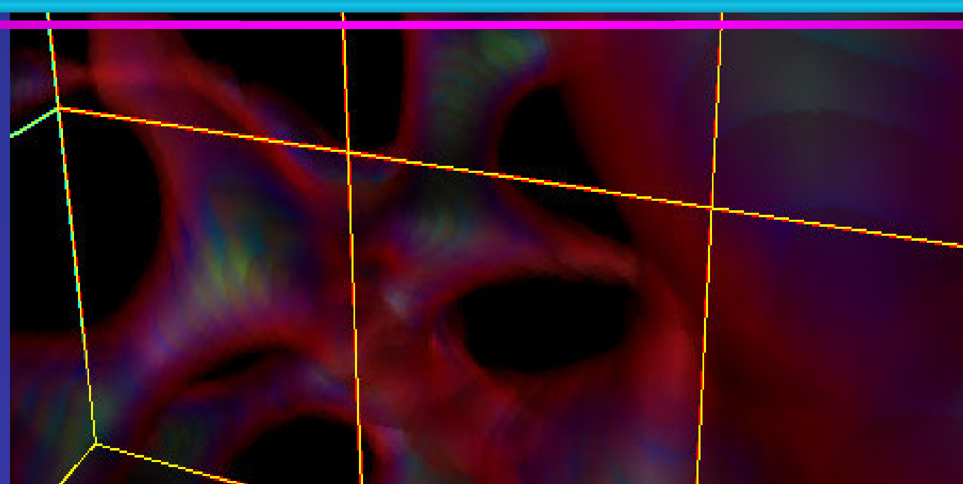
PVG 2003

# A Multi-Layered Image Cache for Scientific Visualization

Eric Lamar and Valerio Pascucci

*Center for Applied Scientific Computing*
*Lawrence Livermore National Laboratory*



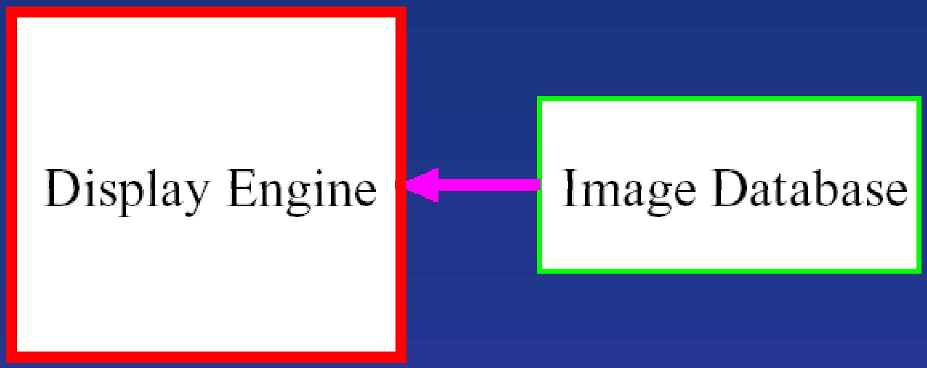
This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48. UCRL-PRES-200743

# Visual data exploration requires interactive visualization tools.

- The user is less frustrated by transient artifacts then by slow refresh rates.
- “Interactive display” is not synonym with “interactive rendering.”
- One obstacle to scalable visualization is the need to render everything within a single frame.
- A possible solution is to decouple the rendering from the display.

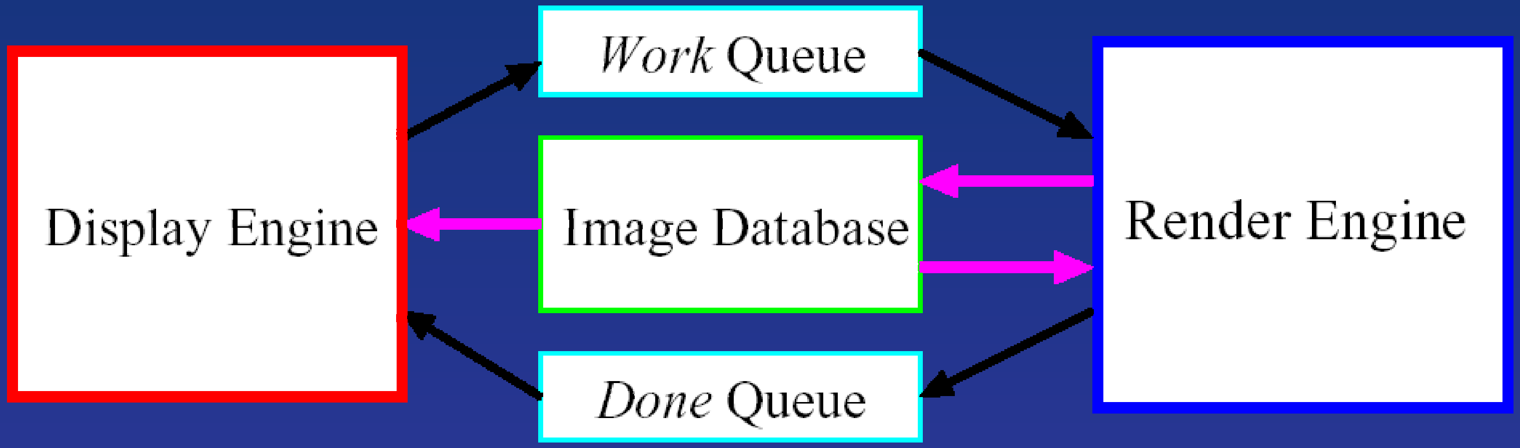
# We develop a hybrid approach to generic asynchronous rendering.

We use a light weighted display engine that reads images retrieved from a small local database.



# We develop a hybrid approach to generic asynchronous rendering.

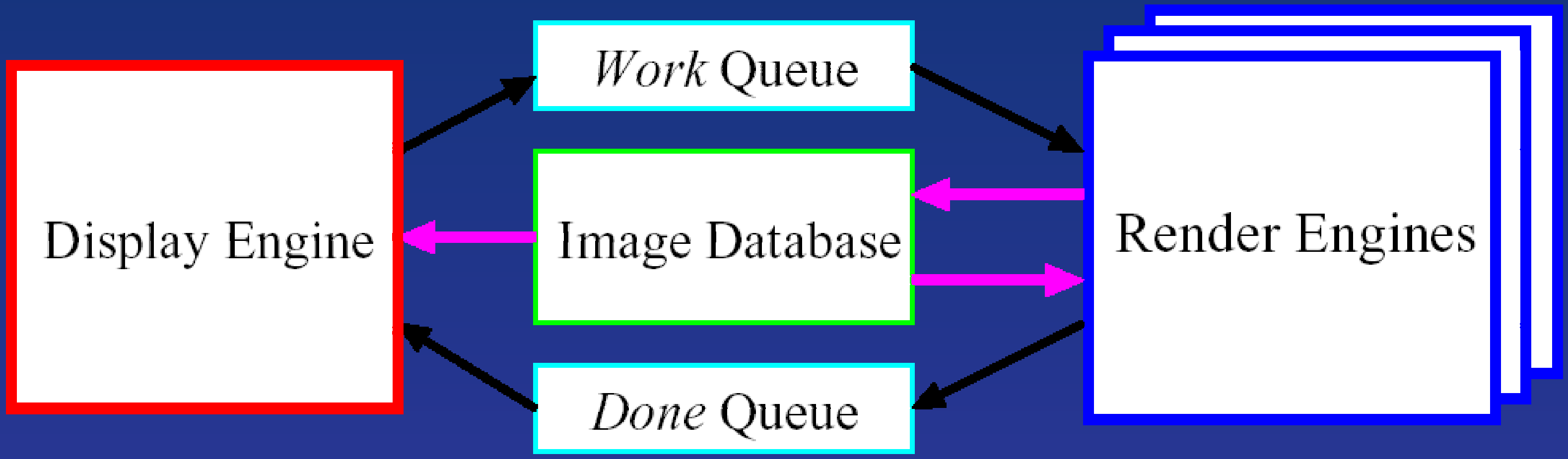
When an image is not available
a request is posted to the “work queue.”




