Parallel Image Searching Using PostgreSQL and PgOpenCL

Tim Child
CEO
3DMashUp
Speaker Bio

- Tim Child

- 35 years experience of software development
- Formerly
  - VP Oracle Corporation
  - VP BEA Systems Inc.
  - VP Informix
  - Leader at Illustra, Autodesk, Navteq, Intuit, ...
- 30+ years experience in 3D, CAD, GIS and DBMS
Outline

• Speaker’s Bio
• Outline
• Goals
• Image Applications
• GPU Comparison
• OpenCL GPU/CPU Programming Language
• PgOpenCL
• System Architecture
• Image Data Types
• Image Processing
• Workflow
• Example Image = Operator
• Future Directions
• Q &A
GPU Accelerated Database

Overall Goals

• Develop New Applications
  o Develop new GPU Accelerated Database Applications that are computationally intensive.

• Ease of Use
  o Make use GPU accelerated code easier to use
  o Make GPU accelerated code more mainstream to Information Technology

• Data Scalability
  o Scale GPU application data size

• Enhance existing database internal operations
Image Applications

• E-Commerce
  o Custom catalog images
• Medical
  o X-Ray, CT Scan, MRI
• Earth Sciences, GIS
  o Remote Sensing, Aerial Photography, LIDAR
• Industrial
  o QA, Metrology
• Games, Entertainment
  o VR, AR, Social Media
• Arts
  o Photography
• Security
  o Biometrics (Face, Iris, Finger prints)
• Many others
# GPU Comparison

<table>
<thead>
<tr>
<th>Vendor Architecture</th>
<th>NVidia Fermi</th>
<th>ATI Radeon Evergreen</th>
<th>Intel Nehalem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cores/ ALU</td>
<td>448 Simple</td>
<td>1408 Simple</td>
<td>4 Complex</td>
</tr>
<tr>
<td>Transistors</td>
<td>3.1 B</td>
<td>2.15 B</td>
<td>731 M</td>
</tr>
<tr>
<td>Clock</td>
<td>1.5 G Hz</td>
<td>851 M Hz</td>
<td>3 G Hz</td>
</tr>
<tr>
<td>Peak Float Performance</td>
<td>1500 G FLOP / s</td>
<td>2720 G FLOP / s</td>
<td>96 G FLOP / s</td>
</tr>
<tr>
<td>Peak Double Performance</td>
<td>750 G FLOP / s</td>
<td>544 G FLOP / s</td>
<td>48 G FLOP / s</td>
</tr>
<tr>
<td>Memory Bandwidth</td>
<td>~ 190 G / s</td>
<td>~ 153 G / s</td>
<td>~ 30 G / s</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>250 W</td>
<td>253 W</td>
<td>80 W</td>
</tr>
<tr>
<td>SIMD / Vector Instructions</td>
<td>Many</td>
<td>Many</td>
<td>SSE4+</td>
</tr>
</tbody>
</table>
OpenCL

- OpenCL - Open Compute Language
  - Subset of C 99
  - Open Specification
  - Proposed by Apple
  - Many Companies Collaborated on the Specification
    - Portable, Device Agnostic (Nvidia, AMD, Intel, ARM)
  - Specification maintained by Khronos Group

PgOpenCL

- OpenCL as a PostgreSQL Procedural Language
PgOpenCL

- New PostgreSQL Procedural Language
  - Language Handler
    - Determines N Dimensional Range Work Group Size (Number of Threads)
    - Maps Arguments to Buffers
    - Retains Buffers for Reuse
    - Calls Kernel Function
    - Returns results
  - Language Validator
    - Creates Function – Kernel Binding
      - Parameter Modes, Names, Types, Qualifiers and Attribute
    - Compiles
      - Syntax Checking
      - Generate Program Binary
- New data types
  - cl_double4, cl_double8, ...
  - Image2D,
- System Admin Pseudo-Tables
  - Platform, Device, Run-Time, ...
PgOpenCL System Architecture

PgOpenCL SQL Procedure

GPGPU

PCIe Bus

Disk I/O

Tables

PostgreSQL

SQL Statement

Web Browser

Web Server

App Server

PostgreSQL Client

HTTP

TCP/IP

2011
Image Data Types

2D or 3D

Channel Order

Channel Type

byte, short, half, integer, float
Image Processing

- Geometric Transformation
  - Scale, Rotate, Warp, Interpolation, ...

