


# What is GPGPU

- **General Purpose Computation on GPUs**
  - The GPU is a graphics-specific processor but very useful for general purpose computation
    - Parallel processor
    - Flexible programming

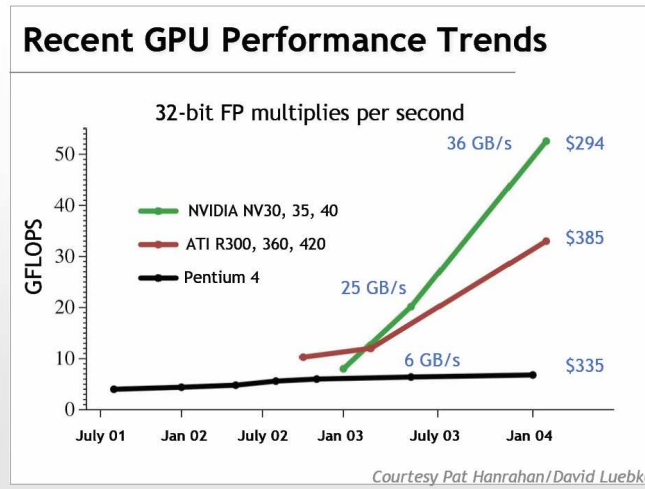
# DirectX 9.0 / Shader Model 3.0

**Complete Native Shader Model 3.0 Support** 

	DirectX 9.0	Shader Model 3.0
<b>Vertex Shader Model</b>	<b>2.0</b>	<b>3.0</b>
Vertex Shader Instructions	256	2 <sup>16</sup> (65,535)
Displacement Mapping	-	✓
Vertex Texture Fetch	-	✓
Geometry Instancing	-	✓
Dynamic Flow Control	-	✓
<b>Pixel Shader Model</b>	<b>2.0a</b>	<b>3.0</b>
Required Shader Precision	fp24	fp32
Pixel Shader Instructions	512	2 <sup>16</sup> (65,535)
Subroutines	-	✓
Loops & Branches	-	✓
Dynamic Flow Control	-	✓

Source: Randima Fernando, NVidia.  
[http://eg04.inrialpes.fr/Programme/IndustrialSeminar/PPT/Trends\\_in\\_GPU\\_Evolution.pdf](http://eg04.inrialpes.fr/Programme/IndustrialSeminar/PPT/Trends_in_GPU_Evolution.pdf)

## Growth



## Computational Power

- GPUs are fast...
  - 3 GHz Pentium 4 *theoretical*: 6 GFLOPS
  - GeForce FX 5900 *observed\**: 20 GFLOPS
  - GeForce 6800 Ultra *observed\**: >50 GFLOPS

\*Observed on a synthetic benchmark: A long pixel shader with nothing but MUL instructions

## Performance growth

- CPU
  - Annual growth  $\sim 1.5\times \rightarrow$  decade growth  $\sim 60\times$
  - Moore's law
- GPU
  - Annual growth  $> 2.0\times \rightarrow$  decade growth  $> 1000\times$
  - Much faster than Moore's law

## Motivation

- GPUs are
  - Inexpensive
  - Dedicated to graphical operations
  - **Uses transistors for logic and not cache**

# Looking forward



Source: Randima Fernando, NVidia.  
[http://eg04.inrialpes.fr/Programme/IndustrialSeminar/PPT/Trends\\_in\\_GPU\\_Evolution.pdf](http://eg04.inrialpes.fr/Programme/IndustrialSeminar/PPT/Trends_in_GPU_Evolution.pdf)

# Looking forward

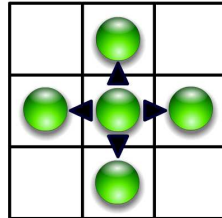
**Performance 1994-2014**

	1994	2004	2014
CPU Frequency (GHz)	.1	3.2	100
Memory Frequency (GHz)	.03	1.2	44
Bus Bandwidth (GB/sec)	.1	4	160
Hard Disk Size (GB)	.5	200	30 TB
Pixel Fill Rate (MPixels/sec)	.40	3300	270 GP
Vertex Rate (MVerts/sec)	.5	356	127 GV
Graphics Flops (GFlops/sec)	.001	40	10 TF
Graphics Bandwidth (GB/sec)	.3	30	3 TB
Frame Buffer Size (MB)	2	256	32 GB

Source: Randima Fernando, NVidia.  
[http://eg04.inrialpes.fr/Programme/IndustrialSeminar/PPT/Trends\\_in\\_GPU\\_Evolution.pdf](http://eg04.inrialpes.fr/Programme/IndustrialSeminar/PPT/Trends_in_GPU_Evolution.pdf)

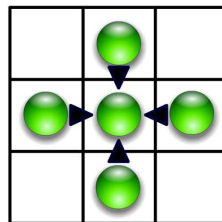
## Vertex programs

- Fully programmable (SIMD / MIMD)
- Processes 4-vectors (RGBA / XYZW)
- Capable of scatter
  - Can change the location of current vertex
  - Cannot read info from other vertices
- Latest GPUs (NV40):
  - Vertex Texture Fetch (gather)
  - Random access memory for vertices



M. Harris

Scatter

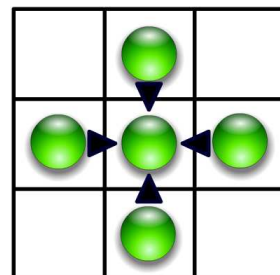


M. Harris

Gather

## Fragment programs

- Fully programmable (SIMD / MIMD)
- Processes 4-vectors (RGBA / XYZW)
- Random access memory read (textures)
- Capable of gather but not scatter
  - RAM read (texture), but no RAM write
  - Output address fixed to a specific pixel
- Typically more useful than vertex processor
  - More fragment pipelines than vertex pipelines
  - Direct output (fragment processor is at end of pipeline)



M. Harris

Gather