

whoa hola a todos por ahí y

02:09

bienvenido soy Vicki Hanson la presidenta

02:13

de ACM y estoy realmente encantado de estar

02:15

aquí esta noche

02:17

ACM inició el premio Turing en 1966.

02:21

Reconocer contribuciones de perdurable.

02:24

y gran importancia técnica para el

02:27

campo de computación

02:28

De hecho, somos afortunados de que el 45

02:31

simposio internacional sobre informática

02:33

La arquitectura del foro principal para nuevos

02:36

Ideas y resultados de investigación en informática.

02:38

La arquitectura es el lugar para esto.

02:40

conferencia anual de la ACM como conferencia

02:46

resultado de su fundamental

02:46

contribuciones al desarrollo de

02:49

Procesadores basados en riesgo de eficiencia energética.

02:52

Los dos destinatarios del 2017 Turing.

02:55

premio han jugado un papel clave tanto en el

02:58

Móvil y en el internet de las cosas.

03:01

revoluciones John L Hennessy ex

03:06

presidente de la Universidad de Stanford y

03:08

David a Patterson se retiró presidente I

03:11

Dijo que ante profesor retirado de

03:14

la universidad de california Berkeley

03:16

y el ex presidente de ACM concebido de  
03:20  
utilizando un conjunto de simples y generales  
03:22  
instrucciones que necesitaban menos  
03:24  
transistores y así reducir la cantidad  
03:27  
de trabajo una computadora debe realizar el  
03:29  
El concepto era revolucionario, el trabajo dirigido.  
03:34  
por Hennessy en Stanford y Patterson en  
03:37  
Berkeley daría lugar a una sistemática  
03:39  
enfoque cuantitativo para el diseño  
03:41  
Más rápido reduce la potencia y reduce  
03:45  
instrucciones en microprocesadores  
03:49  
El equipo de Patterson en Berkeley acuñó el  
03:51  
riesgo a largo plazo y construyó el riesgo uno  
03:53  
procesador en 1982 sería más tarde  
03:58  
comercializado por Sun Microsystems en  
04:00  
su microarquitectura chispa Hennessy  
04:04  
cofundó los sistemas informáticos MIPS en 1984  
04:08  
Para comercializar el trabajo de los equipos de Stanford.  
04:12  
durante el último cuarto de siglo su  
04:14  
arquitectura de computadora de un libro de texto  
04:17  
enfoque cuantitativo que ahora está en  
04:19  
Su sexta edición ha influido.  
04:22  
generaciones de ingenieros y computadores  
04:24  
diseñadores que adoptarían y seguirían  
04:27  
refinar sus ideas se convertiría en el  
04:30

base para nuestra capacidad de modelar y

04:32

Analizar las arquitecturas de los nuevos.

04:34

procesadores acelerando así los avances en

04:38

diseño de microprocesador hoy 99% de la

04:44

más de 16 mil millones

04:45

microprocesadores producidos anualmente son

04:48

Los procesadores RISC se encuentran en casi

04:52

todos los teléfonos inteligentes tabletas y en el

04:55

miles de millones de dispositivos integrados que

04:57

Comprende el Internet de las cosas, es mi

05:01

privilegio de presentar nuestro 2017 Turing

05:04

el presidente John Hennessey de los laureados

05:08

Junta del alfabeto Inc y director de

05:10

El programa de los Caballeros Hennessy Scholars en

05:12

Universidad de Stanford y Dave Patterson

05:16

distinguido ingeniero en Google y

05:18

vicepresidente de la junta directiva del riesgo cinco

05:21

Fundación John y Dave bienvenidos

05:38

bien así que cuando John y yo estábamos pensando

05:44

cómo hacer esto parecía como

05:46

Sería una locura si damos dos independientes.

05:49

conferencias dadas nuestra carrera, así que vamos a

05:51

haz esto como equipo de etiqueta, veamos si, de acuerdo

06:00

de esta manera uno pensaría que podría con Gotta

06:06

disparador oh no está bien bien

06:10

así que la primera parte será historia

06:13

vamos a hacer la historia de la computadora

06:15

La arquitectura en unos 20 minutos estoy

06:16

va a hacer esa parte que John va a hacer

06:18

parte 2 hablando de los retos

06:19

Frente a nuestro campo y luego para la parte 3

06:22

Vamos a juntarlo en equipo haciendo el

06:24

primera parte en dominio específico

06:25

Arquitecturas y estaremos haciendo la última parte.

