

Moore's Wall?

Technology Advances and Online Game Design

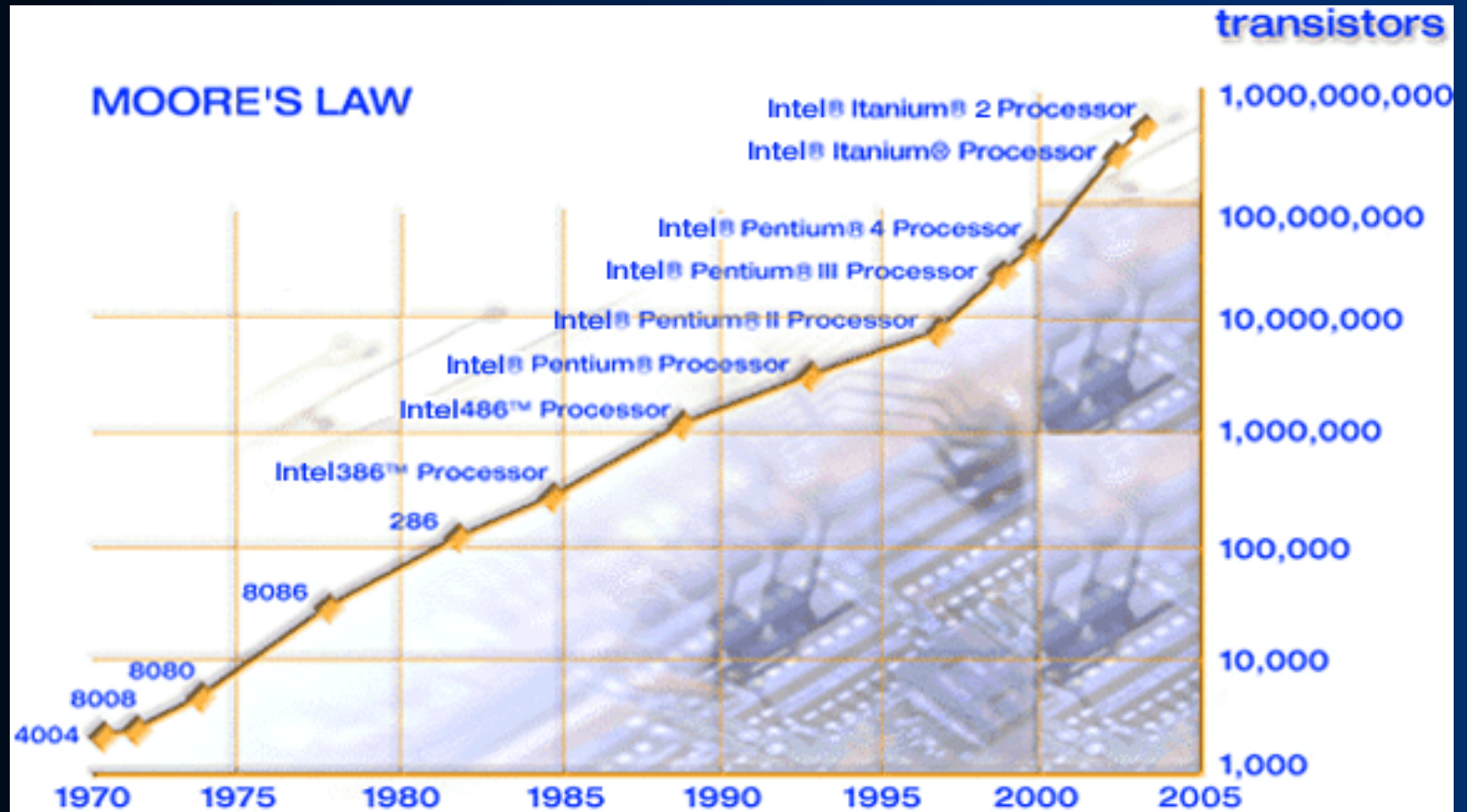
Raph Koster
Chief Creative Officer
Sony Online Entertainment

The march of technology

- CPU has risen according to Moore's Law
- Along with it, our capabilities have increased as well
- And our budgets keep pace too!