When a request is satisfied
the image database is updated and
a notification is posted to the “done queue.”

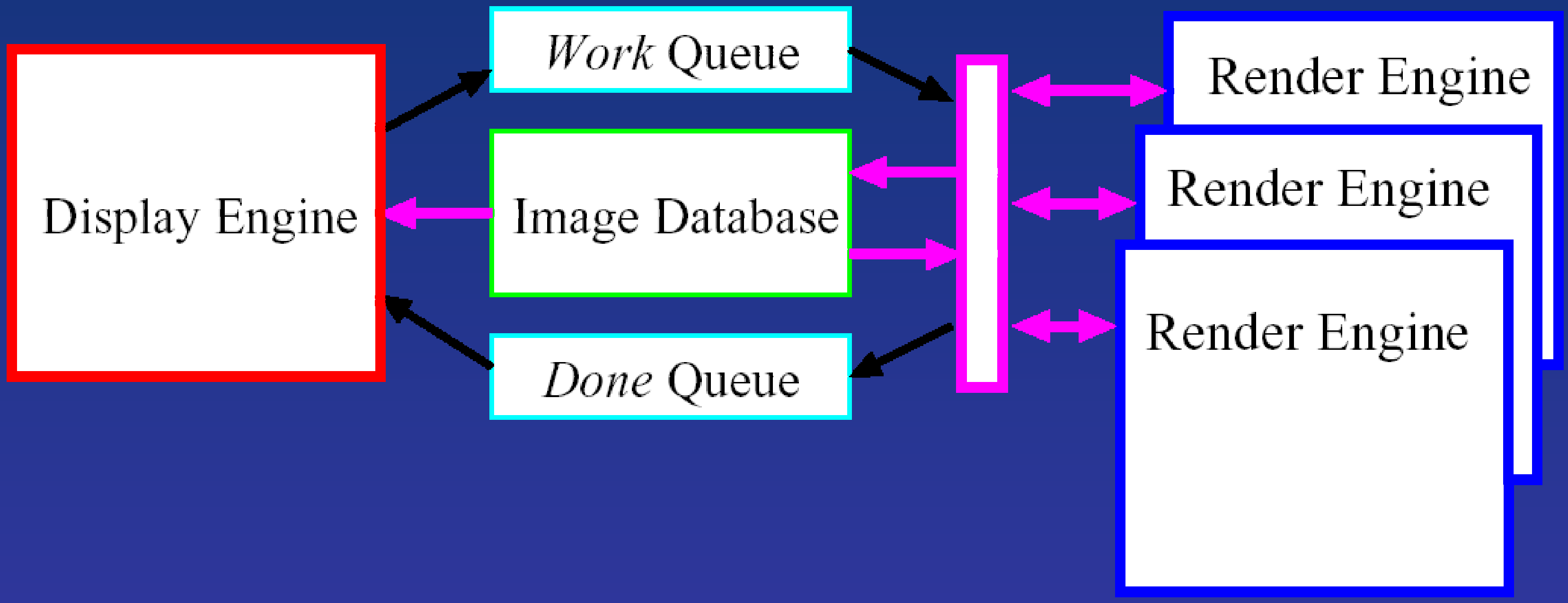
# We develop a hybrid approach to generic asynchronous rendering.

**Multiple rendering engines can access the queues at the same time to achieve parallelism.**



# We develop a hybrid approach to generic asynchronous rendering.

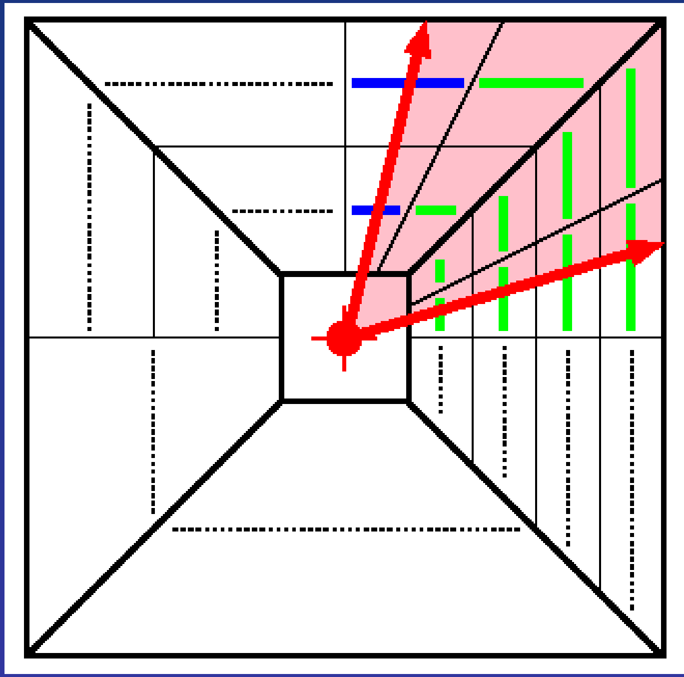
The queues can be accessed remotely to tackle distributed rendering resources.

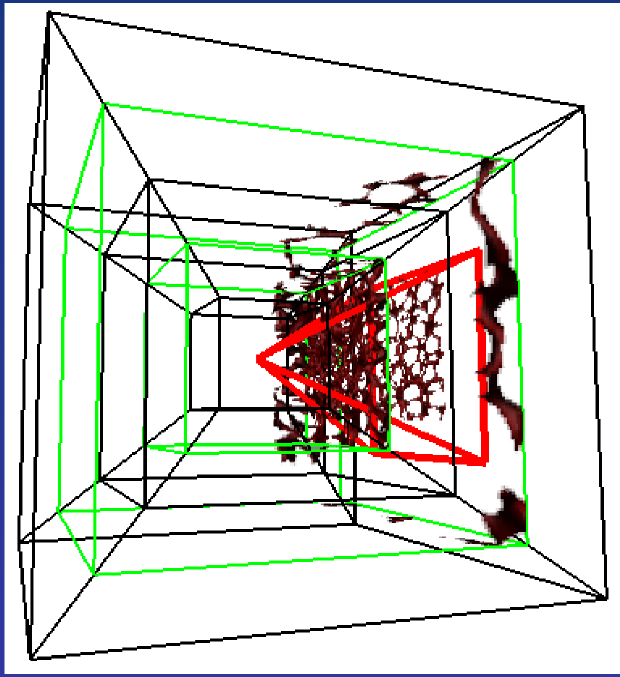


# We develop a view-centric decomposition of the scene.

**A standard cube is anchored around the view point.**

**The view frustum intersects some of the cube faces.**



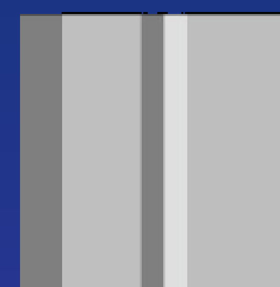


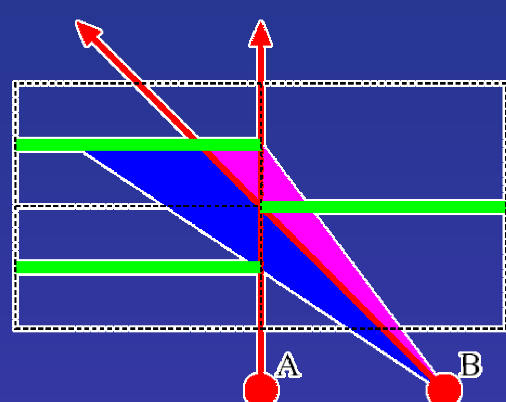
Valerio Pascucci

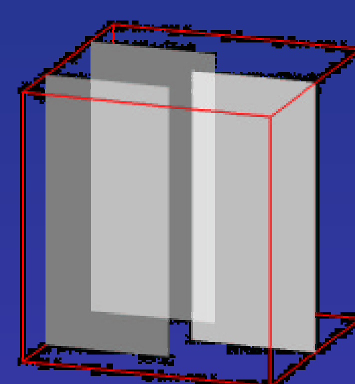
# Transient artifacts appear after a translation of the point of view.

