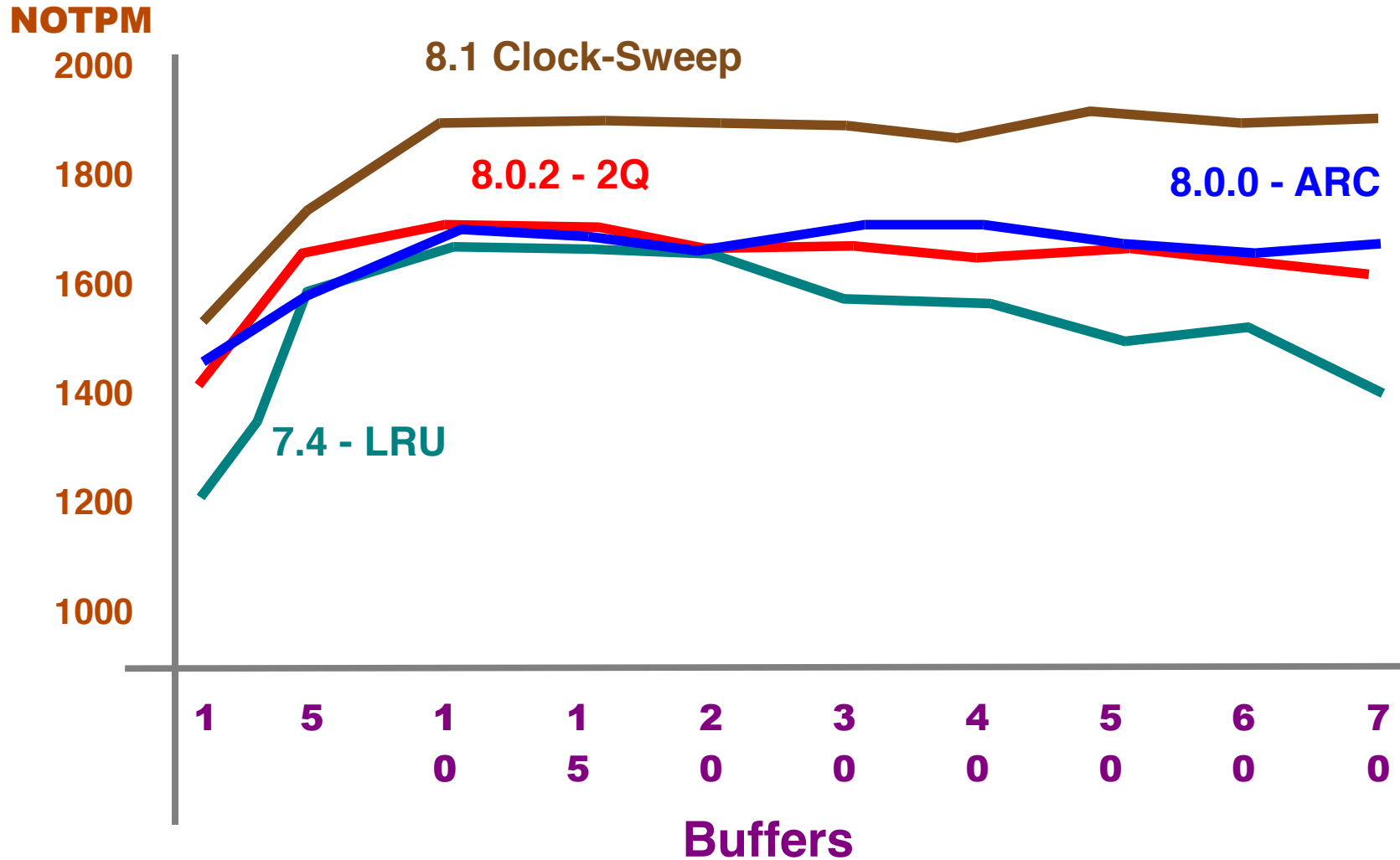
March 2005: After 5 weeks of discussion, Tom Lane hacks a 2Q-based memory manager for PostgreSQL as a temporary measure until 8.1

November 2005: 8.1 includes a new buffer cache manager, Clock-Sweep.

