

Software-Defined Servers

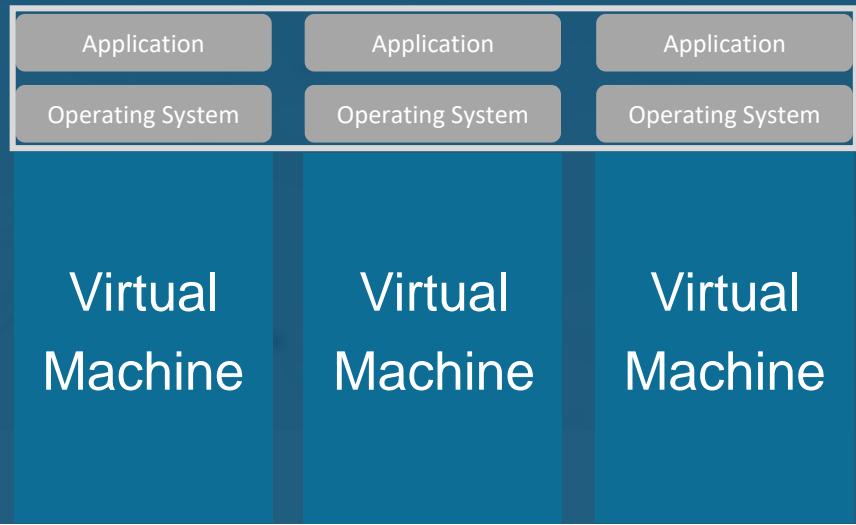
Flexible - Fast - Easy

TidalScale

TidalScale

Traditional Virtualization

Virtual



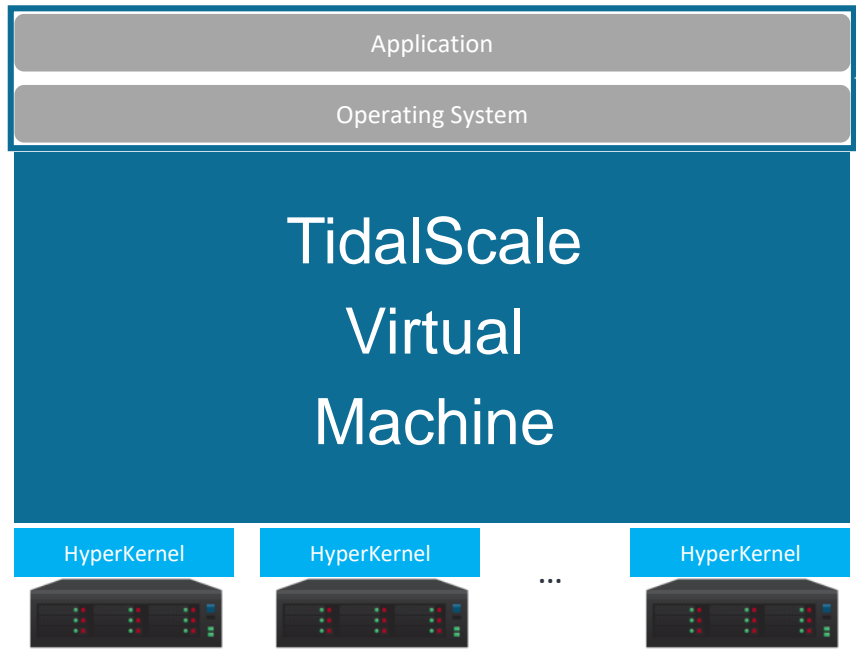
100%, bit-for-bit unmodified

Physical



Multiple virtual machines share a single physical server

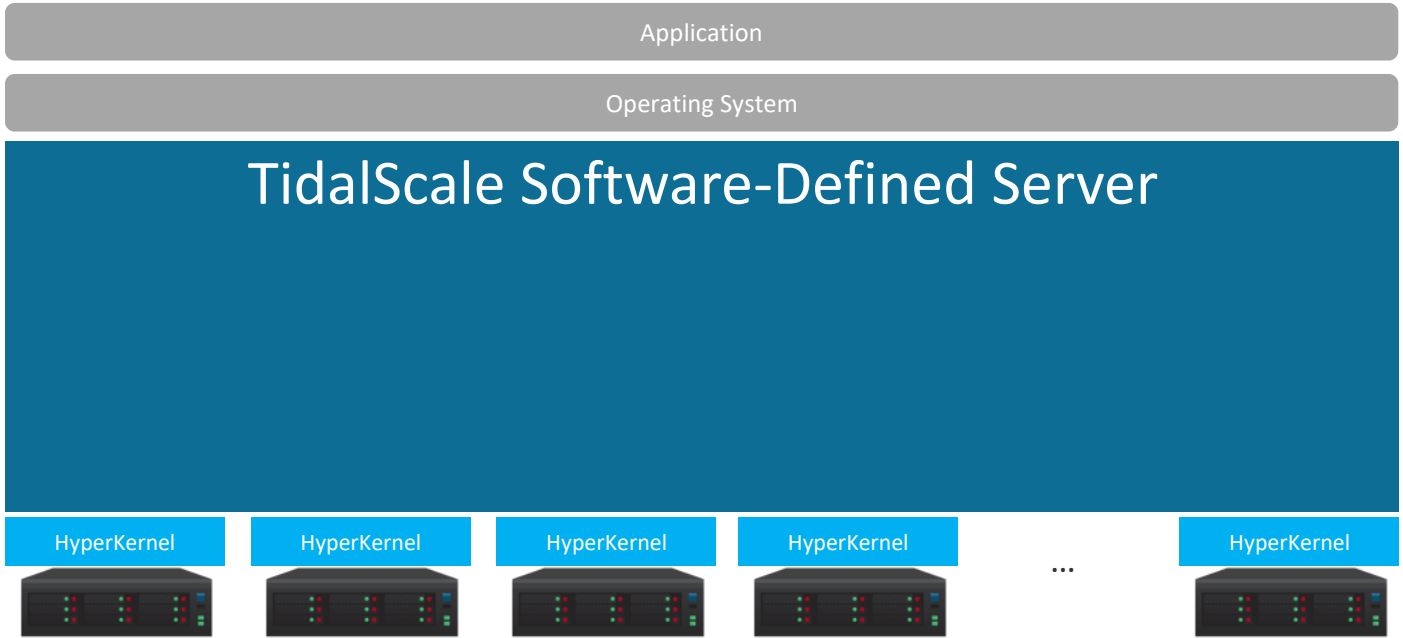
TidalScale Software-Defined Servers



100%, bit-for-bit
unmodified

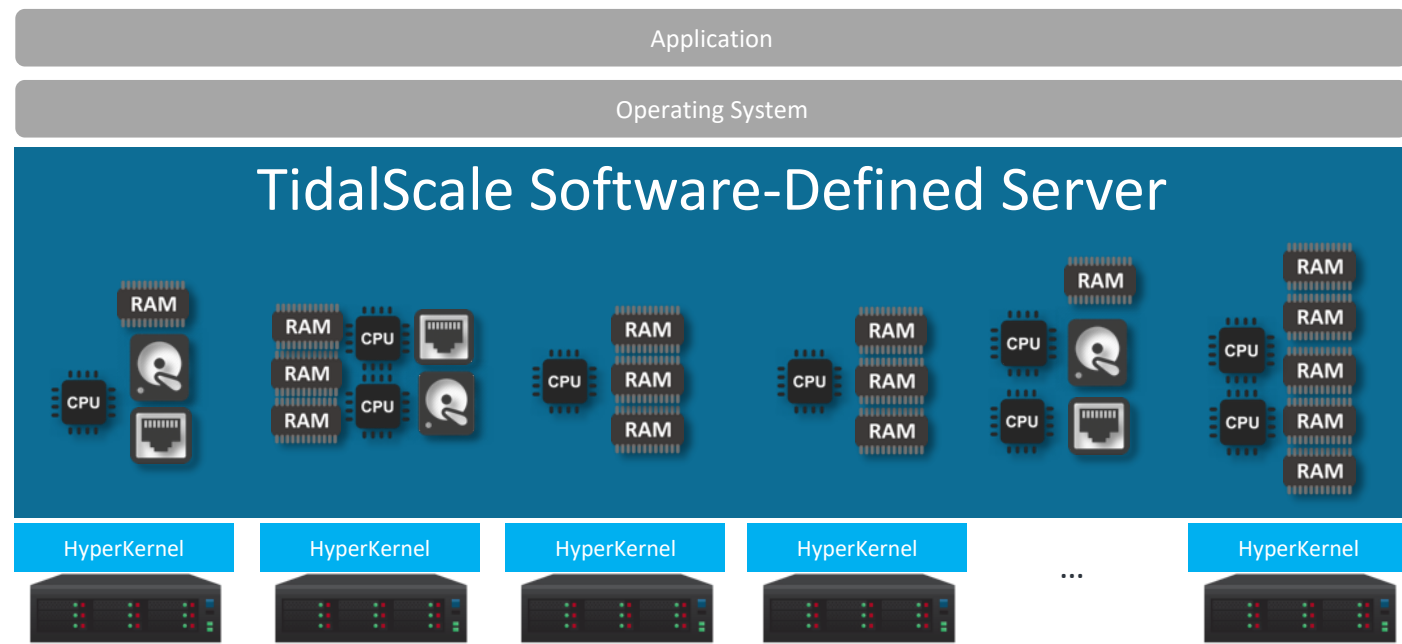
Single virtual machine spans multiple physical servers

Seamless Scalability



Flexible – Scales Up or Down Quickly