**Rotation and zoom do not invalidate the current images.**

**After a translation the entire image database is invalid but the old images are used until replaced.**

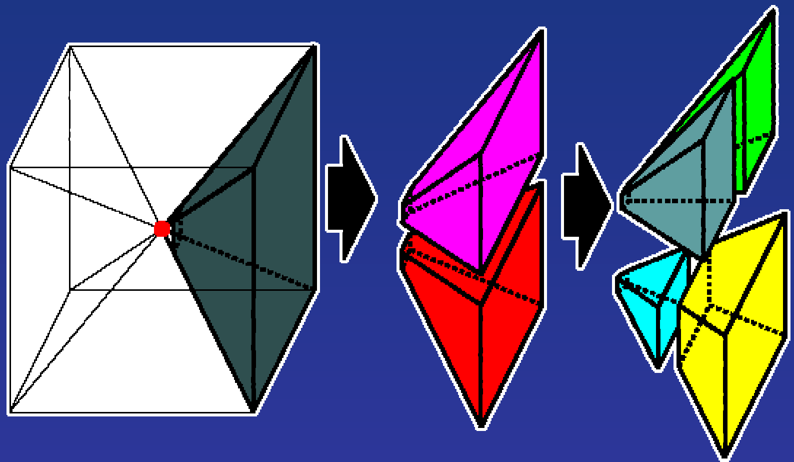




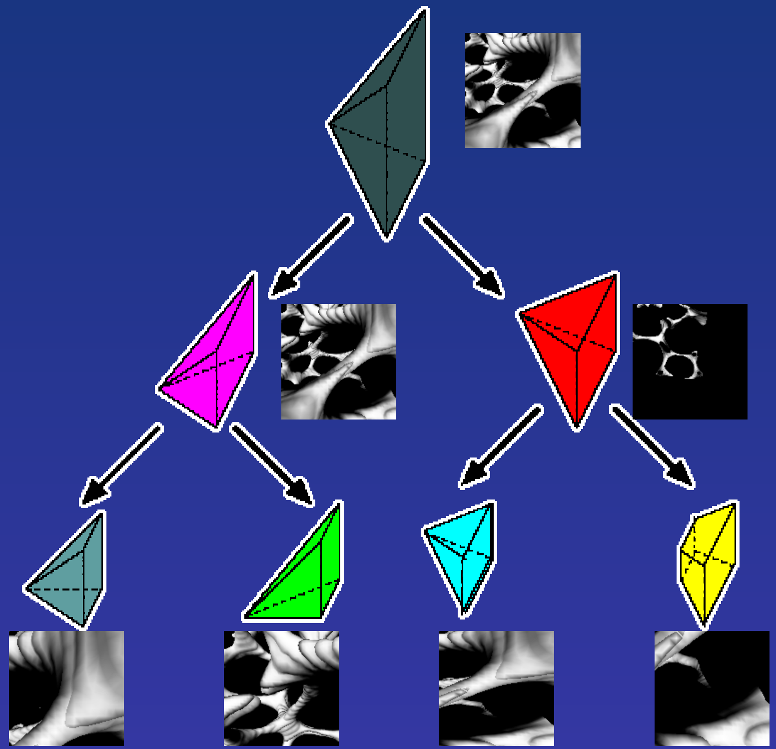



# A view-dependent refinement is performed on each cube face.

A pyramidal Kd-tree refinement is performed in image space.



A binary tree of tiles of equal resolution is maintained in the image database.

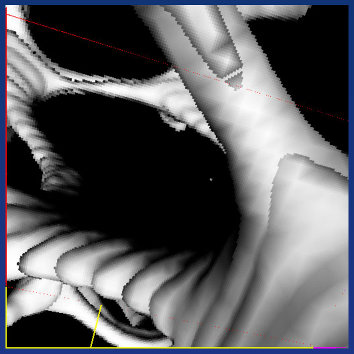


Valerio Pascucci

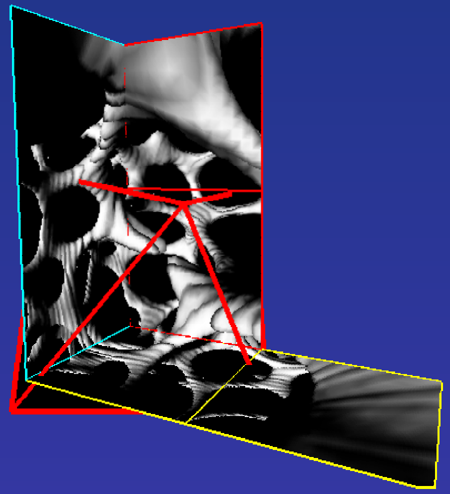
# At each view a Kd-tree refinement determines the new working set.

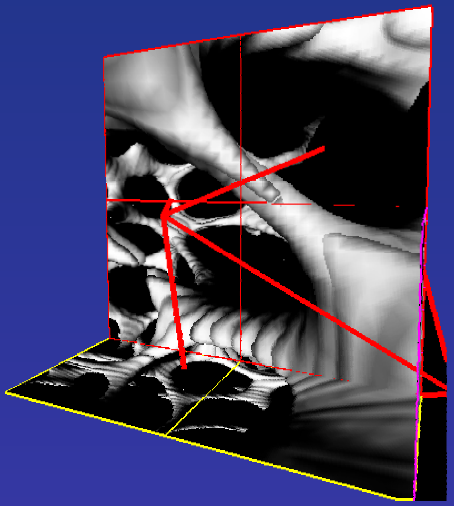
**View 1**

**View 2**



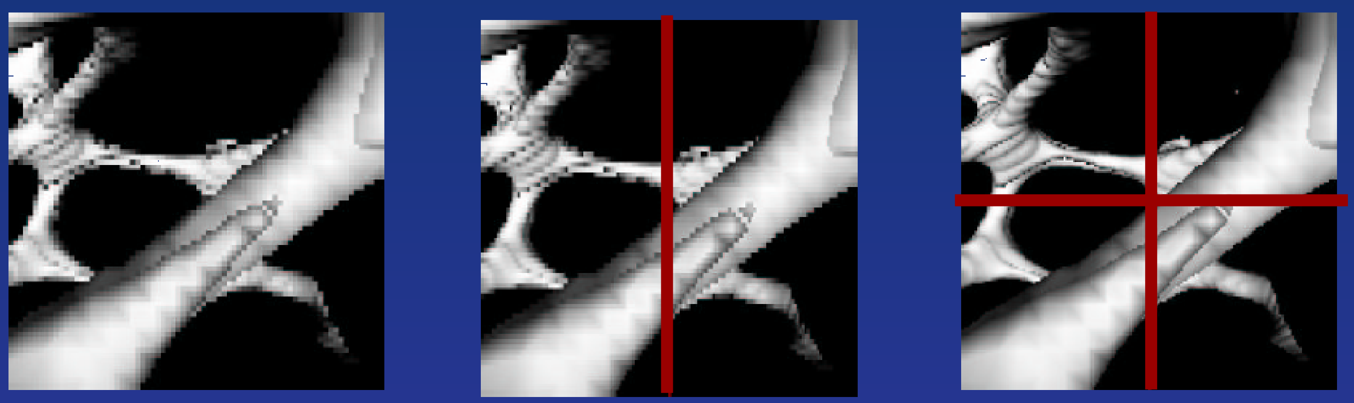
**The refinement determines the new images needed.**





