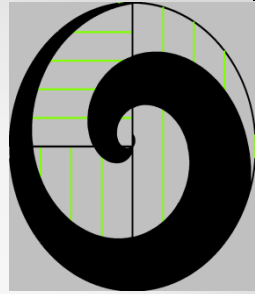


Parallel Image Searching Using PostgreSQL and PgOpenCL

Tim Child
CEO
3DMashUp



Speaker Bio



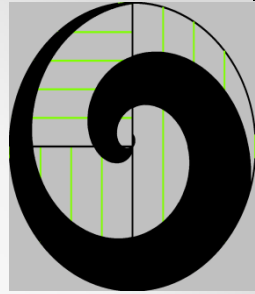
- **Tim Child**

- 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



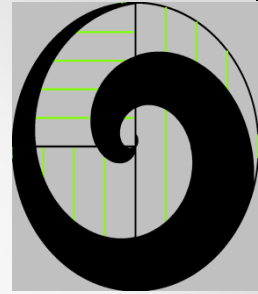
- Speaker's Bio
- Outline
- Goals
- Image Applications
- GPU Comparison
- OpenCL GPU/CPU Programming Language
- PgOpenCL
- System Architecture
- Image Data Types
- Image Processing
- Workflow
- Example Image = Operator
- Future Directions
- Q &A





GPU Accelerated Database

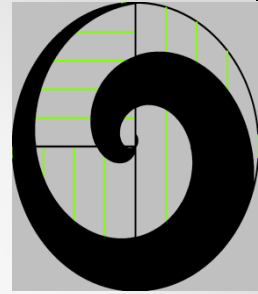
Overall Goals



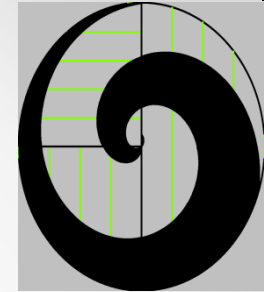
- **Develop New Applications**
 - Develop new GPU Accelerated Database Applications that are computationally intensive.
- **Ease of Use**
 - Make use GPU accelerated code easier to use
 - Make GPU accelerated code more mainstream to Information Technology
- **Data Scalability**
 - Scale GPU application data size
- **Enhance existing database internal operations**



Image Applications



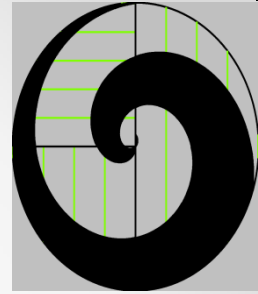
- E-Commerce
 - Custom catalog images
- Medical
 - X-Ray, CT Scan, MRI
- Earth Sciences, GIS
 - Remote Sensing, Aerial Photography, LIDAR
- Industrial
 - QA, Metrology
- Games, Entertainment
 - VR, AR, Social Media
- Arts
 - Photography
- Security
 - Biometrics (Face, Iris, Finger prints)
- Many others



GPU Comparison

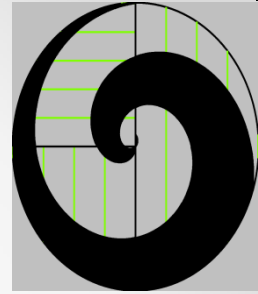


Vendor Architecture	NVidia Fermi	ATI Radeon Evergreen	Intel Nehalem
Cores/ ALU	448 Simple	1408 Simple	4 Complex
Transistors	3.1 B	2.15 B	731 M
Clock	1.5 G Hz	851 M Hz	3 G Hz
Peak Float Performance	1500 G FLOP / s	2720 G FLOP / s	96 G FLOP / s
Peak Double Performance	750 G FLOP / s	544 G FLOP / s	48 G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