06:27

y luego hay tiempo para preguntas y

06:29

respuestas y y estamos deseando

06:31

que bien volvamos unos 50 años

06:34

A principios de la década de 1960 IBM tenía este problema que he

06:38

Incluso tenía cuatro líneas incompatibles de

06:40

computadoras y una incompatible la

06:42

los conjuntos de instrucciones son diferentes

06:43

pilas de software son diferentes las

06:45

sistemas operativos diferentes la I / O es

06:47

Diferentes y los mercados son diferentes.

06:48

así que los ingenieros de IBM tuvieron la idea de que

06:53

Apostaría la empresa a apostar el conjunto.

06:55

Empresa que podrían tener una sola.

06:57

conjunto de instrucciones que manejaría todo

07:00

Cuatro líneas independientes que unificarían.

07:02

alrededor de eso para lograr eso y luego

07:06

Ahora la parte difícil es el control, pero Maurice.

07:09

Wilks la segunda persona en ganar el

07:11

Premio Turing y de nuestro desde nuestro campo.

07:15

Tenía esta idea de cómo hacerlo más fácil.

07:17

Para construir el control y su idea fue llamada

07:19

microprogramación de la visión que la lógica

07:22

Es lo que y su tecnología en ese momento.

07:25

Era más caro que ROM o Ram pero

07:28

ROM fue más barato que la memoria RAM y fue más rápido

07:31

así que vamos a especificar el control como de sólo lectura

07:35

memoria y llamaron a cada palabra de ese

07:37

control de microinstrucción de memoria para IBM

07:40

iba a apostar a la empresa que eran

07:42

va a lograr esto usando micro

07:44

programando así que en abril de 1964 hicieron

07:47

el mayor anuncio en el

07:49

La historia de la empresa y aquí hay un ejemplo de

07:51

cuatro

07:51

Diferentes máquinas se pueden ver los datos.

07:53

El ancho va desde 8 bits a 64 bits y

07:56

Las microinstrucciones van desde 50 bits.

07:58

a 87 por lo que en aquel entonces con el código micro

08:00

cuanto más amplio es su trabajo duro, más amplio

08:02

La microinstrucción, pero no tomó

08:04

tantas instrucciones micro para interpretar

08:06

así se acortó por lo que era una especie de

08:08

llamada microprogramación horizontal y

08:09

el de la izquierda es de 50 bits pero es

08:12

torta más larga toma más ciclos de reloj

08:14

para ejecutar así que eso es lo que se llama

08:15

Programación vertical menor y apuestan.

08:17

la empresa y en ese entonces usando hoy

08:19

dólares un millón y medio de dólares para

08:21

El pequeño 8 ms el grande y ellos.

08:24

Ganó la apuesta, la compañía ganó la oferta.

08:25

por lo que IBM dominó la computación de mainframe

08:29

industria y todavía hoy en día que

08:30

conjunto de instrucciones todavía está disponible y

08:32

domina los mainframes y son los

08:35

segundo arquitecto de computadoras para obtener una

08:37

Premio Turing fred brooks jugó un gran

08:39

Rol en ese esfuerzo viene la ley morris.

08:46

a lo largo de la tecnología de semiconductores viene

08:48

a lo largo y en el área de computación midi ahora

08:50

lógica RAM y ROM son todas iguales

08:52

transistores por lo que son todos acerca de la

08:54

Lo mismo y Rams aman la velocidad a la misma velocidad.

08:56

Pero con la ley de Moore podríamos tener

08:58

grandes tiendas de control y también porque

09:01

fue RAM lo hizo

09:03

Podrías tener más grande porque el control.

09:05

tiendas porque podría arreglarlo por lo que este

09:06

llevado a estos más complicados

09:07

Conjuntos de instrucciones y el ejemplo clásico.

09:10

Eso fue equipos digitales de fax.

09:12

conjunto de instrucciones y se puede ver que tenía

09:14

96 bits de ancho pero 5000 micro instrucciones

09:18

una idea surgió porque estaba en la memoria RAM

09:22

se llama almacén de control de escritura

09:23

Básicamente se podría entonces ya que era

09:27

alterable en lugar de simplemente ejecutar el

09:29

conjunto de instrucciones estándar que podría poner

09:30

ajustes en adaptarlo exactamente a

09:33

tu aplicación para un control escribible

09:34

almacenar

09:35

así que la micro programación se hizo muy popular

09:37

en la academia y fue entonces cuando era un

09:38

estudiante graduado mi tesis doctoral fue en

09:42

esta área y mi primer trabajo fue en cig

09:44

micro pero sig micro es en realidad puede

09:47

No sé, esto es lo internacional.