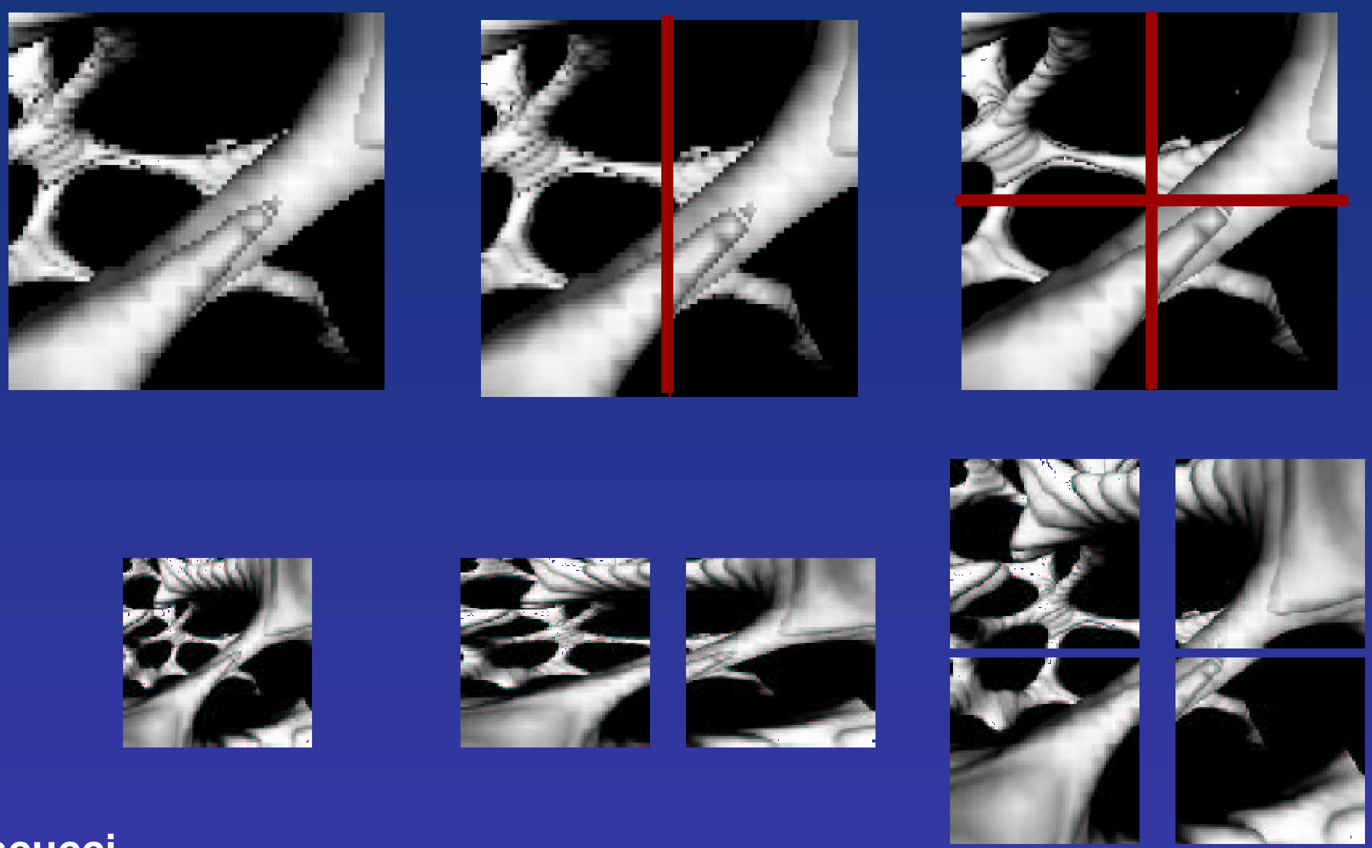
Valerio Pascucci

# Horizontal/vertical refinements require simple image stretching.



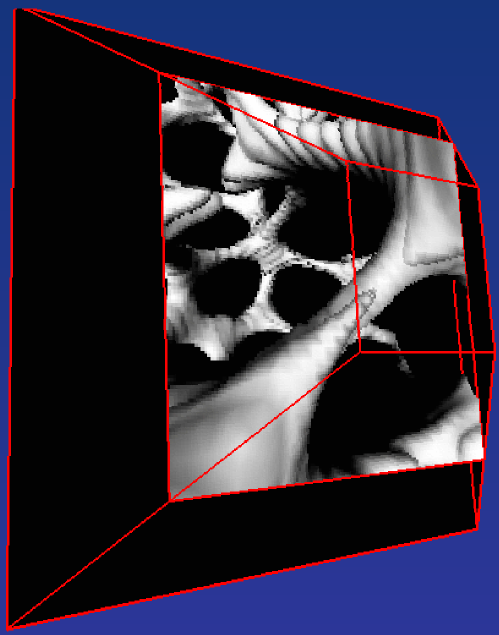
# Horizontal/vertical refinements require simple image stretching.

The missing images are temporarily extrapolated from those available.



Valerio Pascucci

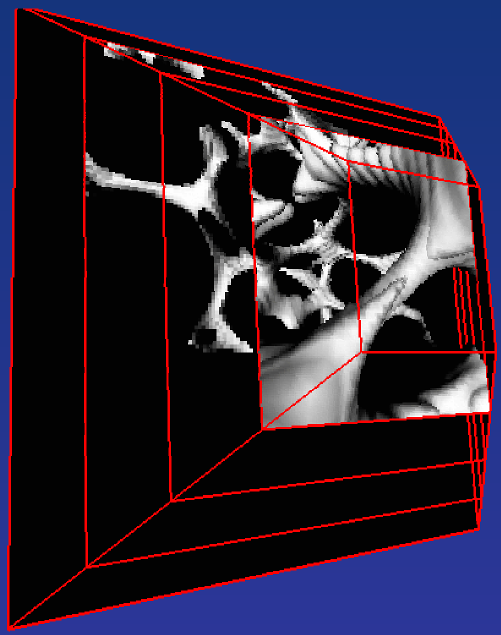
# Front-back refinement requires inverse compositing refinement.