- Color Transformation
  - Gray Scale, Color Balance, Histograms

- Mensuration
  - Area, Centroid, Perimeter, ..

- Noise Reduction,
  - Noise reduction, de-blurring, sharpening,

- Spatial Filtering
  - Gaussian Filter, Laplace Filter, Sobel Filter, ..

- Transformations
  - DCT, DFT, FFT
PgOpenCL Image2D Type

• New UDT

```
CREATE TYPE opencl.image2d
  (INPUT="opencl.image2d_in", OUTPUT="opencl.image2d_out", DEFAULT="",
   INTERNALLENGTH=-1, ALIGNMENT=int4, STORAGE=EXTENDED);
```

ALTER TYPE opencl.image2d OWNER TO postgres;
COMMENT ON TYPE opencl.image2d IS 'PgOpenCL 2D Image type';

• Input Format

```
insert into images (image) values
('CL_ARGB CL_UNSIGNED_INT8 1 1 4 //!!!w==');
```

• Casts from - to bytea type
Basic Example

Image Operator =
Example C Code

/** Now check each Pixel **/
for ( y = 0;  y < height; y++)
{
   for( x = 0;x  x < width; x++)
   {
      if ( pixelA[ x, y ] != pixelB[ x, y ]
      {
         PG_RETURN_BOOL(false);
      }
   }
}
Example OpenCL Code

```c
kernel void imageequalskernel(__read_only image2d_t inputImageA, __read_only image2d_t inputImageB, __global int * result)
{
    int2 coord = (int2)(get_global_id(0), get_global_id(1));
    Uint4 pixelA;
    Uint4 pixelB;

    /** If the coordinates are within range **/
    if (coord.x < (get_global_size(0) - 1) && coord.y < get_global_size(1) - 1)
    {
        /** Read and compare each pixel channel, if they're not equal increment the result counter **/

        pixelA = read_imageui( inputImageA, imageSampler, (int2)(coord.x, coord.y) );
        pixelB = read_imageui( inputImageB, imageSampler, (int2)(coord.x, coord.y) );

        /** compare each RGBA component **/

        if ( pixelA.x != pixelB.x || pixelA.y != pixelB.y || pixelA.z != pixelB.z || pixelA.w != pixelB.w )
            atomic_inc(&result[0]);
    }
}
```
create type imagetype as enum ('jpg');

create temporary table images
(
    id    serial,
    name  text,
    type  imagetype,
    image image2d
);

insert into images (name, type, image) values ('test', 'jpg', 'CL_ARGB CL_UNSIGNED_INT8 1 1 4 //////w==');

select imageequalskernel(image, image) = '{0}'::integer[1];
Wishful Thinking

Possible Nest Step Future Directions

- Develop Image Processing Applications
  - Rounds out function
  - Provides an example
- Server Side Compression/Decompressions
  - Jpg
  - Tiff
- Client Side Support
  - C Library
  - Java Library
- New Image Types
  - Image3D
- New Formats
  - Tiff
  - Geo-Tiff
  - DICOM
- Enhance Language Manager
  - Async
  - Image Processing Pipelines
Summary

• PostgreSQL – Centralized Data and Image Storage Platform

• Minimizing Data Transfer - Data Remains on Server

• Fast - Runs at GPU Speed

• Integrated Data and Image Queries

• Standards – OpenCL, SQL
Q&A

- **PgOpenCL**
  - Twitter @3DMashUp

- **OpenCL**
  - [www.khronos.org/opencl/](http://www.khronos.org/opencl/)