09:49

taller sobre microprogramación después

09:51

cambió su objetivo pero era

09:53

microprogramación comenzando seguramente la

09:55

La máquina más famosa con grabable

09:58

tienda de control fue el Xerox Alto  
10:00  
construido en 1973, este fue el generador Pro  
10:04  
de las ideas que todos tenemos  
10:05  
Hoy tuvo la primera computadora con una  
10:07  
interfaz gráfica de usuario que la primera  
10:09  
computadora era una Ethernet y esto es todo  
10:11  
Escrito en microcódigo en el escribible.  
10:13  
historia de control podría hacer poco Blit la  
10:14  
Gráficos en la combinación de ethernet.  
10:16  
controlador y la persona hay una  
10:17  
Imagen del alto y del tercero.  
10:19  
arquitectura de la computadora ganar el Turing  
10:21  
Premio Chuck Thacker y este su tenedor  
10:24  
contribuciones con el alto ahora  
10:27  
Los microprocesadores estaban detrás de los tiempos.  
10:31  
Ellos microprocesadores seguirían lo que  
10:33  
los grandes lo hicieron en la tecnología perdida  
10:37  
estaba creciendo rápidamente y lo harían  
10:39  
imitan lo que los grandes harían ellos  
10:41  
hacer estas guerras microprocesador porque  
10:43  
todo el mundo estaba escribiendo y aún en el  
10:44  
semi idioma que dirían aquí es mi nuevo  
10:46  
Instrucción y mírame usar esto y  
10:48  
serían contrarrestados de ida y vuelta  
10:50  
de personas inventando nuevas instrucciones  
10:52



Seguramente el microprocesador más ambicioso.

10:56

tal vez de todos los tiempos pero sin duda en el

10:57

1970 fue el IEP 432 Gordon Moore

11:02

de la Ley de Moore Fama que fue un visionario

11:05

y creía que tenían el 8080 de 8 bits

11:08

microprocesador creyó el siguiente

11:10

Hicieron que el adulto iba a estar estancado

11:13

con una eternidad que duraría tanto tiempo

11:15

como Intel, pero por desgracia por lo que contrató a un montón de

11:18

Los doctores en ciencias de la computación envían

11:20

Oregon para inventar la próxima gran

11:22

conjunto de instrucciones y esto fue muy

11:23

proyecto ambicioso que hicieron y esa era

11:27

Capacidades de 32 bits era un objeto.

11:29

La arquitectura orientada tenía una costumbre.

11:32

sistema operativo escrito en un exótico

11:34

lenguaje de programación tan grandes ideas por desgracia

11:38

Eran grandes pero eran eran eran

11:40

tarde, así que no cabía en un chip que era

11:43

repartidas en unas pocas fichas que tenía

11:46

problemas de rendimiento y iba a ser

11:47

Años tarde, así que la gente de Oregon tenía

11:49

decirle a Gordon Moore lo siento, no estamos

11:52

va a ser hecho en el tiempo, así que lo que era Gordon

11:54

Tenía que hacer lo que tenía que hacer es comenzar una

11:56

proyecto de emergencia se llamaba el 8086

11:59

tenían 52 semanas el equipo tenía 52 semanas

12:03

para crear un stop back stopgap de 16 bits

12:06

procesador para conjunto de instrucciones

12:08

chip de arquitectura todo en 52 semanas

12:10

tomaron tres semanas de tiempo transcurrido

12:13

diez semanas para diseñar la

12:15

conjunto de instrucciones bien y esto

12:18

básicamente

12:19

El 8080 de 16 bits y se anunció.

12:22

no demasiado fanfarria

12:25

La gran noticia gratuita de inteligencia es la de IBM.