Level 1

$$C_0 = C_1 + C_2\,(1-\alpha_0)$$
$$\alpha_0 = \alpha_0 + \alpha_1\,(1-\alpha_0)$$

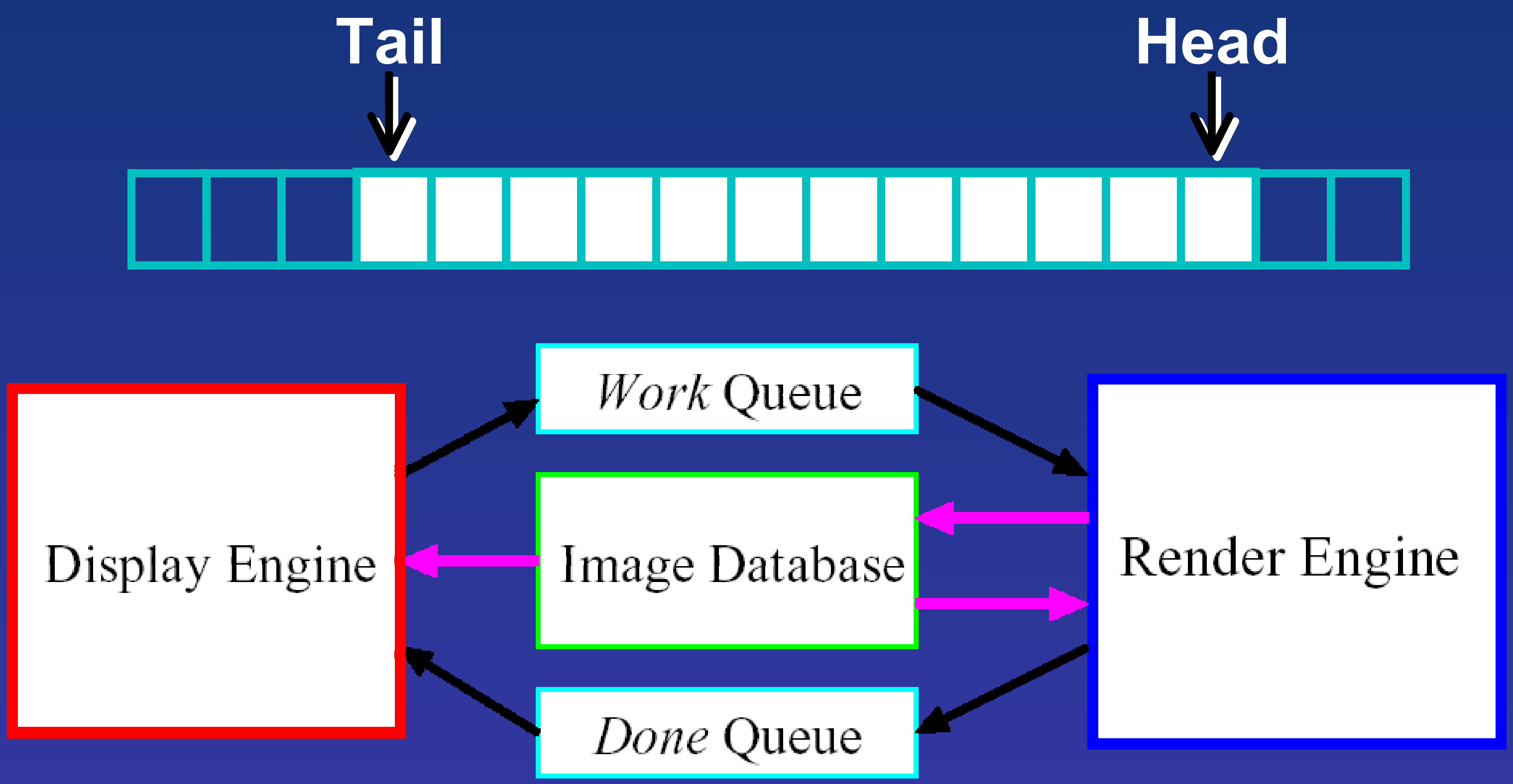
$$C_1 = C_2 = C_0\,/\,(2-\alpha_0)$$
$$\alpha_1 = \alpha_2 = 1 - \sqrt{1-\alpha_0}$$



Level 2

# We use two queues to avoid producer-consumer contention.

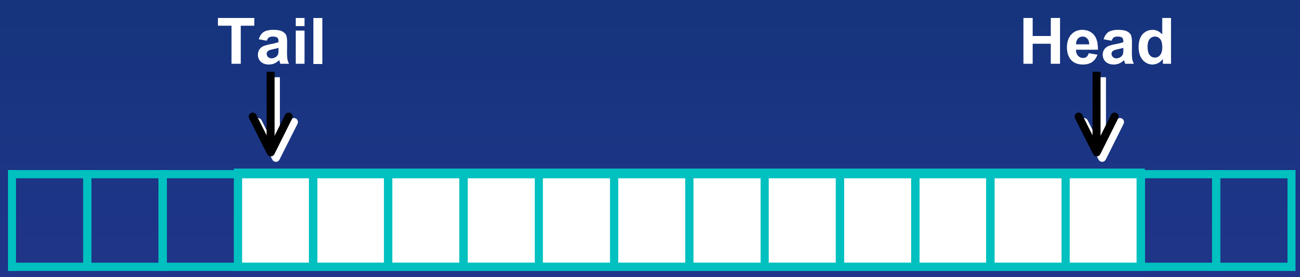
The display engine does not block the access of the rendering engine.

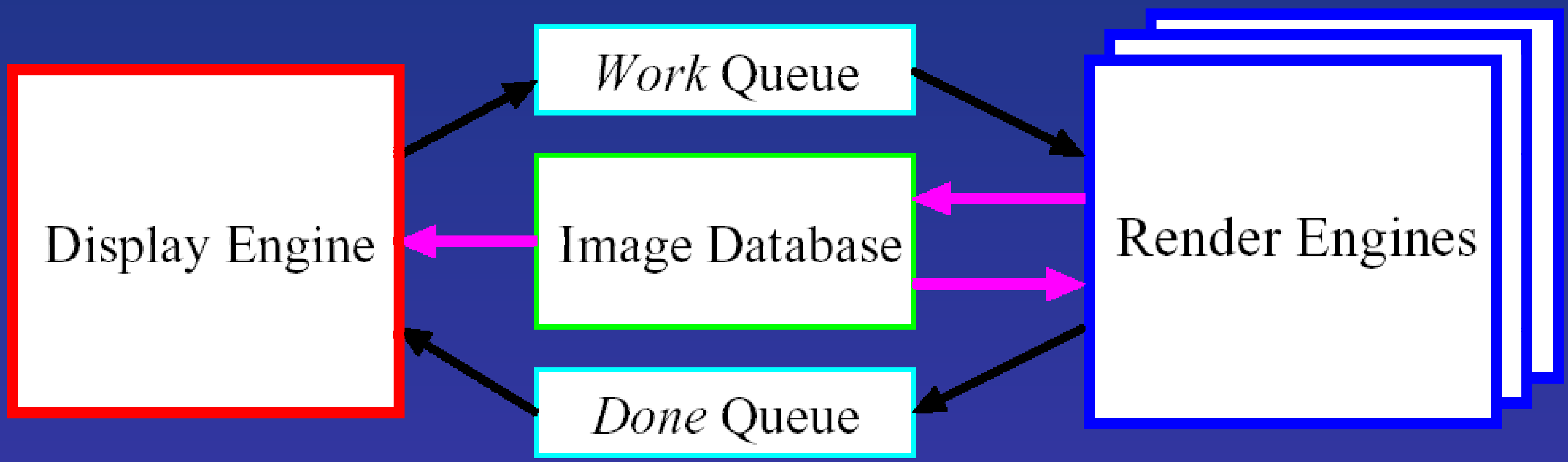



Valerio Pascucci

# Contention among rendering engines is not problematic.

The asynchronous nature of the system and the granularity of the work mitigates contention of rendering engines.