Several weeks of development wasted, and the worst part: ARC did not perform very well!

Faster Transactions: SMP

DBT2: Average New Orders Per Minute (NOTPM)



Faster Transactions

■ Shared Row Locks

- ➔ No more Foreign Key exclusive locks
- ➔ 2-3x more concurrent inserts in high-volume systems
- ➔ locking FOR SHARE

```
SELECT name, type FROM users
WHERE registered_on > ( now() - '1 day' )
FOR SHARE;
```

Faster Transactions

- High-Concurrency GiST -- a team Effort
 - ➔ Our B-Tree Indexes have been high-concurrency since 1991
 - ➔ But GiST indexes required exclusive locks.
 - ➔ Limited the usefulness of PostGIS, TSearch2 indexes
 - ➔ Refractions Inc., 4 other companies raised funds for development by DeltaSoft
 - ➔ Now GiST indexes update quickly in a multi-user environment.



Development

User companies pool on PostgreSQL enhancement

Monday October 24, 2005 (03:00 PM GMT)

By: Jay Lyman



Bigger, Better Functions

■ Full error-handling for PL/pgSQL

...

```
PERFORM catch('modify_contact',vresult);
RETURN vresult;
EXCEPTION
    WHEN RAISE_EXCEPTION THEN
        IF vresult.id = -99 OR vresult.id IS NULL THEN
            INSERT INTO exceptions
                VALUES ( now(), vuser, vid, SQLERRM,
                        'modify_contact_manager' );
        END IF;
        RETURN vresult;
    WHEN OTHERS THEN
        INSERT INTO exceptions
            VALUES ( now(), vuser, vid, SQLERRM,
                    'modify_contact_manager' );
        vresult := row(-99,
                      'A database error occurred.
                       Please consult the support staff.');
```

```
END;
$fs$ language plpgsql security definer;
```

Bigger, Better Functions

- INOUT Functions per SQL, JDBC specification

```
CREATE FUNCTION sum_n_product(  
    IN x int, IN y int,  
    OUT sum int, OUT prod int) AS $$  
BEGIN  
    sum := x + y;  
    prod := x * y;  
END;  
$$ LANGUAGE plpgsql;
```

```
select sum_n_product(6,7);
```

```
sum_n_product  
-----  
(13,42)  
(1 row)
```


Bigger, Better Functions

■ First-class PL/perl

```
CREATE OR REPLACE FUNCTION get_ranks_revised()  
RETURNS SETOF class_ranks  
LANGUAGE plperl AS $$
```

```
    my $query = q{  
  
        SELECT name, class, score  
        FROM   scores  
        ORDER BY class, score desc, name  
  
    };  
  
    my $rank = 0;  
    my $prev_class = "";  
    my $handle = spi_query($query);  
    while (my $row = spi_fetchrow($handle))  
    {  
        $rank = 0 if ($row->{class} ne $prev_class);  
        $prev_class = $row->{class};  
        $rank++;  
        $row->{class_rank} = "$rank";  
        return_next($row);  
    }  
    return undef;
```

```
$$;
```

Integrated AutoVacuum

■ A longstanding request

```
#-----  
# AUTOVACUUM PARAMETERS  
#-----  
  
autovacuum = on                # enable autovacuum subprocess?  
autovacuum_naptime = 60       # time between autovacuum  
                               # runs, in secs  
autovacuum_vacuum_threshold = 300 # min # of tuple updates before  
                               # vacuum  
autovacuum_analyze_threshold = 100 # min # of tuple updates before  
                               # analyze  
autovacuum_vacuum_scale_factor = 0.4 # fraction of rel size before  
                               # vacuum  
autovacuum_analyze_scale_factor = 0.2 # fraction of rel size before  
                               # analyze  
#autovacuum_vacuum_cost_delay = -1 # default vacuum cost delay for  
                               # autovac, -1 means use  
                               # vacuum_cost_delay  
#autovacuum_vacuum_cost_limit = -1 # default vacuum cost limit for  
                               # autovac, -1 means use  
                               # vacuum_cost_limit
```