12:28

decidió elegir la versión de 8 bits de 8 bits

12:32

versión del busto de 1886 les gustó el

12:34

68000 tenía un conjunto de instrucciones más elegante

12:37

más cerca de la 360 pero era tan tarde

12:39

se fueron con el 8088

12:41

IBM pensó en el momento que

12:44

Podrían vender de la PC tal vez

12:48

250,000 de ellos en cambio vendieron una

12:51

Cien millones por lo que el 8086 se convirtió en un

12:55

éxito de la noche y gracias a binario

12:58

compatibilidad que IBM había inventado

13:00

antes se compilaba binario

13:02

compatible con software de PC y por lo tanto una

13:05

Futuro muy brillante para el 8086 por lo que ahora  
13:09  
Los investigadores comenzaron a echar un vistazo a  
13:11  
estas máquinas micro codificadas esto es una  
13:13  
Imagen de un cuarto arquitecto informático.  
13:15  
para ganar la naranja en lo que es  
13:18  
sucediendo es esta transición de  
13:20  
programación en lenguaje ensamblador para programar  
13:22  
en lenguajes de alto nivel de alto nivel  
13:23  
Los idiomas eran populares pero no podías  
13:25  
escribir sistemas operativos Unix refutar  
13:27  
que UNIX fue escrito en un alto nivel  
13:29  
idioma para que puedas escribir todo  
13:30  
en un lenguaje de alto nivel por lo que ahora  
13:33  
no era el lenguaje ensamblador  
13:35  
Los programadores hicieron fue la salida de la  
13:36  
compilador que importaba Johncock ins  
13:39  
grupo es IBM que construyó este hardware  
13:42  
este servidor ECL, sino también en particular  
13:45  
tecnología de compilación avanzada dijeron  
13:47  
Tomemos nuestra tecnología de compilación  
13:49  
este conjunto de instrucciones IBM mainframe pero  
13:52  
Solo usa las simples instrucciones que  
13:54  
Cargas y almacenes y el registro.  
13:55  
Registrar lo que pasaría el  
13:57  
el rendimiento fue tres veces más rápido  
13:59

mediante el uso de un subconjunto bien que era una especie de

14:01

un resultado impactante y luego a nuestra

14:04

arquitectos informáticos Joe Lima y Doug

14:06

Clark hizo un estudio de ese VAX.

14:08

Arquitectura que te mostré algunas

14:09

diapositivas antes de lo que encontraron primero

14:11

De todo lo que el número medio de reloj.

14:13

ciclos por instrucción fue de 10 por lo que con

14:16

ese micro intérprete de código tomó una

14:18

promedio de 10 pulgash 10 micro instrucciones

14:20

para poder hacer eso y luego el otro

14:22

Lo que encontraron es el 20% de eso.

14:25

conjunto de instrucciones fue el 60% del código y

14:28

casi nunca fue usado

14:29

así que Wow, ¿por qué estamos haciendo eso

14:32

Vine a la escena a la que me uniría

14:34

Berkeley en 1976 y luego hice una cierta

14:38

un poco extraño como asistente

14:40

profesor hice tres años sabáticos

14:42

más tarde en diciembre porque había hecho mi

14:44

disertación en la programación de Michael y

14:46

Querían ayuda con la micro programación.

14:48

Libros y salí un poco asombrado

14:51

lo difícil que fue sacar a los bichos de la

14:54

fuera del conjunto de instrucciones VEX

14:57

La arquitectura, así que tan pronto como volví  
14:59  
Escribí un papel que dije mira si el  
15:01  
la gente del microprocesador va a seguir  
15:03  
Las tendencias de la mini computadora y  
15:06  
Mainframes y compila mas complicado  
15:07  
conjuntos de instrucciones que van a tener una  
15:10  
problema de micro código y tendríamos que ser  
15:12  
capaz de reparar el microprocesador por lo que  
15:13  
iba a usar la historia de control escribible  
15:15  
sobre eso así que lo que pasó con eso  
15:17  
Papel vuelvo de mi sabático hacer  
15:19  
Todo ese trabajo fue rechazado aunque yo  
15:22  
recuerda espectador está diciendo que esto es una  
15:23  
manera estúpida de construir microprocesadores con  
15:25  
microcódigo reparable por lo que me gusta si  
15:27  
lo van a hacer mas  
15:28  
complicado que necesitas para repararlo pero  
15:31  
Es una forma estúpida de construir.  
15:32  
microprocesador así entonces algo así como bien  
15:34  
¿Por qué estamos haciendo esto, así que este es el  
15:37  
Traducción de hermana riesgo por lo que la SRAM  
15:41  
La lectura de la memoria rápida dentro de la  
15:43  
procesador en lugar de utilizarlo para una  
15:45  
intérprete de código micro que vamos a hacer  
15:48  
Es un caché de uso de get simple.  
15:50