Machine Learning-Driven Self Optimization



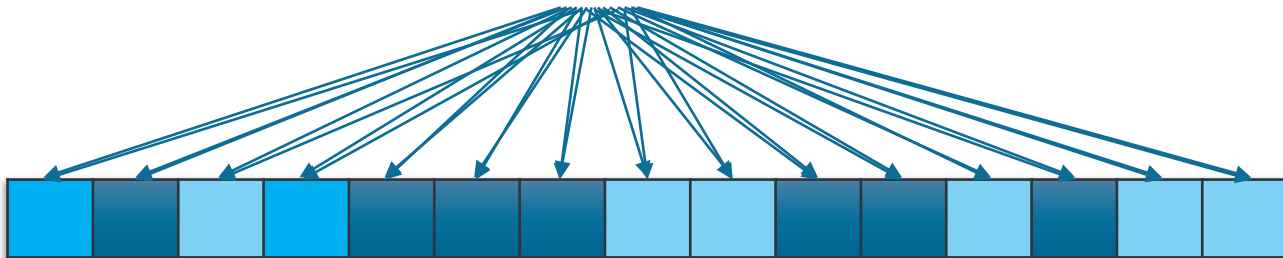
Uses patented machine learning to transparently align resources

Rethinking Programming with Large Memories

- Optimize for large memories rather than conserving memory - let the hyperkernel optimize it
 - Ingest large amounts of data in parallel. They will help with locality by distributing the working sets across the nodes
 - Once ingest is complete, don't move it. This works well for historical data, since history is constant. It doesn't change.
 - Since every byte of memory is directly addressable by every processor, build the indexes you need to operate efficiently USING POINTERS.
 - Example - Sorting:



Sorting with Pointers



- Sort the pointers to rows in the table, not the tables themselves
- THIS IS NEW!!! We've never been able to do this before, since the memories we use were never large enough.
- See my upcoming paper in IEEE Computer, or use this link to see some source code: https://www.tidalscale.com/how_to_use_large_memory

“This is the way all servers will be built in the future.”

Gordon Bell

Industry legend & 1st outside investor in TidalScale

TidalScale

TidalScale