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

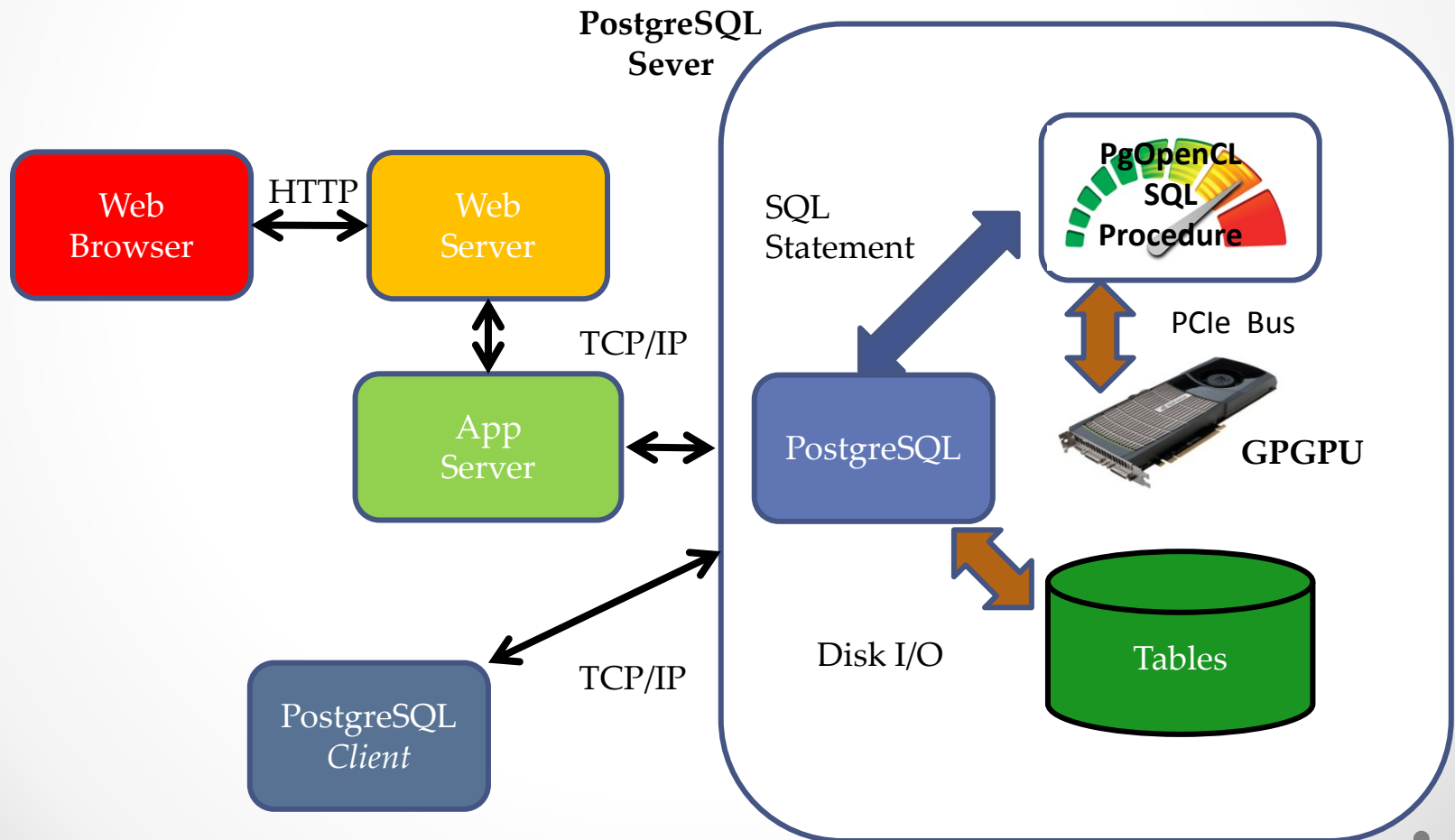
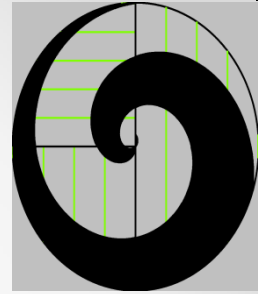


