Capabilities on the rise for the end user
“...we see a very significant shift in what architectures will look like in the future...

fundamentally the way we've begun to look at doing that is to move from instruction level concurrency to ... multiple cores per die. But we're going to continue to go beyond there. And that just won't be in our server lines in the future; this will permeate every architecture that we build. All will have massively multicore implementations.”

Intel Cancels Top-Speed Pentium 4 Chip
Thu Oct 14, 6:50 PM ET  Technology - Reuters
By Daniel Sorid

Intel ...canceled plans to introduce its highest-speed desktop computer chip, ending for now a 25-year run that has seen the speeds of Intel's microprocessors increase by more than 750 times.
Lots more computing power
Los Alamos Computing Center? No a single CPU chip.

**Desktop:** 200 mm², 100 W, $400

- LPIA x86
- LPIA x86
- DRAM ctrlr
- DRAM ctrlr
- OoO x86

- LPIA x86
- LPIA x86
- 1 MB cache
- 1 MB cache
- 1 MB cache

- PCIe ctrlr
- PCIe ctrlr
- NoC
- NoC
- 1 MB cache

- LPIA x86
- LPIA x86
- 1 MB cache
- 1 MB cache
- 1 MB cache

- LPIA x86
- LPIA x86
- DRAM ctrlr
- DRAM ctrlr
- OoO x86

**Server:** 350 mm², 120 W, $2000

- LPIA x86
- LPIA x86
- DRAM ctrlr
- DRAM ctrlr
- DRAM ctrlr
- DRAM ctrlr
- LPIA x86
- LPIA x86

- LPIA x86
- LPIA x86
- 1 MB cache
- 1 MB cache
- 1 MB cache
- 1 MB cache

- PCIe ctrlr
- PCIe ctrlr
- NoC
- NoC
- NoC
- NoC
- PCIe ctrlr

- LPIA x86
- LPIA x86
- 1 MB cache
- 1 MB cache
- 1 MB cache
- 1 MB cache

- LPIA x86
- LPIA x86
- OoO x86

**Ultra-Mobile:** 40 mm², 5 W, $50

- LPIA x86
- LPIA x86
- DRAM ctrlr

- LPIA x86
- LPIA x86
- 1 MB cache
- 1 MB cache

- PCIe ctrlr

(2008 45 nm process)
Many Core in a Nutshell

1. We believe user experiences will benefit from 100 fold improvements in computational power.
2. Because of physics, you won’t get this power from frequency scaling or programmer transparent hardware like ILP. The only way is parallelism.
3. Chips with many heterogeneous cores can be manufactured now. Only with chips like this can such performance scaling continue.
4. There are difficult but solvable issues related to memory bandwidth and I/O for this to work well over-all.
5. There are other benefits from such chips like good power and manufacturing characteristics.
6. Application of such hardware does not appear to be limited by any intrinsic lack of parallel algorithms.
Constructing parallel applications
- Encapsulating parallelism in reusable components
- Integrating concurrency & coordination into existing programming languages
- Raising the semantic level to eliminate sequencing
- Reducing the complexity of debugging, tuning and testing

Executing fine-grain concurrent applications
- Managing large amounts potential concurrency
- Supporting lightweight transactions
- Interoperating with legacy thread model and interfaces
- Evolving hardware to effectively support parallel programs

Coordination of system resources and services
- Assigning resources securely
- Hosting concurrent operating environments
- Managing resources cooperatively
- Providing concurrent system services
- Managing heterogeneous resources
Manycore Stack

- Applications
- Libraries
- Languages, Compilers and Tools
- Concurrency Runtime
- OS Kernel
- Partitioning Hypervisor
- Hardware

Constructing Parallel Applications

Executing Fine-Grain Parallel Applications

Coordinating System Resources and Services
Developer Impact

Programming Abstractions

- Asynchronous Agents
- Concurrent Collections
- Transactions

Existing Languages
- Data Parallel Extensions
- Transactions
- API for messaging

New Languages
- Safety & Portability
- Streaming Data Parallel
- Functional Parallel

Infrastructure
- Transaction & Parallel Safe
- Numerical Libraries
- Declarative Languages

Developer Tools

- Visual Designers
  Agents, Dataflow
- Diagnostics
  Debugging & Performance Tuning
- Testing & Verification
Simulating The Physical World

DEM
- Space Shuttle
- GTOPO 30
- NED (US)
- EarthSat (Europe)
- GeoScience (Australia)

Landclass
- Tilling Textures
- Satellite Imagery

Vectors
- Roads
- Lakes
- Coasts
- Pipelines
- Power Lines
- Etc.