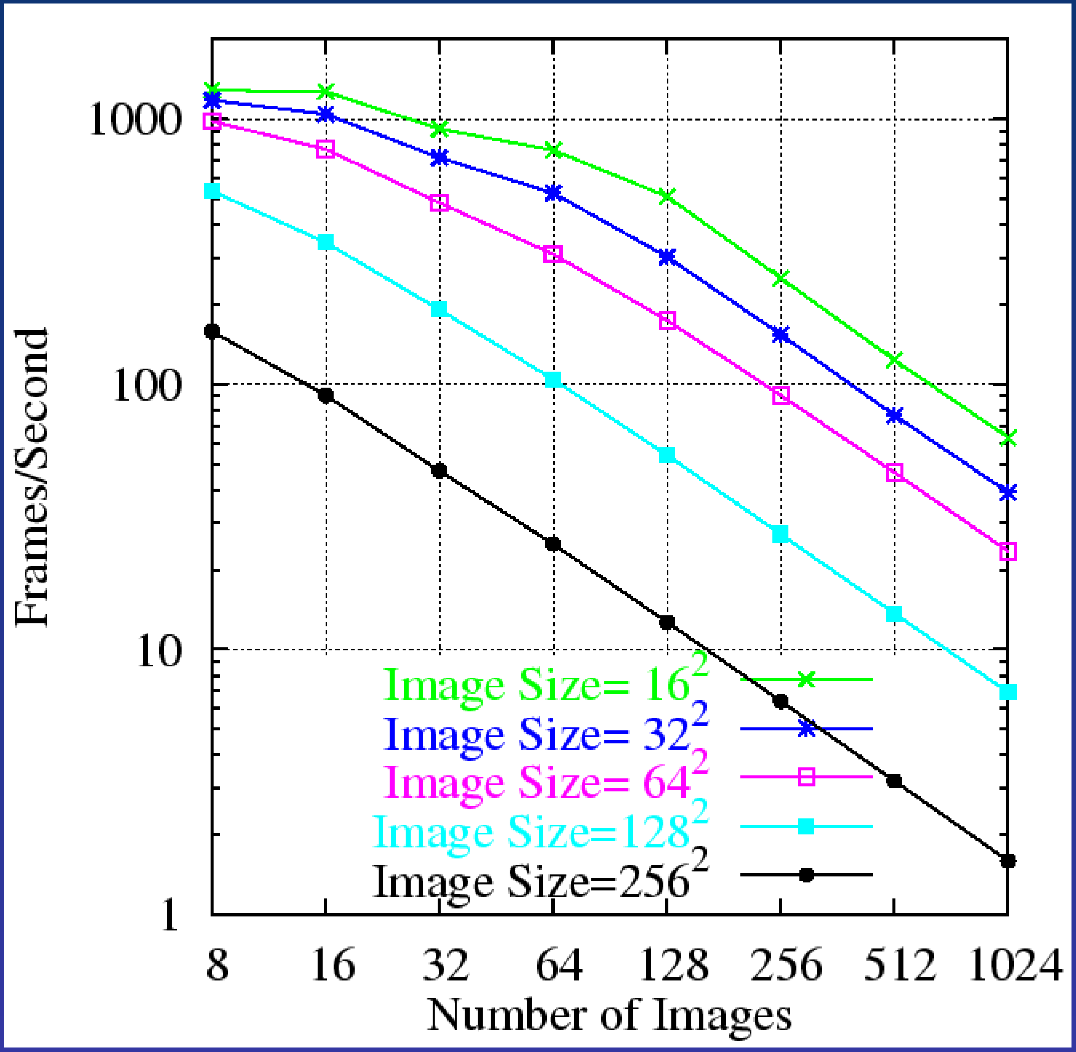
# We tested the scalability of the image cache on two platforms.

| | | |
|---|---|---|
| SGI Origin 3000, | 48 processors | (250MHz R10k) |
| Linux cluster, | 64 nodes | (2.5GHz Pentium) |

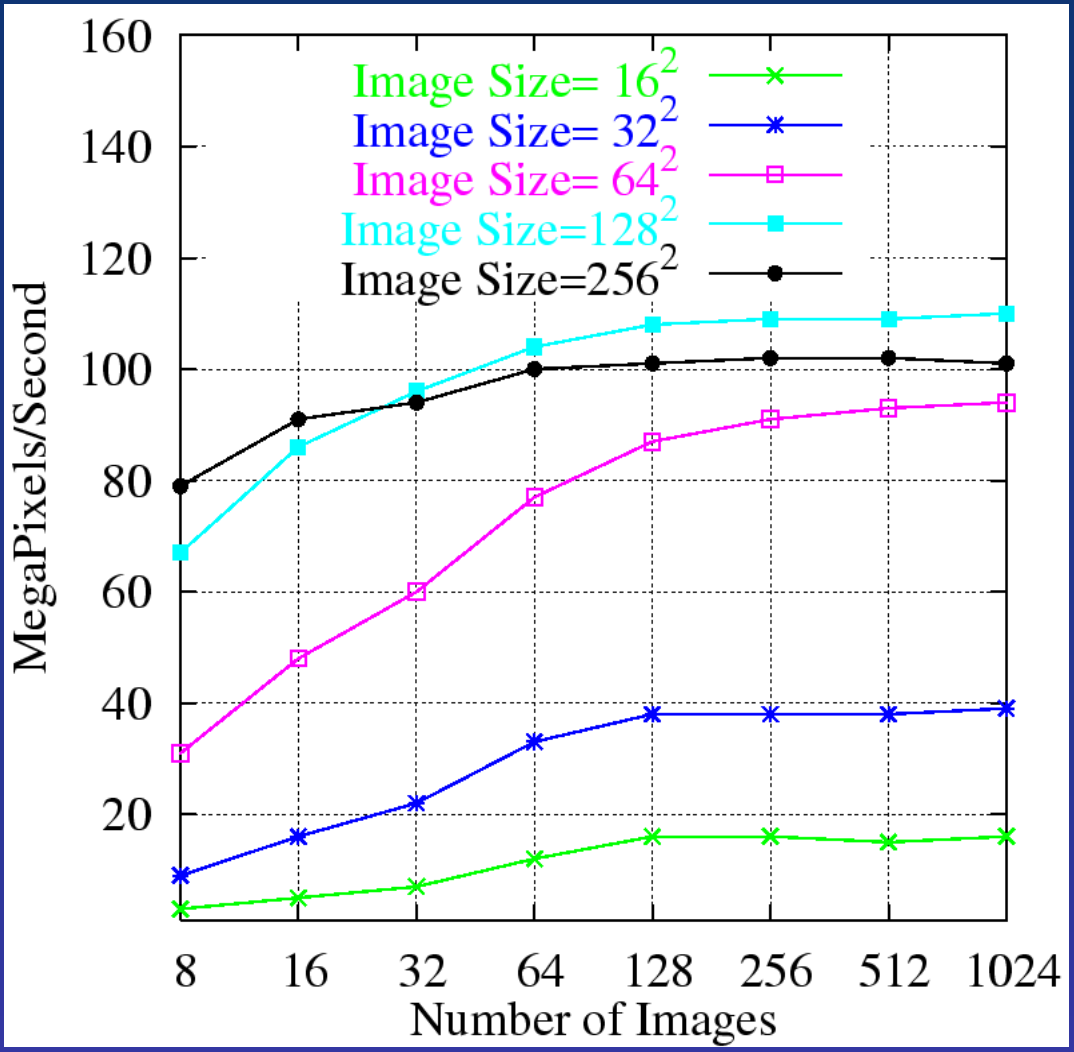
We tested the following parameters of the system:

- number of images in the scene
- number of rendering servers
- size of the images
- pixels per second
- network bandwidth utilization