instrucciones para que el contenido de la  
15:52  
la memoria cambiará dependiendo de lo que el  
15:53  
programas ejecutados vamos a mantener la instrucción  
15:56  
sets tan simples que no necesitas un  
15:59  
intérprete que básicamente podría obtener una  
16:01  
implementación de tubería para que pueda pensar  
16:03  
De ellas las más sencillas en mis construcciones.  
16:05  
Simplemente no tan ancho por la forma en que el sisk  
16:09  
Sabemos que los compiladores sólo utilizan como pocos  
16:11  
de este complicado 16 crushin por lo que eres  
16:13  
No perder mucho de todos modos alrededor de eso  
16:16  
Era que las fichas están creciendo debido a  
16:18  
La ley de Moore y nos podríamos sanear.  
16:19  
Rutas de datos de 32 bits y cachés en un solo  
16:22  
chip por lo que hizo este riesgo más  
16:24  
atractivo y luego hubo una  
16:26  
avance en la asignación de registros por una  
16:29  
Greg Chapman Kaitlyn lo siento usando  
16:33  
Gráfico para colorear que lo hizo registrar.  
16:35  
Arquitecturas mucho más eficientes que  
16:37  
en el pasado  
16:39  
así que eso es cuando Berkeley y  
16:41  
Stanford entró en escena que hicimos nuestro  
16:44  
trabajo originalmente con una serie de  
16:46  
Cursos de postgrado para cursos de postgrado.  
16:48

Se investigó la arquitectura que

16:50

Y este fue un riesgo que se mencionó.

16:52

antes dos de los estudiantes de posgrado

16:55

decidió construir un más eficiente

16:57

Versión casi al mismo tiempo que Hennessy

16:59

y sus estudiantes en Stanford construyeron el

17:03

MIPS uno así que todo esto fue hecho sobre

17:04

pepperoni contemporáneo de su sueño

17:08

Sabes que nos gustaría haber tenido esto

17:12

explicación temprana pero eventualmente

17:13

A través de creo que la evaluación de trabajo de Clark

17:16

que los hechos y esa arquitectura nos

17:17

Hablamos de que podemos hacer esta ley de hierro

17:19

factoring en tres cosas y como Dilip

17:23

y nuestro coche que está aquí escribió en el

17:25

papel después de las cosas de la muñeca es mirar esto

17:29

s Keats menos instrucciones tal vez tres

17:31

cursos de muchos días de riesgo pero mucho mas

17:33

ciclos de reloj por lo que hay este tipo de

17:35

factores de ventaja por lo que ahora estoy

17:38

Te voy a mostrar volver a la historia y

17:41

vas a ver un par de cosas

17:42

En primer lugar la gente que mira algo.

17:44

como John y yo pero con mucho más pelo

17:47

y vas a ver estas cosas estas

17:49

pequeñas piezas de plástico de lujo, pero esto

17:52

Es la forma en que solíamos hacer presentaciones.

17:54

para los más jóvenes la gran cosa sobre

17:55

Cuando solíamos hacer estas transparencias.

17:57

recuerdan que empezarías la clase

17:59

Enciende el proyector y enciéndelo.

18:01

Tomar cinco segundos para comenzar hoy nosotros lo

18:03

toma unos treinta minutos como parte de ello

18:06

bien aquí vamos que hicimos este treinta

18:09

Hace cinco años y mucho cambió el

18:12

industria de la computación es una lección sobre

18:14

la ventaja de la simplicidad cuando

18:17

comenzar con algo más simple es probable

18:20

tener mucha más longevidad siempre he

18:22

Tenía seguridad y el poder de las buenas ideas.

18:25

Somos una comunidad tecnológicamente impulsada.

18:27

Teníamos la ley de Moore de nuestro lado sabes

18:29

microprocesadores para el futuro, así que

18:31

totalmente esperado que domine pienso

18:34

tal vez éramos jóvenes y optimistas en

18:38

1982 informáticos estaban hartos

18:40

Con complicados sistemas de código y buggy.

18:42

John Hennessy y David Patterson son

18:44

Los creadores de riesgo o los reducidos.