World
- Weather
- Seasons
- Celestial Catalogue
- Moon Phases
- Time Zones
- Magnetic Variation

Autogen
- Buildings
- Houses
- Trees
- Fences
- Etc.

Objects
- Landmarks
- Bridges
- Dams

Facilities Data
- Jeppesen Navdata
- NOAA Hazards
- DAFIF
- FAA

Vehicle Simulation
- Airplanes
- Boats
- Ground Vehicles
- Cars & Trucks
- Trains
- Ambient Traffic

AI Paths and Waypoints
- Jeppesen
- Nortech

Triggers
- Text
- Audio
- Cameras
- Animations
- Cut Scenes

After Action Review
- Goals
- Timing
- Scoring
Gameplay Simulation
Technical Computing Is Far Reaching

Earth Sciences

Life Sciences

Social Sciences

New Materials, Technologies and Processes

Multidisciplinary Research

Computer and Information Sciences

Math and Physical Science

E=MC²
Reduced Time To Insight

Computational Modeling

Sensors

Persist

Data Mining & Algorithms

Interpretation & Insight

Computational Modeling
A transformative example

Modeling of the instrumented world

First Responder Scenario (empowering users to make good decisions)

First responders need to quickly and safely experience unfamiliar areas under different disaster scenarios before they deploy into it. For example, earthquakes in San Francisco.

Setup Environment

Explore Possibilities

Monitor and Take Action
Scenario Features ➔ Tech Domains

Modeling the instrumented world
Scenario: First responders need to quickly and safely experience unfamiliar areas under different disaster scenarios before they deploy into it.

Technology Domains: To realize this scenario, we need to compose these technologies and accelerate their performance at least 10-fold on the manycore client.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Machine Learning</th>
<th>Math</th>
<th>Physics</th>
<th>Statistics</th>
<th>Image Processing</th>
<th>3D Modeling</th>
<th>Rendering</th>
<th>Database</th>
<th>Query engine</th>
<th>Index engine</th>
<th>Data compression</th>
<th>Motion Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stitching scenes together (2D, 3D)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object recognition in scene</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interaction (navigation) in scene</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linking scene objects to live data</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filling in missing information</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image and video correction</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long running scene statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personalization of content</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Simulating physical events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Real-time scene update</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
To realize this scenario, we need to compose these technologies and accelerate their performance at least 10-fold on the manycore client.

**Technology Domains**

<table>
<thead>
<tr>
<th>Platform: To compose technology domains and accelerate their performance 10-fold, we need to build the following platform components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrating concurrency into existing languages</td>
</tr>
<tr>
<td>Machine Learning</td>
</tr>
<tr>
<td>Math</td>
</tr>
<tr>
<td>Physics</td>
</tr>
<tr>
<td>Statistics</td>
</tr>
<tr>
<td>Image Processing</td>
</tr>
<tr>
<td>3D Modeling</td>
</tr>
<tr>
<td>Rendering</td>
</tr>
<tr>
<td>Database</td>
</tr>
<tr>
<td>Query engine</td>
</tr>
<tr>
<td>Data compression</td>
</tr>
<tr>
<td>Gesture/Motion Input</td>
</tr>
</tbody>
</table>

Tech Domains ➔ Platform
Application/Libraries Architecture

Applications

- Personal Assistant
- Natural UI
- Program Development
- Technical Computing
- System Design
- Information Management
- Games & Entertainment
- Biology/Health care
- Business Intelligence
- Robotics

Application Services

- Speech Engine, Gaming (Physics/AI), Unified Communications
- Database
- Vision
- Machine Learning
- Semantic Processing

Domain Specific Libraries

- Single domain physics, search algorithms, audio/video characterization and extraction, biology simulations, optimization and constraint resolution

Base Libraries

- Common data structures and algorithms: trees, graphs, tables, sorting, traversal.
Isolation and cooperative device support provided by VM/OS

Resource Management between OS/Runtime and VM/OS (under Machine Management)

Scalable Services provided by OS (async, cancellable, thread affinity free)