

PostgreSQL @ OmniTI

- Keith Fiske
- DBA @ OmniTI

<http://www.omniti.com>
<http://www.keithf4.com>

keith@omniti.com
@keithf4

OmniTI, Inc

- Full-stack support for high-traffic websites & applications
 - Millions of users
 - Terabytes of data
 - Etsy, Gilt, Ora.TV, Freelotto
- Surge Conference - <http://omniti.com/surge>
 - Disaster Porn
 - Sept 12-13 in DC National Harbor area
 - Really awesome secret keynote speaker

PostgreSQL

- Free, Open-source RDBMS
- Started in 1986 as a followup project to its predecessor, Ingres
 - <http://www.postgresql.org/about/history/>
- I started with version 8.4. Since then
 - Streaming replication / Hot standby
 - Extensions
 - Foreign Data Wrappers
 - Common Table Expressions (writable)
 - Index-only scans
 - Range Types

OmniPITR

- Write-Ahead Log (WAL)
 - Serial log of all database changes
 - Delays random-disk access to actual data files
 - Crash recovery
 - Less frequent full backups
 - Replay on a base backup for Point-In-Time Recovery

OmniPITR

- `omnipitr-archive`
- `omnipitr-restore`
- `omnipitr-cleanup`
- `omnipitr-backup-master`
- `omnipitr-backup-slave`
- `omnipitr-synch`
- `omnipitr-monitor`

PG Extractor

- pg_dump/pg_restore limitations
- Filter by schema, table, view, function, type, owner
- Dumps each database object to its own file
- Use regex matching
- Integration with SVN & Git

PG Jobmon

- PostgreSQL Extension system
- Autonomous functions
- Log steps of running function
- Monitors logged functions to ensure they complete
- If/when they fail, where and why

PG Jobmon

```
add_job('job name');
```

```
add_step(job_id, 'What this step will do');
```

```
... do some stuff...
```

```
update_step(step_id, 'good_status', 'What this step did successfully');
```

```
add_step(job_id, 'What this next step will do');
```

```
...do some stuff in a loop...
```

```
update_step(step_id, 'good_status', 'update every loop iteration to track progress');
```

```
add_step(job_id, 'One last step');
```

```
... do just a bit more stuff...
```

```
update_step(step_id, 'good_status', 'Job finished ok');
```

```
close_job(job_id);
```

```
EXCEPTION
```

```
WHEN OTHERS THEN
```

```
update_step(step_id, 'bad_status', 'Uh..oh...: '||coalesce(SQLERRM,'wat'));
```

```
fail_job(job_id);
```


PG Jobmon

show_job('my job name', [int]);

```
-[ RECORD 3 ]-----  
job_id  | 10  
owner   | keith  
job_name | PG_JOBMON TEST BAD JOB  
start_time | 2012-09-15 00:55:44.742176-04  
end_time  | 2012-09-15 00:55:44.851514-04  
status    | CRITICAL  
pid       | 5848  
-[ RECORD 4 ]-----  
job_id  | 9  
owner   | keith  
job_name | PG_JOBMON TEST GOOD JOB  
start_time | 2012-09-15 00:55:44.293575-04  
end_time  | 2012-09-15 00:55:44.725483-04  
status    | OK  
pid       | 5848
```

show_job_like('I forgot my job's whole name', [int]);

show_detail(job_id);

show_detail('job_name', [int]);

show_job_status('bad_status', [int]);

show_job_status('job name', 'status', [int]);

show_running([int]);

```
-[ RECORD 1 ]+-----  
job_id  | 9  
step_id | 19  
action   | Test step 1  
start_time | 2012-09-15 00:55:44.501825-04  
end_time  | 2012-09-15 00:55:44.593389-04  
elapsed_time | 0.091564  
status    | OK  
message   | Successful Step 1  
-[ RECORD 2 ]+-----  
job_id  | 9  
step_id | 20  
action   | Test step 2  
start_time | 2012-09-15 00:55:44.643017-04  
end_time  | 2012-09-15 00:55:44.659336-04  
elapsed_time | 0.016319  
status    | OK  
message   | Rows affected: 2  
-[ RECORD 3 ]+-----  
job_id  | 9  
step_id | 21  
action   | Test step 3  
start_time | 2012-09-15 00:55:44.692518-04  
end_time  | 2012-09-15 00:55:44.7087-04  
elapsed_time | 0.016182  
status    | OK  
message   | Successful Step 3
```

PG Jobmon

check_job_status(interval);

- **Make nagios check (command and service configs on my blog)**
- **Shameless plug – <http://circonus.com> (howto on my blog)**

```
SELECT t.alert_text || c.alert_text AS alert_status
FROM jobmon.check_job_status() c
JOIN jobmon.job_status_text t ON c.alert_code = t.alert_code;
```

alert_status

OK(All jobs run successfully)

alert_status

CRITICAL(KEITH.SOME_OTHER_PROCESS: MISSING - Last run at 2012-09-13
07:17:07.86378-04; KEITH.ANOTHER_PROCESS: MISSING - Last run at 2012-09-13
07:16:30.169683-04;)

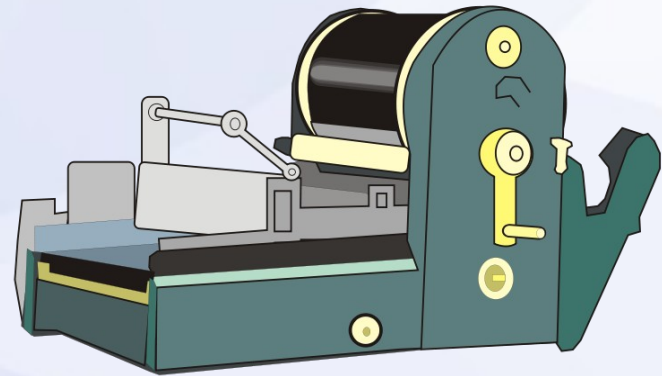
alert_status

WARNING(KEITH.SOME_CRITICAL_PROCESS: RUNNING;)

Mimeo

Per-table Replication Extension

*“The stencil duplicator or mimeograph machine (often abbreviated to mimeo) is a **low-cost** printing press that works by forcing ink through a stencil onto paper...Mimeographs were a common technology in printing **small quantities**, as in office work, classroom materials, and church bulletins.” – Wikipedia*



Mimeo

- Streaming & Log Shipping Replication (omnipitr)
- Per-table replication
 - Snapshot
 - Incremental
 - DML
- Quick replication setup and tear-down
- No superuser required
- Column filter
- Where Condition
- Monitor & Audit Trail w/ PG Jobmon

PG Partition Manager

- What is table partitioning?
- No built in partition management
 - Use combination of table inheritance & triggers
 - Must manually manage this or script it for each partitioned table

PG Partman

- Yet another extension
- Time & Serial partitioning only
- Static & Dynamic trigger function
- Manages creation of new tables, triggers & functions
- Manages old table retention

Links

- <https://github.com/omniti-labs/omnipitr>
- https://github.com/omniti-labs/pg_extractor
- https://github.com/omniti-labs/pg_jobmon
- <https://github.com/omniti-labs/mimeo>
- https://github.com/keithf4/pg_partman