PgOpenCL

- 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



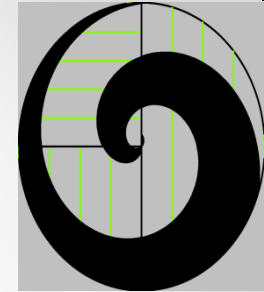
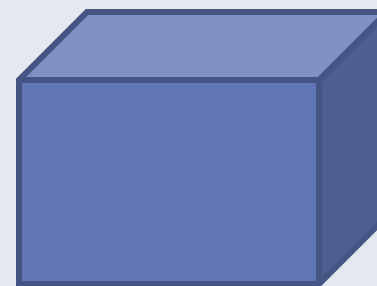


Image Data Types

2D or 3D



Channel Order



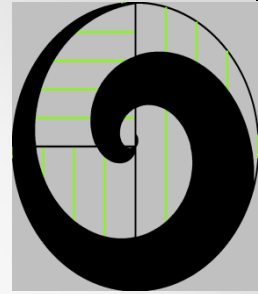
Channel Type



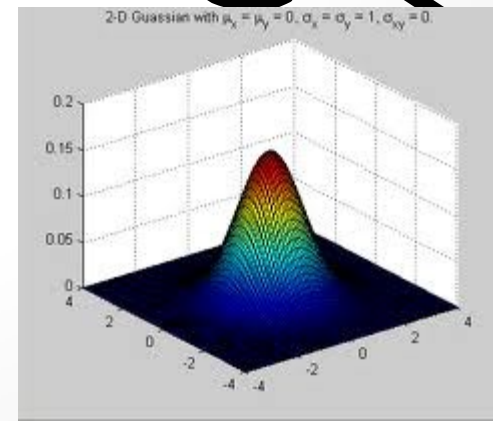
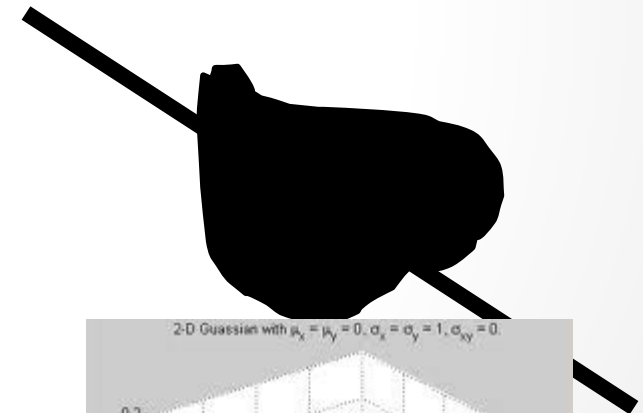
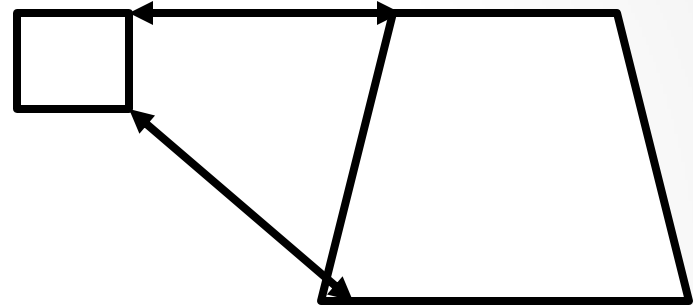
byte, short, half, integer, float