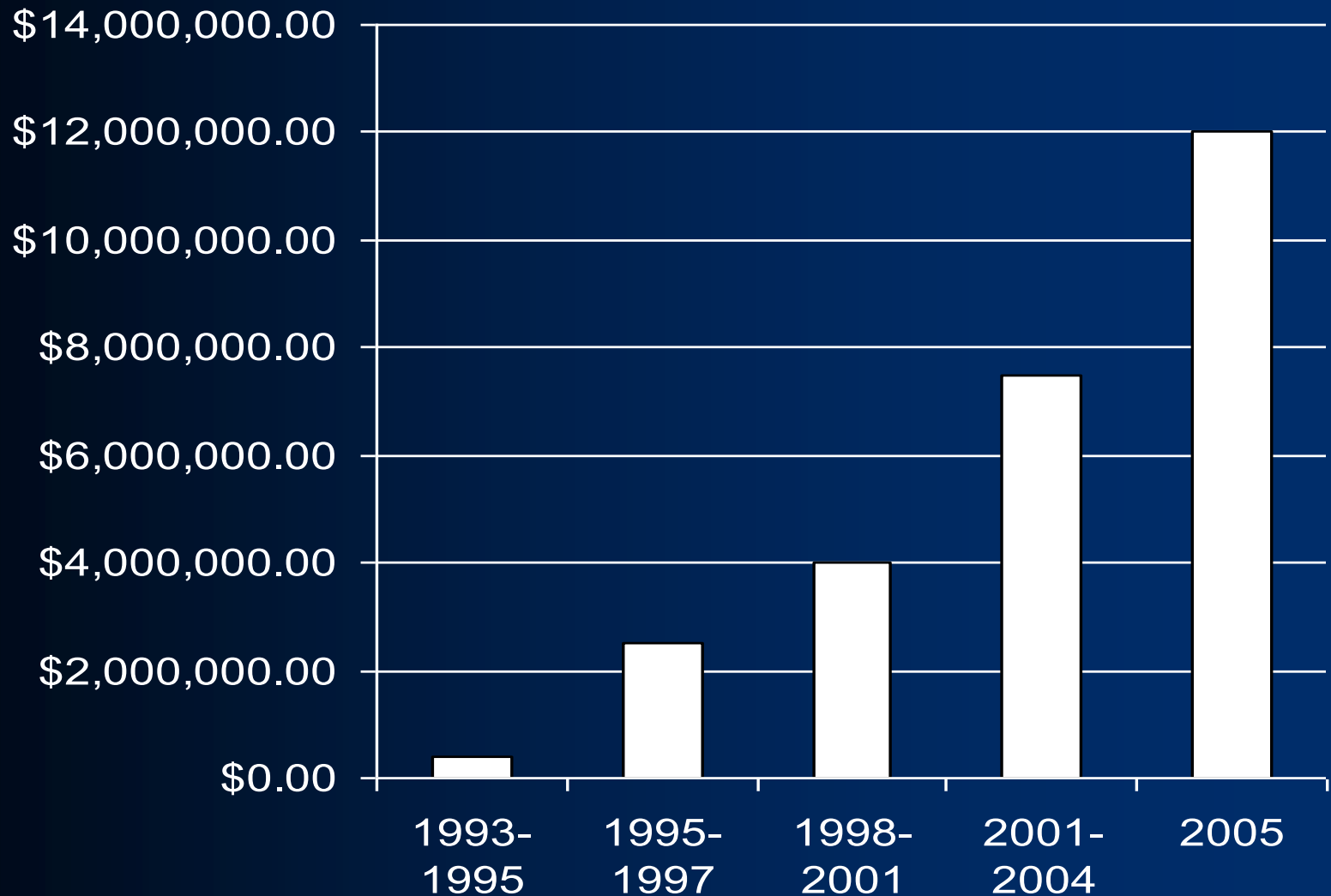
CPU power over time

- Graph from Intel Research



“Software is a gas; it expands to fill its container.”

Historical game budgets



AAA game was

\$50,000

64k on disk

And in 2004 was

\$10,000,000 (not including MDF!)

1.4GB on disk

Historical game sizes

In 12 years, budgets have gone up by a factor of

$22x^*$

The amount of data that has to be created has risen by

$40-150x$

- To some degree, we're winning the war—we're up to six times as efficient at making data for the dollar...

*in CPI-adjusted dollars, and *excluding* MDF!

The causes

- Gameplay elements have not become that much more complex
- Enabling gameplay technologies have made game development far more expensive and complicated
- Technology has focused on presentation rather than functionality

1994



1996



1998



1999



2003



2004



The fundamental problem

Technology is curtailing
game creativity, rather than
assisting it.