18:47

conjunto de instrucciones

18:48



La peuta en arquitectura ha dado forma al  
18:50  
riesgo mundial conectado revolucionado  
18:55  
La computación en la década de 1980 pero por el año  
18:57  
2000 la arquitectura fue abandonando  
18:59  
de favor sin embargo, la última década tiene  
19:01  
Revivió un riesgo y demostró lo versátil que es.  
19:03  
Puede ser fichas de riesgo ahora el poder  
19:05  
todo desde nuestros teléfonos inteligentes y  
19:07  
dispositivos conectados a algunos de los más  
19:09  
Supercomputadoras potentes en el mundo.  
19:10  
David Patterson y John Hennessy son  
19:13  
Los destinatarios del 2017.  
19:15  
ACM soy el premio Turing este podría ser  
19:25  
Todo el video, así que estamos parados.  
19:29  
frente de Sibley Arut Perum que es  
19:30  
donde John y yo nos conocimos en la primavera  
19:33  
de 1980 sí en una conferencia donde estamos  
19:36  
ambos hablando de nuestro trabajo, sí en el  
19:38  
tiempo de microprocesadores estaban a punto  
19:40  
para desarrollarse en ordenadores reales de 32 bits  
19:43  
que estaba pendiente los microprocesadores tenían  
19:46  
En gran parte solo seguido con el gran  
19:48  
las empresas de informática habían hecho el software  
19:50  
Habla con el hardware tiene un vocabulario del  
19:52  
sabiduría prevaleciente era tener una muy  
19:54

Amplio y muy rico vocabulario de muchos.

19:57

Diviértete el video completo y de hecho en eso

19:59

tiempo los microprocesadores estaban compitiendo

20:01

en cosas locas como aquí está mi nuevo

20:03

instrucciones para hacer este tipo de cosas y

20:05

Así es como podrías usarlo así fue como

20:07

estaban haciendo la comparación y no

20:08

que diciendo que había un conjunto de normas

20:11

puntos de referencia de mi máquina es más rápido que

20:12

tuyo

20:13

bien con los miles sin plomo son los

20:15

mini computadoras es que en realidad tenían que

20:16

cambio de memoria de solo lectura que es

20:19

Tú te conoces cuando lo fabricas tú.

20:22

Especifique el código de mi y es que no se puede

20:24

cambiarlo tenía tantos errores que tenían

20:26

Para hacerlo alterar un poco la memoria RAM porque

20:28

tenían esto muy complicado

20:30

conjunto de instrucciones que era difícil de conseguir todo

20:31

Los bichos de John y yo pensamos que sería

20:34

mejor tener un vocabulario más simple con

20:36

palabras más simples y eso sería un mejor

20:38

base para los microprocesadores que había sido

20:40

trabajando en la tecnología de compilación por lo que

20:42

dicho bien, ¿qué pasa con los procesadores y como

20:44

Dave empezamos con un estudiante graduado

20:47

clase que era una especie de lluvia de ideas

20:48

clase ambos empezamos con eso y eso

20:50

que creo que nos dio un único

20:53

perspectiva en la que ambos empezamos con

20:54

una sábana completamente limpia la gente estaba

20:56

ignorando lo que eran

20:58

fueron para la implementación de este más complejo

21:01

conjunto de instrucciones y particularmente

21:02

estaban ignorando lo que el rendimiento

21:04

implicaciones fueron completamente y la

21:06

estudiantes de posgrado en Berkeley dijeron que sabes

21:07

esto no tiene mucho sentido por qué por qué

21:10

no puede un compilador cruzar esa brecha por qué lo hace

21:12

El hardware tiene que hacerlo así que había

21:13

un argumento de eficiencia desde el muy

21:15

comenzando estas ideas hizo un montón de

21:18

Sentido para nosotros en Stanford y Berkeley, pero

21:20

Fue muy controvertido John y yo

21:22

Otra forma en que colaboramos es nosotros.

21:24

Participé en debates en conferencias.