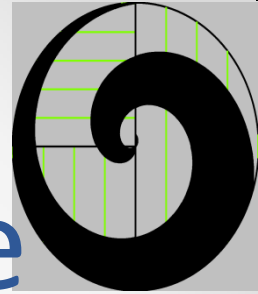


Image Processing



- 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

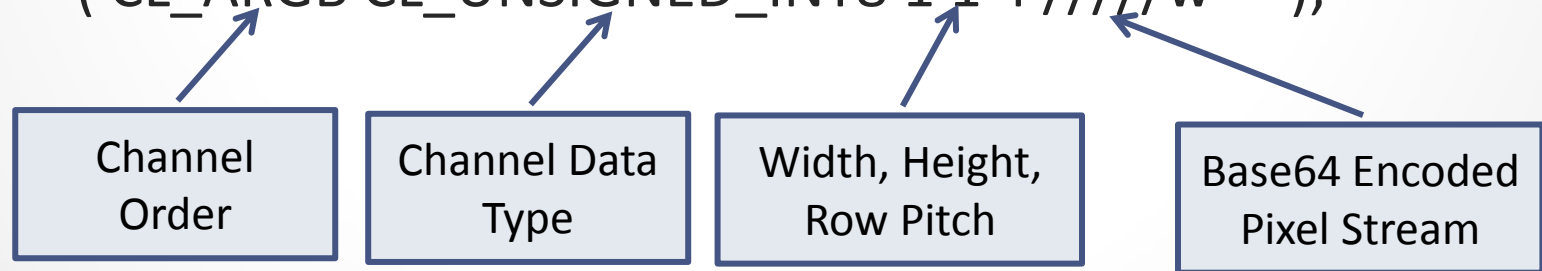
- New UDT

```
CREATE TYPE openc1.image2d
  (INPUT="openc1.image2d_in", OUTPUT="openc1.image2d_out", DEFAULT="",
   INTERNALLENGTH=-1, ALIGNMENT=int4, STORAGE=EXTENDED);
ALTER TYPE openc1.image2d OWNER TO postgres;
COMMENT ON TYPE openc1.image2d IS 'PgOpenCL 2D Image type';
```

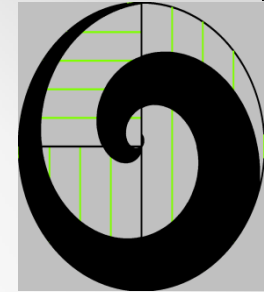
- Input Format

insert into images (image) values

```
('CL_ARGB CL_UNSIGNED_INT8 1 1 4 //w==');
```