- Heavier content load than standalone games
- Already more technically challenging
- Budgets higher than standalone games

40% or more of CPU load goes to just solving A pathfinding!*

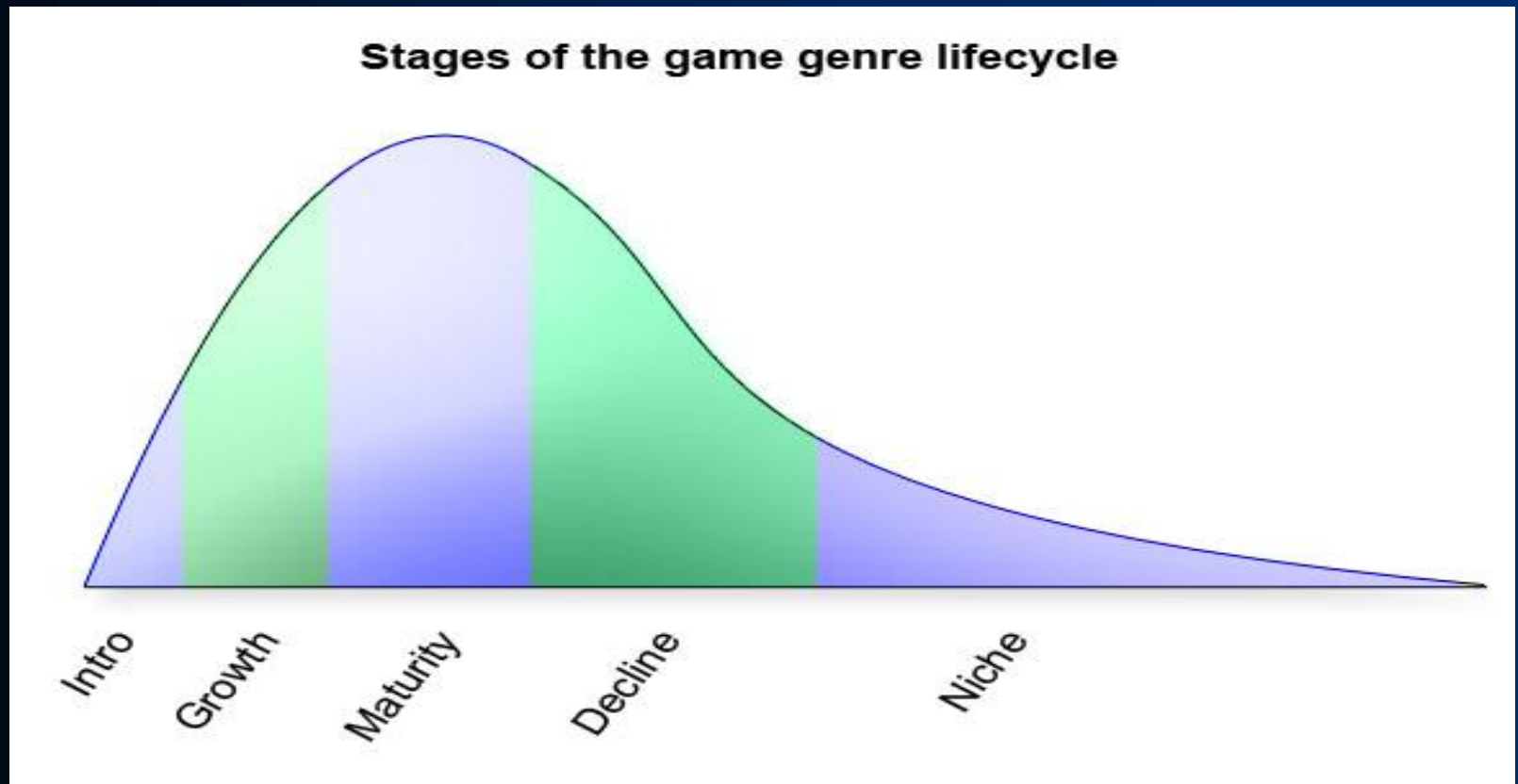
Technology as canvas

- Technology has primarily been a canvas
 - Rather than extending the canvas, we tend to fill it up with software
- We find more colors, more textures, and more stuff to pile into the limited space
 - Which makes it more expensive to create

- Real world factors take a long long time to catch up to capacity
 - Alan Kay was playing with “tablet PCs” in 1967! But they were unreasonably expensive
 - The initial PC revolution was driven by microprocessors that ran the *keyboard* in an Alto

Limitations of content

- New games typically calcify into genres
 - This limits acceptance of innovation