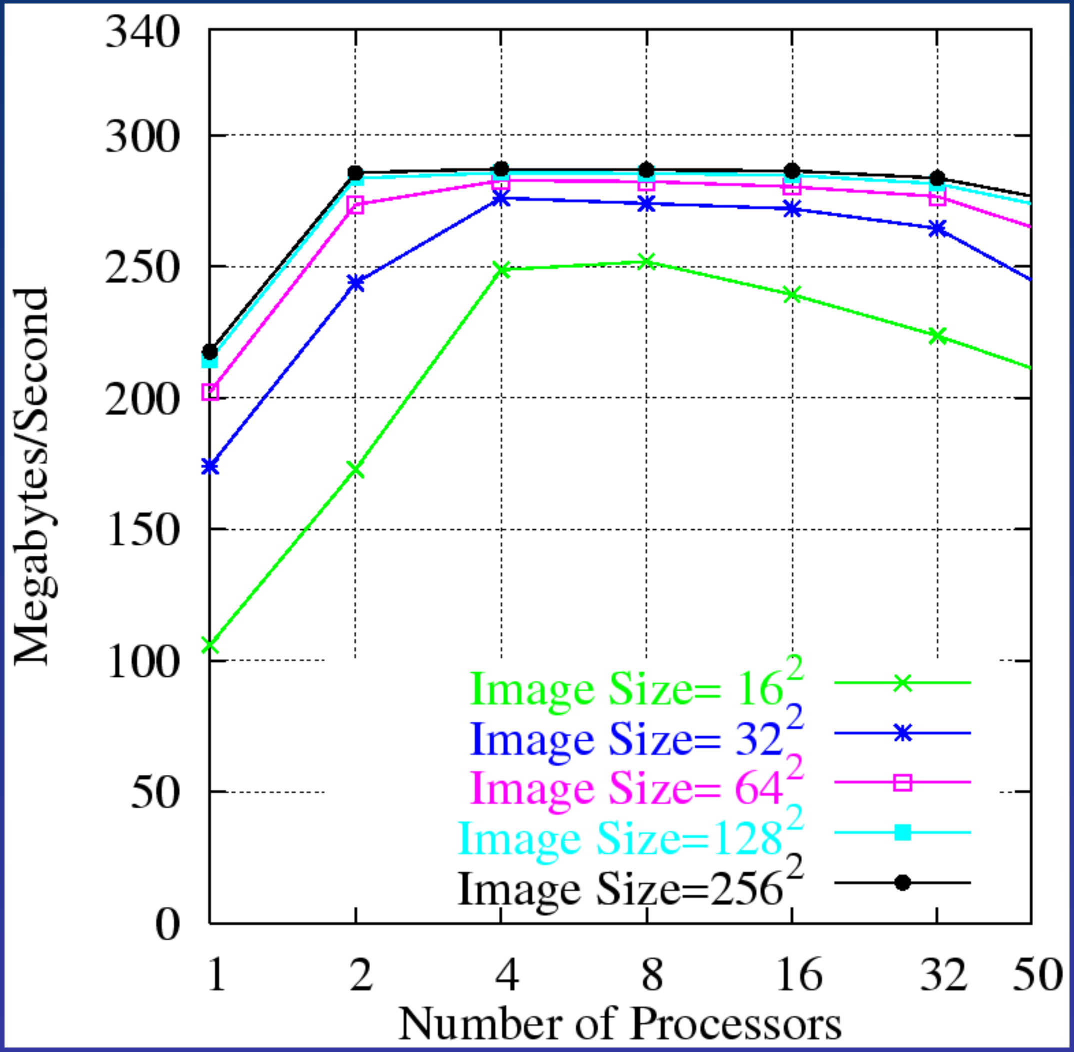
# Display speed is inversely proportional to number of Images.



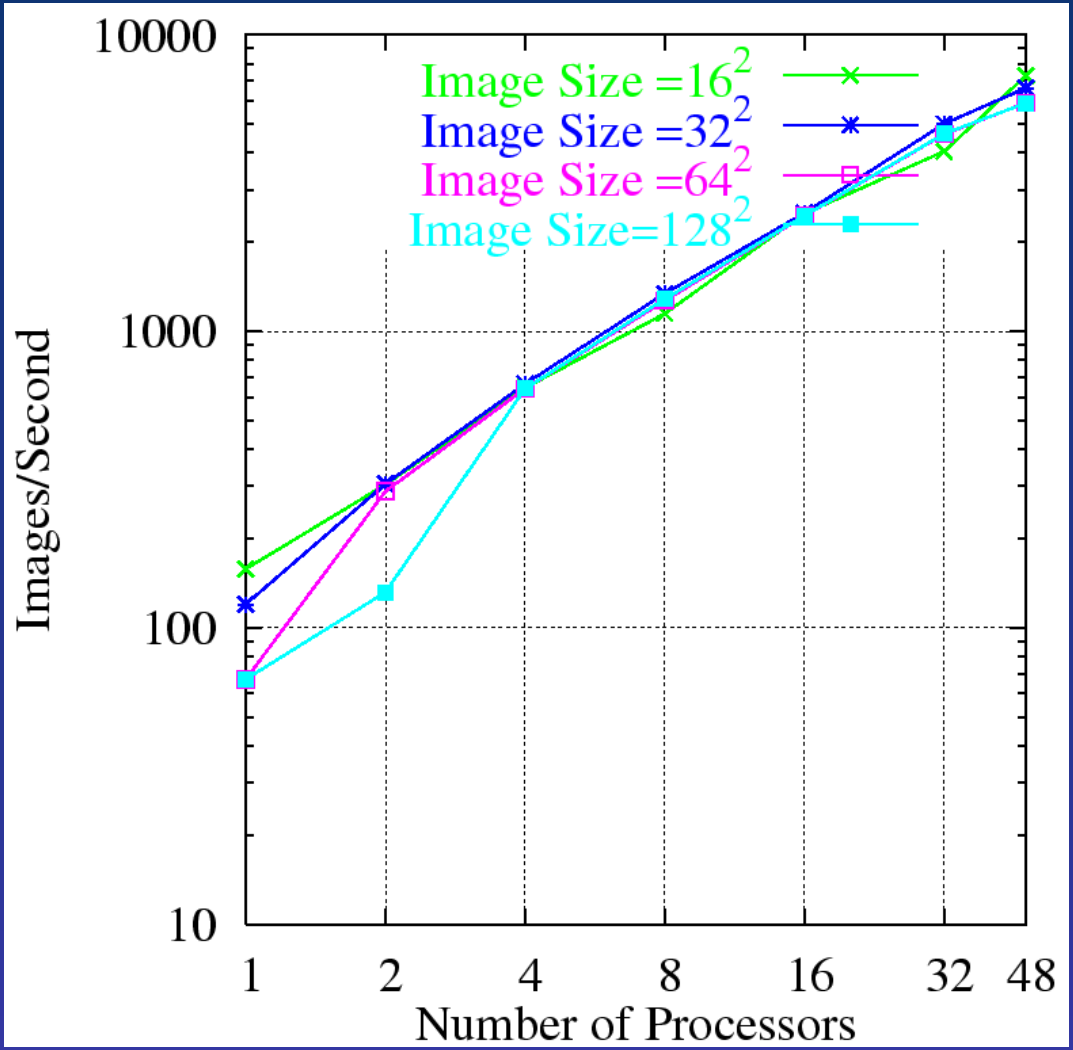
# Fill-rate performance suggest one particular image size.