8.1 Release -- Milestone

- Released *on time*
- Automated testing
- Code analysis
- No bugs for one month after release



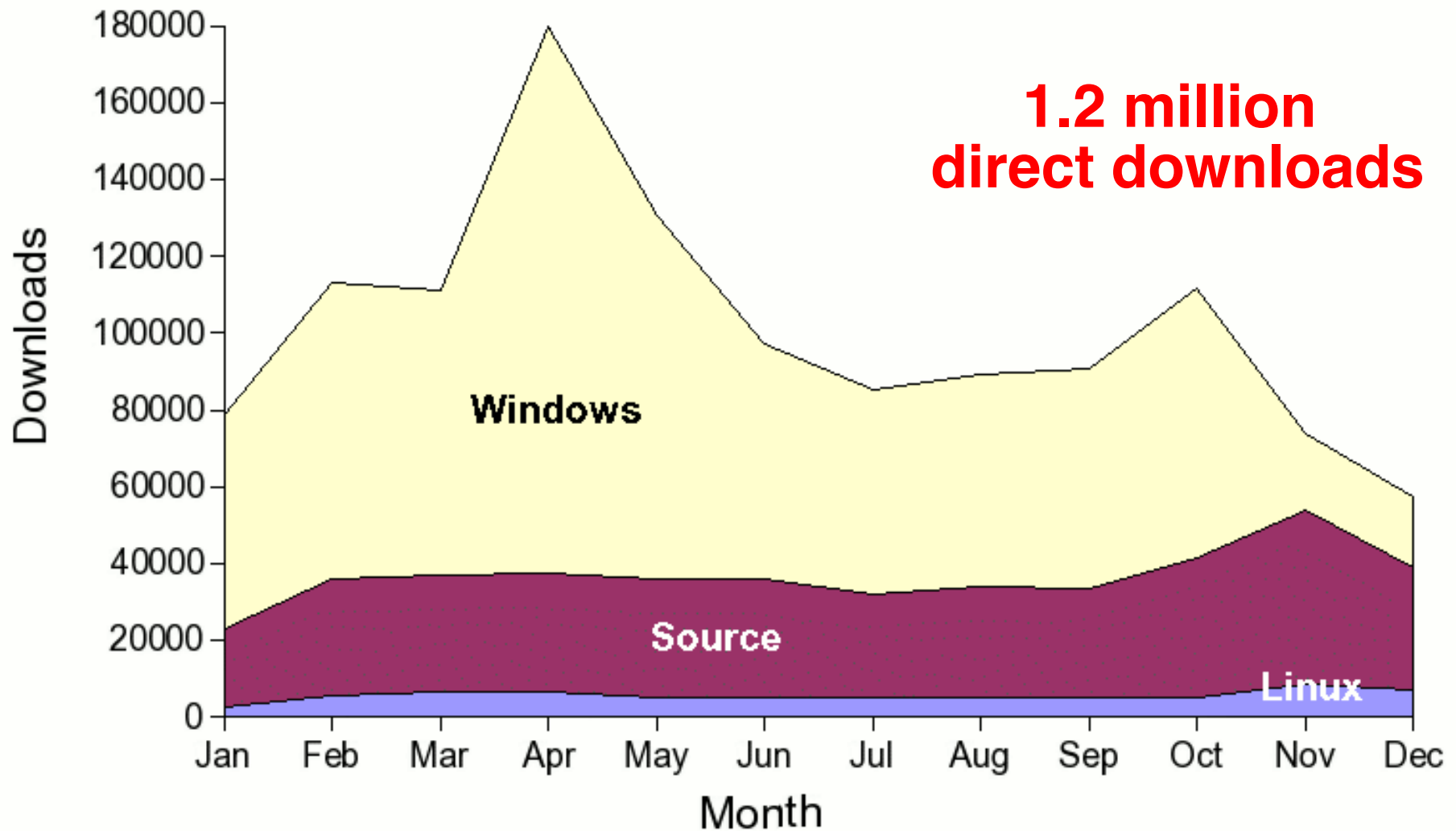
PostgreSQL BuildFarm

Branch: HEAD

Alias	System	Status	Flags
kudu	Solaris 9 Sun WorkShop 6 update 2 C 5.3 x86	00:39:34 ago OK Config	
dove	FreeBSD 6.0-STABLE gcc 3.4.4 amd64	03:59:34 ago OK Config	
snake	Windows Server 2003 SP1 gcc 3.4.2 i686	04:09:35 ago OK Config	[tcl]
spoonbill	OpenBSD 3.6-current gcc 3.3.2 Sparc64	05:19:31 ago ContribCheck Details	
flatworm	FreeBSD 5.4-RELEASE gcc 3.4.2 x86	05:30:34 ago OK Config	
viper	Fedora Core 3 gcc 3.4.4 x86_64	05:59:34 ago OK Config	

More Users

Downloads by Month - 2005



Bigger Community

■ Non-profits

- ➔ JPUG

- ➔ PostgreSQLFr.org

- ➔ The PostgreSQL Fund at SPI

- ➔ Australia?

■ Conferences

- ➔ Hacker's Summit July 8, 2006

- ➔ JPUG

- ➔ Speakers at: OSCON, LWE, FISL, FOSS-IN, OpenDBCon, more ...

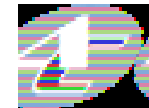
Supporting Companies 2005

PERVASIVE

UNISYS



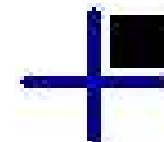
SourceLabs



PostgreSQL, Inc.