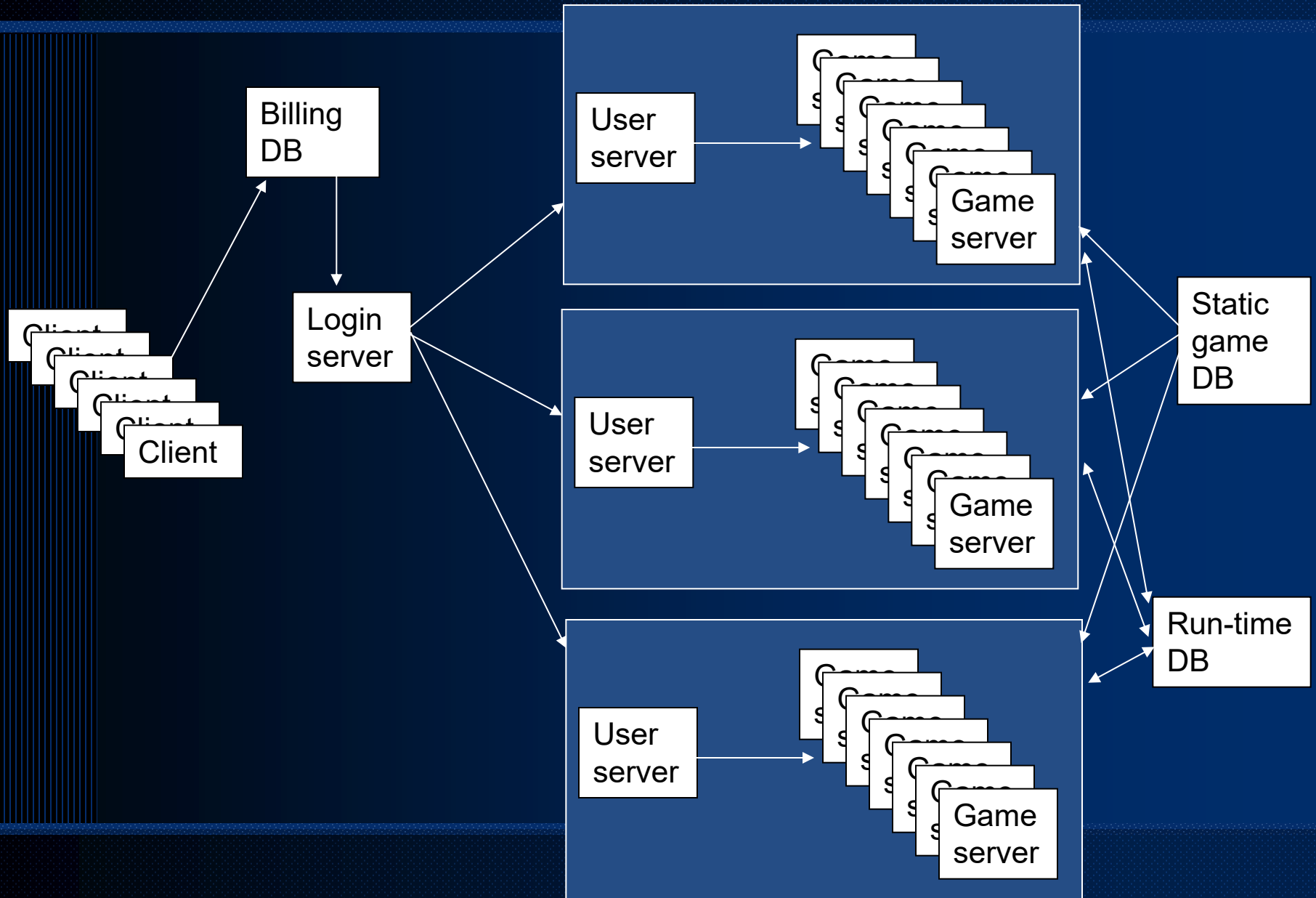
Massively multiplayer

- The most expensive games to make
 - Budgets run up to 3x the AAA budgets you just saw
 - 400 hours of content needed, not 8-12
- Traditionally less emphasis on visuals
 - But this is changing
- The most technically challenging games to develop
 - DB support, networking, fault-tolerance, etc...

Leveraging the power of computers

- Computers excel at
 - Simulation
 - Algorithms (solving smallish problems predictably)
 - Networking (sharing data quickly and easily)

MMO infrastructure



Tech solves many problems

- Networking
- Billing infrastructure
- Authentication

- But we're left with the issues of the run-time and static DBs

- The run-time DBs are extremely limited in terms of data variability
 - We can't do worlds that change much
 - We can't track too much data per object (since we track billions of objects)
- The static DBs demand massive amounts of content
 - We spend tens of millions to generate it
 - Then it cannot change
 - It falls prey to obsolescence

How can tech help?

Procedural content

Sandbox design

User-created content

Procedural content

- CPU-dependent
- Permits easy streaming of seed data
- Often generic and unsatisfying

Sandbox design

- Provide players with tools to amuse themselves
- Self-refreshing content for self-starting players
- Fails players who seek directed content (which is most of them, these days)

User-created content

- The engine behind many powerful franchises
 - Sims, Half-Life, Quake, D&D, etc
- Around 1% of users create decent content
- Tools are far too difficult

If CPU power on desktops never increased again, we'd still have lots of room to grow in massively multiplayer.

- We've barely tapped the above three approaches
- We're approaching "uncanny valley" issues with visuals
- It would force innovation on the marketplace

But tech IS improving

So let's spend the cycles where they
make a difference:

- AI
- Simulation
- Tools