- Casts from - to bytea type

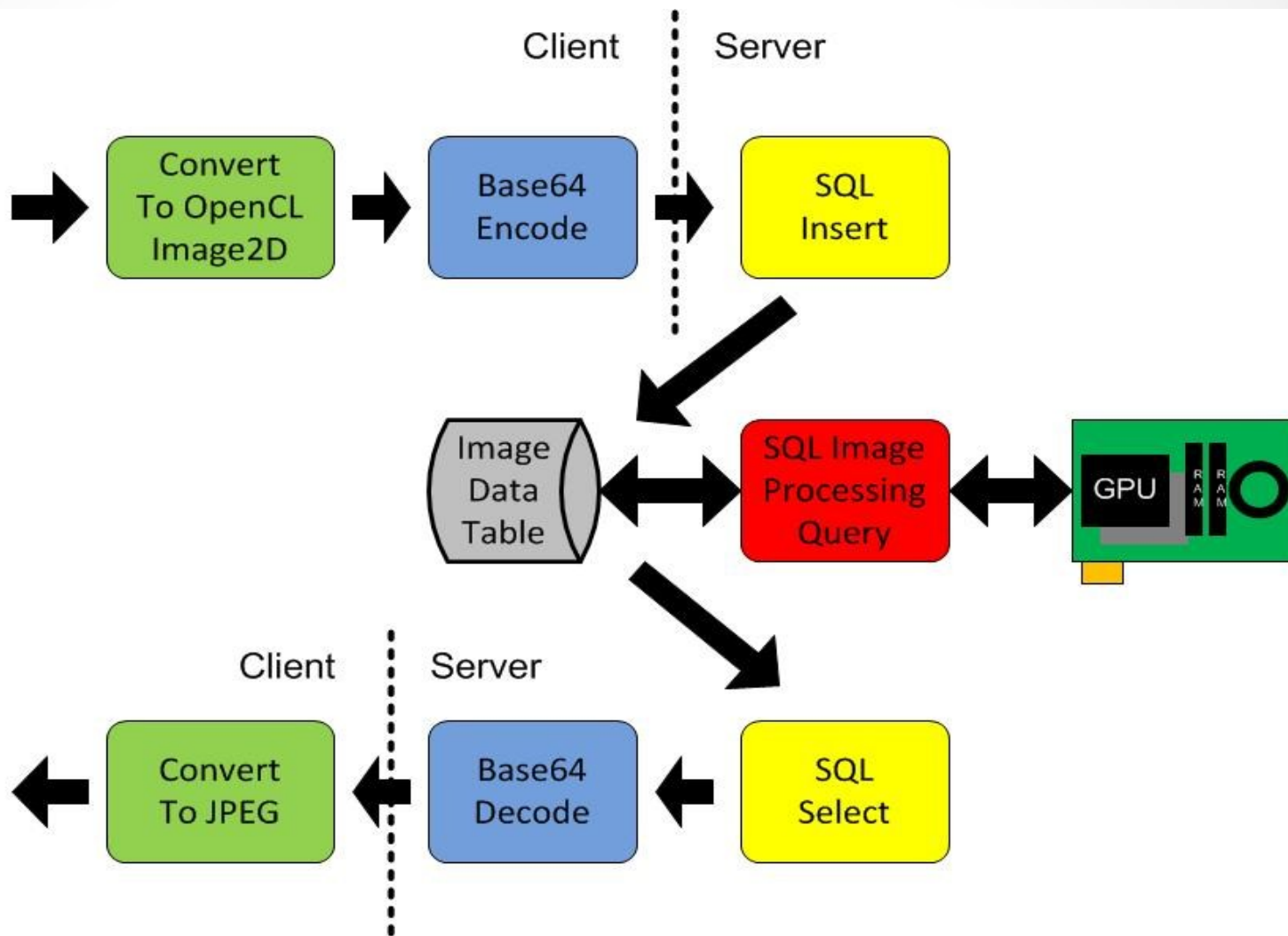


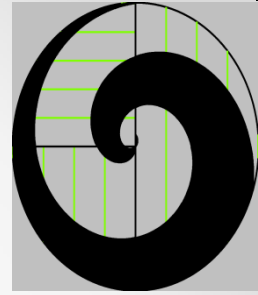
Workflow



JPEG Image

Processed Image



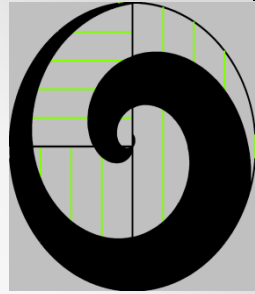


Basic Example

Image Operator =



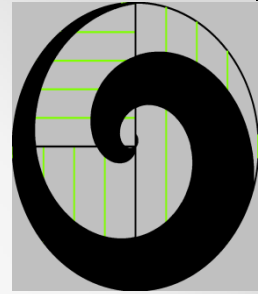
Example C Code



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



Example OpenCL Code



```
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;

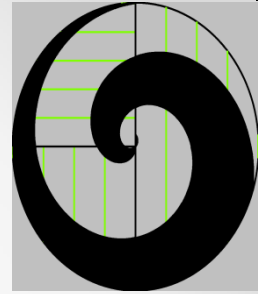
    /** If the coordinates are within range **/

    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]);
    }
}
```

SQL Query

```
create type imagetype as enum ( 'jpg' );

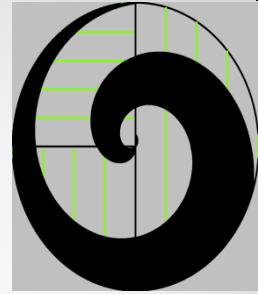
create temporary table images
(
    id        serial,
    name      text,
    type      imagetype,
    image     image2d
);

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];
```

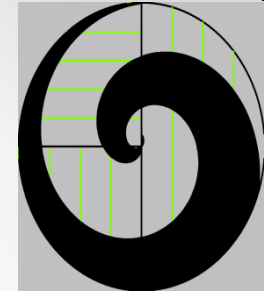


Wishful Thinking



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



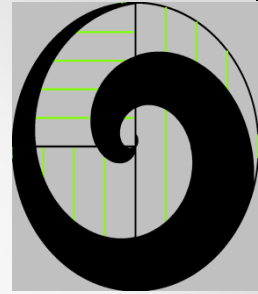
Summary

- 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
 - Blog www.scribd.com/3dmashup
- **OpenCL**
- www.khronos.org/openccl/
- www.amd.com/us/products/technologies/stream-technology/openccl/
- <http://software.intel.com/en-us/articles/intel-openccl-sdk>
- http://www.nvidia.com/object/cuda_openccl_new.html