FUJITSU

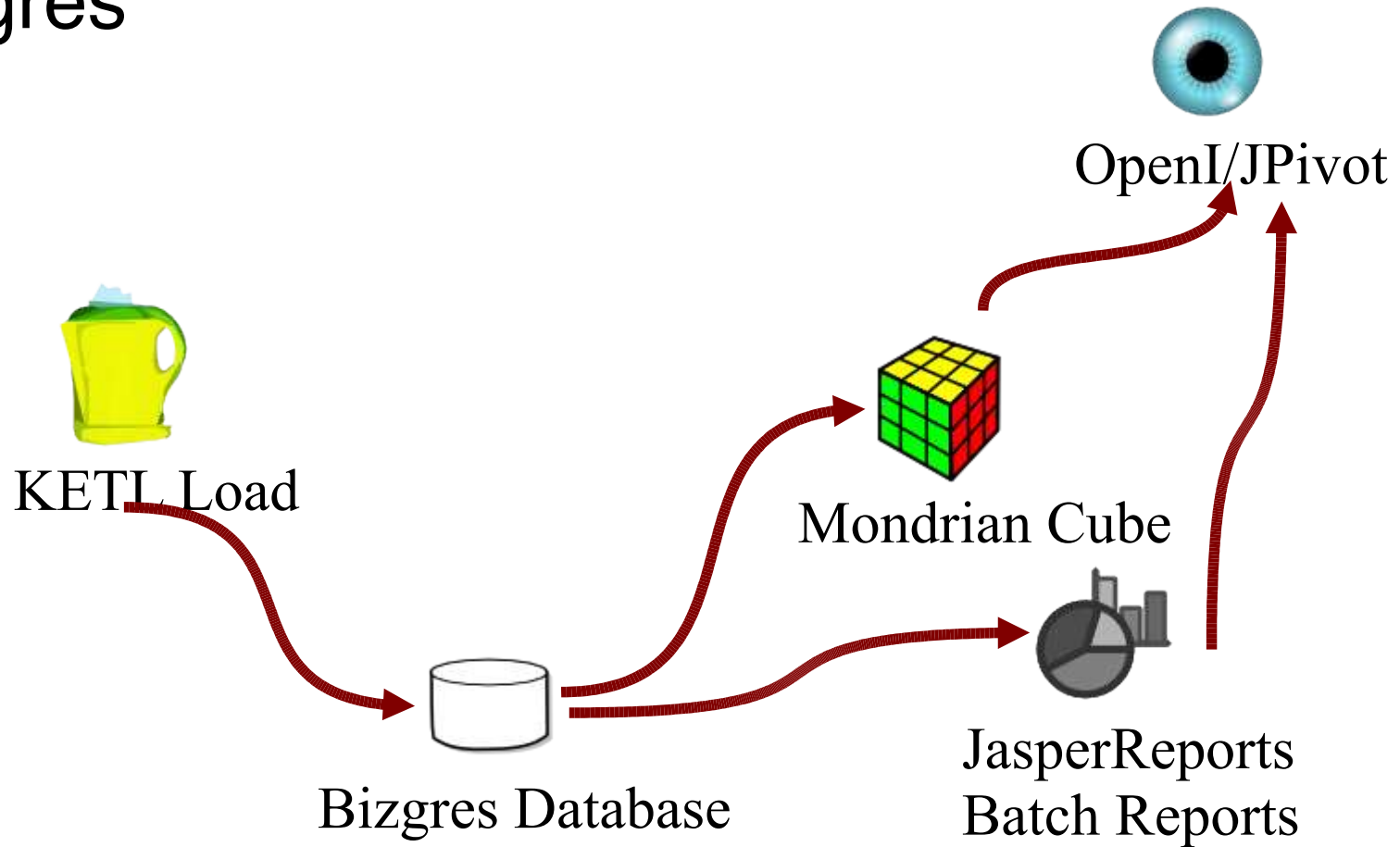


In Development for 8.2

- On-disk Bitmap Indexes
 - ➔ for very large tables with low-cardinality columns
- Faster Sorting
 - ➔ algorithm and memory improvements of 40%
- Outer Join Optimization
 - ➔ up to 200% faster in some queries
- PL/pgSQL Debugger
 - ➔ interactive code testing
- TSearch3
 - ➔ terabyte-scalable TSearch
- PL/Java?

Projects to Watch

■ Bizgres



Projects to Watch

GRASS

The screenshot displays the GRASS GIS 5.1 interface. The main window shows a map of a region with a yellow background and a network of lines. A data table is overlaid on the map, showing the following data:

