Thames Valley Perl Mongers

Revival Meeting

20th March 2013
Agenda...

- Who are the Perl Mongers?
- Iain C Docherty, Topic: Moose, Beanstalk and Lacuna Expanse
- Break for Grub!
- Alex Burzynski, Topic: Optimising Perl Programs
- Future for TVPMs meetings/social events
- Meeting close / agreed actions - Oliver Gorwits
Iain C Docherty

Moose Beanstalk and Lacuna Expanse
Lacuna Expanse

- Multi-user on-line Space based game
- [http://lacunaexpanse.com](http://lacunaexpanse.com)
- Open source code, Perl, Moose, DBIx::Class
- JSON API
- Javascript User Interface
- External 3rd party scripts
- Games::Lacuna::Client
Events in Lacuna Expanse

- A building completes construction
- A ship is completed in the shipyard
- A building completes its work
- A ship arrives at its destination.
- The UI ‘views’ a planet
- ‘Continuous’ processes (resources)
Event handling

Ship arrives

UI event

Building completes
Problems processing on a UI request.

• Lag.
• Events are not processed in ‘real-time’.
• Hundreds of events processed at the same time.
• Lag.
Solution

- Beanstalk queue
- Add job onto queue, with a delay
- Daemons process jobs taken from queue
- Lacuna::DB::Result::Schedule
- (each Schedule object mirrors a beanstalk job)
package Lacuna::DB::Result::Schedule;

use Moose;
extends 'Lacuna::DB::Result';
use DateTime;

__PACKAGE__->table('schedule');
__PACKAGE__->add_columns(
    queue => {data_type => 'varchar', size => 30, is_nullable => 0},
    job_id => {data_type => 'int', size => 11, is_nullable => 0},
    delivery => {data_type => 'datetime', is_nullable => 0},
    priority => {data_type => 'int', size => 11, is_nullable => 0, default => 1000},
    parent_table => {data_type => 'varchar', size => 30, is_nullable => 0},
    parent_id => {data_type => 'int', size => 11, is_nullable => 0},
    task => {data_type => 'varchar', size => 30, is_nullable => 0},
    args => {data_type => 'medium_blob', is_nullable => 1, serializer_class => 'JSON'},
);

Method modifiers

after 'insert' => sub {
  my $self = shift;
  $self->queue_for_delivery;
  return $self;
};

before 'delete' => sub {
  my $self = shift;
  my $job_id = $self->job_id;
  Lacuna->queue->delete($job_id);
};
sub queue_for_delivery {
  my ($self) = @_;

  my $delay   = $self->delivery->epoch - DateTime->now->epoch;
  $delay     = 0 if $delay < 0;
  my $queue  = Lacuna->queue || 'default';
  my $priority = $self->priority || 1000;
  my $job = $queue->publish($self->queue, {
    id              => $self->id,
    parent_table    => $self->parent_table,
    parent_id       => $self->parent_id,
    task            => $self->task,
    args            => $self->args,
  }, {
    delay           => $delay,
    priority        => $priority,
  });

  $self->job_id($job->id);
  $self->update;
}

package Lacuna::Queue::Job;
use Moose;
use YAML;

has 'job' => (
    is => 'ro',
    isa => 'Beanstalk::Job',
    required => 1,
    handles => [qw(id buried reserved data error stats delete touch peek release bury args tube ttr priority)],
);

sub payload {
    my ($self) = @_;

    my $args = $self->job->args;
    my $class = $args->{parent_table};
    my $id = $args->{parent_id};

    my $thing = Lacuna->db->resultset($class)->find($id);
    return $thing;
}

__PACKAGE__->meta->make_immutable;
1;
Open Source Repository

http://github.com/plainblack/Lacuna-Server-Open