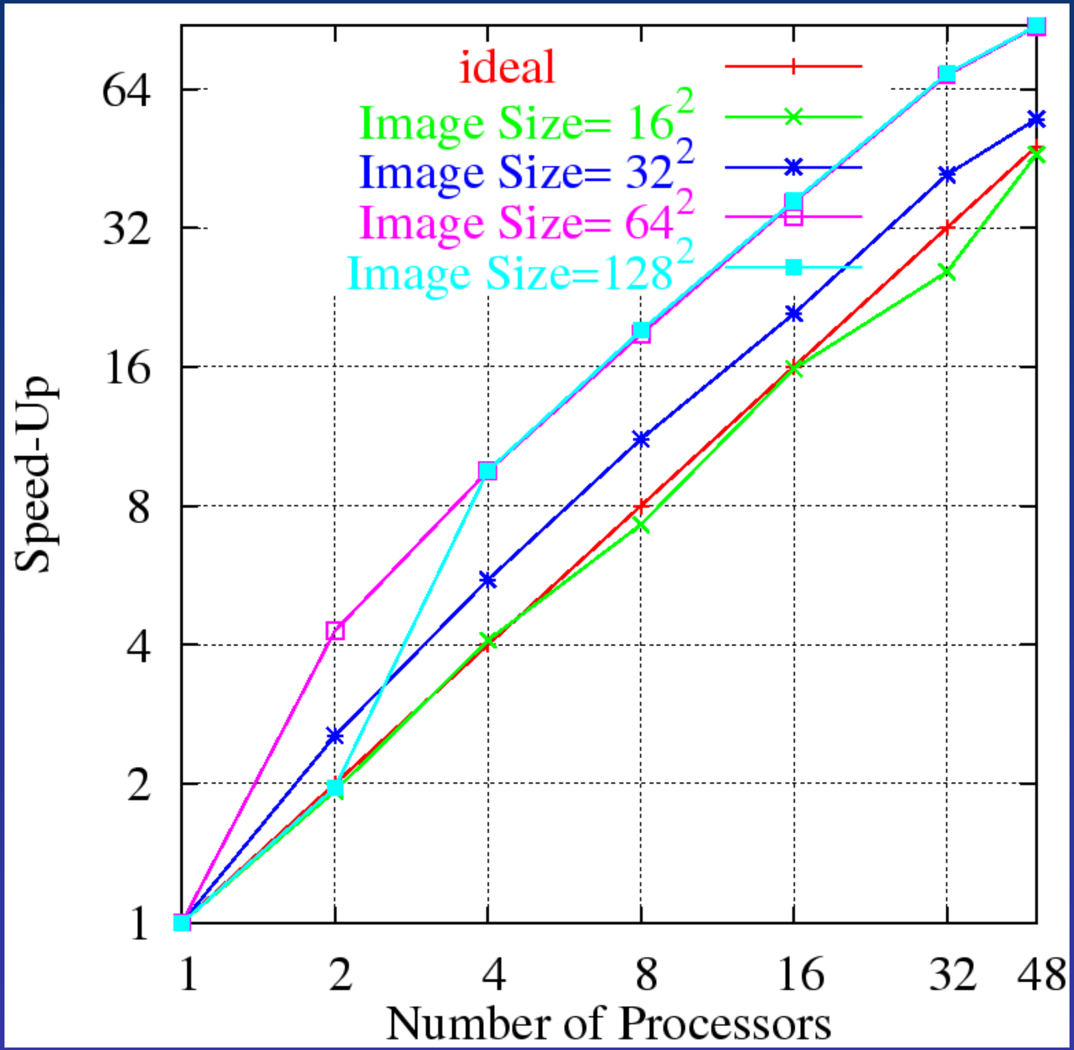
# Sustained bandwidth utilization close to ideal pick performance.



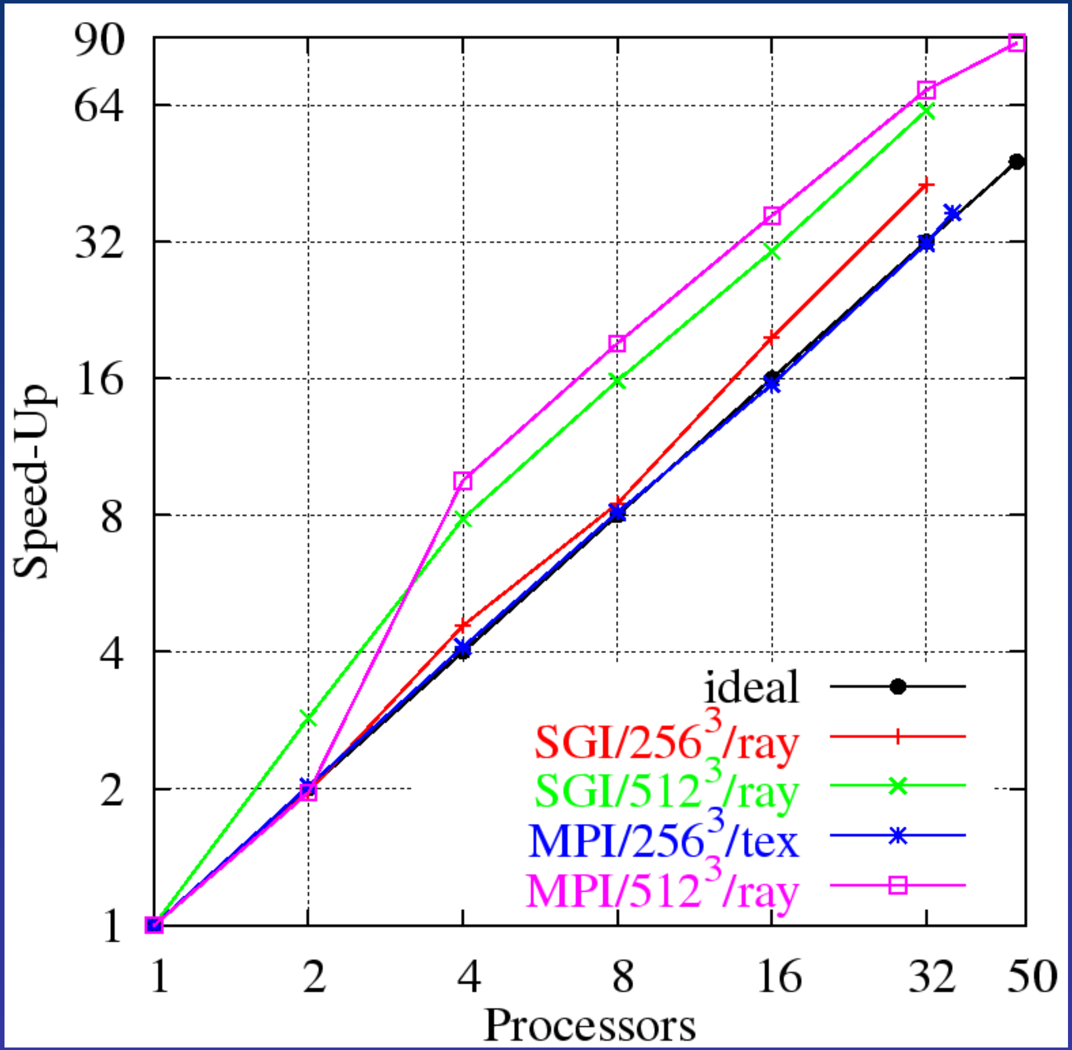
# The system scales well with the number of processing units.



# The system scales well with the number of processing units.



# The behavior of the system is consistent across platforms.



# Current directions

- Heterogeneous distributed system.
- Combined multi-resolution representation of the object.
- Scaling to high resolution output power-walls.
- Combination with other techniques?
- Development of prediction agents for data pre-fetching.

# UCRL-PRES-200743


- This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.