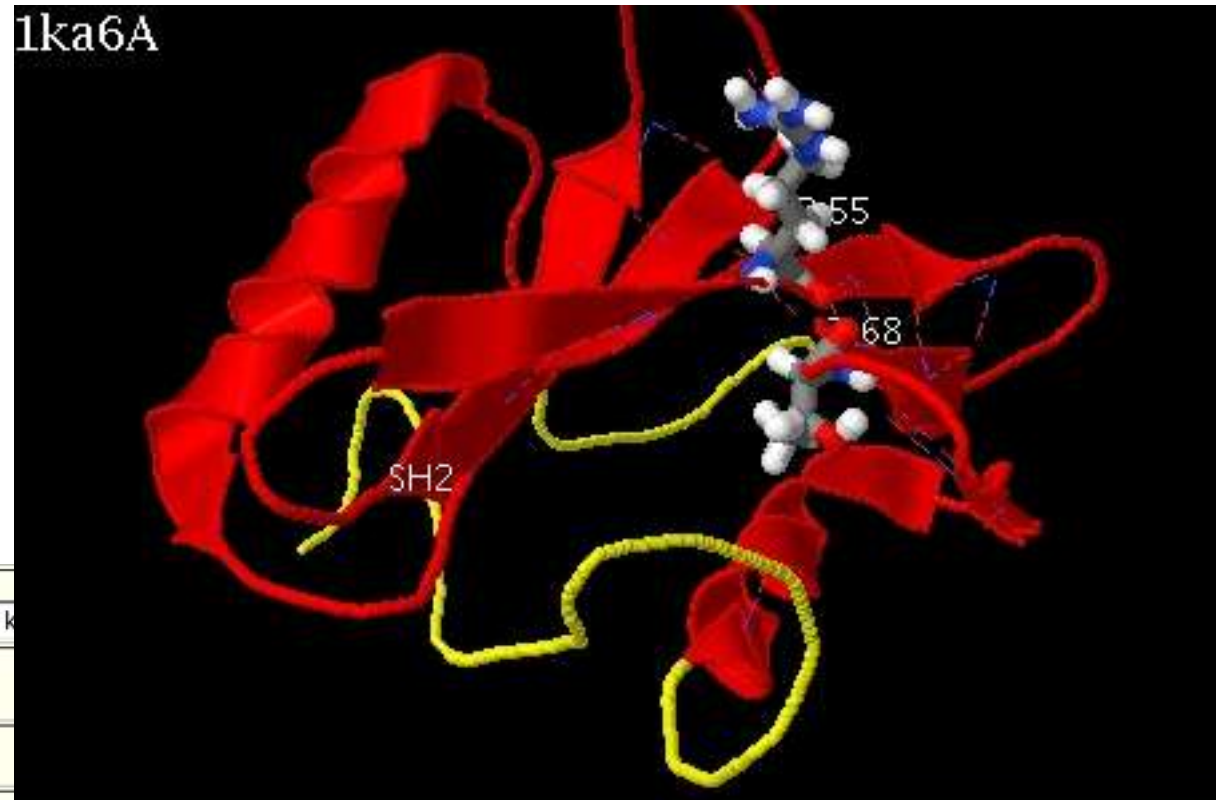
id	cat	li	f_code	nam	file_id
3118	13	13	50-AL000	WRODIEJM	
3116	145	145	50-AL000	WRODZ	
3117	3174	3122	1716-AL000	WRODZ	

The interface also includes a toolbar at the top, a legend on the left, and a status bar at the bottom showing the scale (1:30) and the date (6/7/03 23).



Projects to Watch

- BioPostgres, Unison Protein Database



vroom		TNF k	
Select sequences matching any model in			
TNF superfamily (set 3)			
using these methods:	# models in set	hits	
<input checked="" type="checkbox"/> HMM/Pfam Parameter set: Pfam_fs 14.0 (set 15) with eval <= 1e-10	8	328	12 (63.2%)
<input checked="" type="checkbox"/> PSSM/PSI-BLAST profiles (SBP) Parameter set: PSSM default (set 8) with eval <= 1e-10	93	1724	15 (78.9%)
<input checked="" type="checkbox"/> Prospect2 Parameter set: Prospect2 ssp_pspired default (set 23) with svm >= 12	22	0	0 (0.0%)
NOTE			
Union (hit by ANY of the above)		1856	17 (89.5%)
Intersection (hit by ALL of the above)		0	0 (0.0%)

Resources

- web: www.postgresql.org
- my e-mail: josh@postgresql.org
- associated projects:
 - ➔ pgBuildFarm: www.pgbuildfarm.org
 - ➔ pgFoundry: www.pgfoundry.org
 - ➔ Bizgres: www.bizgres.org
 - ➔ Unison DB: <http://unison-db.org/>
 - ➔ GRASS: <http://www.hpcc.nectec.or.th/grass>
- my site: www.powerpostgresql.com