21:27

por un par de años donde estaríamos

21:28

en el lado de la muñeca y todos los demás

21:30

Estaríamos en el lado de Syst y lo intentamos

21:32

para convencerlos de la sabiduría de nuestra

21:34

abordar las instrucciones que tienden a  
21:37  
dejarse caer son los más complicados  
21:39  
por ejemplo el VAX tiene una instrucción  
21:42  
lo que hace la evaluación polinomial que es  
21:44  
instrucción que es tan infrecuente  
21:46  
Se utiliza que sus beneficios son prácticamente  
21:49  
inexistente esto en realidad trae a una  
21:52  
del tipo de falacias a ese riesgo  
21:54  
la gente dice bien, toma esta máquina  
21:56  
y sacas algo así pues  
21:58  
Realmente he perdido algo y el  
22:01  
analogía que me gustaría usar es uno de una  
22:03  
corredor que un atleta que está ejecutando un  
22:05  
Raza claramente si reduce y pierde.  
22:07  
algo de peso no has perdido algo  
22:08  
En realidad has ganado algo que ambos  
22:10  
Construimos prototipos de nuestros diseños y nos  
22:13  
Podía ver que estas ventajas eran  
22:15  
claro quiero decir que eran y eran  
22:17  
Estos fueron prototipos académicos contruidos por  
22:19  
estudiantes de posgrado hubo una notable  
22:20  
punto en el tiempo esto fue 1984 que en  
22:23  
Estudiantes graduados de Berkeley o Stanford  
22:25  
estudiantes de posgrado podrían diseñar una  
22:26  
microprocesador que se podría decir  
22:27

mejor que Intel podría diseñar era solo

22:29

como wow eso es increíble

22:30

¿Cómo se compara la máquina RISC en

22:33

términos de velocidad

22:35

um la creencia es por aquellos Reacher

22:38

investigadores en el campo es que el uso

22:40

los mismos recursos que puede obtener algo

22:42

como un factor de dos dos para mejorar

22:44

en el rendimiento de costos en comparación con el

22:46

Diseños tradicionales a pesar de su nombre.

22:48

el riesgo era bueno, ¿cómo es esto este fue el

22:52

video largo así que paramos ahí

22:56

bien así que lo que pasó con lo que nuestro

22:59

colegas en Intel somos capaces de acumular

23:01

Equipos de 500 personas para construir micro.

23:04

Procesadores mucho más que el riesgo de

23:06

La gente podía hacer y tenían gran

23:07

tecnología y tuvieron la idea de

23:09

traduciendo las instrucciones x86 como nosotros

23:11

saber en micro instrucciones energéticas por lo

23:13

Cualquiera de las buenas ideas que arriesgan a las personas.

23:15

si hubieran podido usar y así empezaron

23:19

para dominar y llegar a 350 millones

23:21

fichas al año, que es increíble y no

23:23

Solo dominaba el dekstop pero los servidores.

23:26

también, pero la era post PC mira lo  
23:29  
solo digamos que comencemos desde el iPhone  
23:32  
en 2007 ahora no está comprando fichas de  
23:37  
Intel se está volviendo intelectual  
23:39  
propiedad para integrarlo en el SOC que  
23:41  
te estás diseñando y así estamos  
23:42  
eso es diferente y por supuesto en esto  
23:45  
Mercado que valoran el área y la energía.  
23:47  
tanto como simplemente jugando rendimiento, así que  
23:49  
hecho contratado duro sobre cómo manejarlo  
23:52  
Y de hecho el año pasado hay más.  
23:54  
de 20 mil millones de chips con 32 bits  
23:57  
procesadores en ellos el x86 alcanzó su punto máximo en  
24:00  
2011 con la caída de las ventas de PC y  
24:03  
en realidad están vendiendo menos ahora  
24:05  
De lo que usaban a las nubes aún grandes.  
24:08  
que a medida que este documento estima que hay  
24:10  
Sólo me gustan 10 millones de servidores en el  
24:11  
nube por lo que no es que muchos chips por lo  
24:13  
Es el 99% de los procesadores de hoy o riesgo.  
24:15  
bien, ¿qué sigue en la historia de  
24:18  
La arquitectura informática es algo que  
24:20  
iba a reemplazar insiste en el riesgo que  
24:23  
es el tipo de Eli W el  
24:26  
campeón de eso fue Jost Fischer él  
24:28

En realidad hizo su disertación y micro  
24:30  
Programación y puedes pensar en el  
24:31  
Lawaetz microcódigo horizontal derecha es  
24:33  
instrucciones muy amplias que controlan una  
24:35  
mucha ruta de datos, pero el compilador hace  
24:37  
todo el trabajo ya era hora de que Intel  
24:40  
expandir su arquitectura y ellos  
24:42  
Decidieron llegar a los 64 bits que habían hecho.  
24:45  
La transición del 16 al 32 y ahora como  
24:47  
temporizadores a través de  
24:48  
a 64 y decidieron que estaban en una  
24:51  
cama en VL iw y nombran su  
24:53  
Veterinario épico y de hecho unen fuerzas.  
24:56  
con Hewlett-Packard que también fue  
24:57  
trabajando en VL aw y juntos estaban  
25:00  
haciendo la arquitectura épica que es una  
25:04  
VL iw con compatibilidad binaria y ellos  
25:06  
Comenzó a hacer 1994 a considerable  
25:08  
fanfarria y ahora lo que esto significaba era para  
25:13  
razones de negocios que saben que eran  
25:14  
utilizando este conjunto de instrucciones de emergencia para  
25:16  
Durante 20 años y así se hizo  
25:19  
sentido de tener una marca tecnológica  
25:21  
fundación también hubo un negocio  
25:23  
ventaja es que AMD tenía los derechos de  
25:25

Crea el x86 pero dale el nuevo.

25:28

conjunto de instrucciones que no tenían la

25:30

derechos de que por lo que no estaban permitidos

25:32

por lo que AMD se vio obligado a extender el

25:34

x86 a 64 bits mientras este nuevo

25:36

la arquitectura iba a hacerse cargo de la

25:38

mundo

25:38

y un montón de gente cuando Intel y HP

25:42

fuerzas unidas en los años 90 dicen que este es el

25:44

futuro un montón de empresas simplemente creen

25:46

ellos wow que va a pasar como si o

25:48

No así que simplemente dejaron lo que eran.

25:50

haciendo caer las arquitecturas RISC y

25:51

Comenzó a abrazar lo épico que pasó.

25:55

fallo épico bien por lo que el compilador

26:02

lo que pasó fue que sabes el código que

26:05

podría funcionar bien en código de punto flotante

26:06

Realmente no funcionó en energía o código

26:08

derecho los punteros causaron un problema en

26:10

en particular los problemas fueron el código

26:13

tamaño es V liw derecho por lo que el programa es

26:15

se hizo un poco más grande su largo

26:16

instrucciones ese es el problema

26:18

Las ramas impredecibles eran un problema.

26:20

efectivamente para estos códigos enteros y

26:21



entonces los errores de caché que eran  
26:23  
impredecible por lo que los tres de los hechos  
26:25  
Bueno, dos de los que lo hicieron muy difícil  
26:27  
para compilar y el programa es realmente una  
26:29  
mucho más grande y fuera de servicio  
26:30  
Las técnicas funcionaron bastante bien para el  
26:32  
Layton ve mejor que el tonto  
26:34  
tan poco fuera de servicio lo subsumido y  
26:37  
eso, pero lo más grande fue el  
26:38  
compiladores que la apuesta a VL estaba en l  
26:42  
Creo que los compiladores pueden manejar todo esto  
26:43  
La complejidad del calendario de todos estos brazos anchos.  
26:46  
resultaron como Don Knuth otro  
26:48  
Ganador del premio Turing dijo imposible  
26:50  
Escribe ahora dado toda la publicidad.  
26:52  
Alrededor del Itanium y épica fue.  
26:56  
Llamado cuando empezó a no funcionar.  
26:58  
La gente nota y por lo que algunos menean en lugar de  
27:01  
llamándolo el  
27:02  
Las prisiones son el tánico y así el  
27:07  
ellos y podías ver el hundimiento en  
27:10  
el futuro, así que eso es algo de lo que hacemos  
27:12  
En arquitectura de computadores tenemos derecho  
27:14  
Estos argumentos y luego las empresas van  
27:16  
gastar miles de millones de dólares apostando en  
27:17

Ambos lados y luego dejamos que el mercado

27:18

Los técnicos lo descubren y en este caso

27:21

Fracasé, así que terminé mi parte antes de que yo

27:24

entregar a mi colega el consenso

27:26

conjuntos de instrucciones de hoy no es sis sin

27:28

uno propuso uno de estos micro código

27:29

la instrucción del intérprete pone en más

27:31

Hace 30 años que no trabajaba para Debbie.

27:34

propósito general por algunas de las razones

27:35

decimos sin embargo usted sabe que se ha encontrado una

27:37

Lugar y más cosas incrustadas o DSP

27:40

Porque sabes que es que las ramas.

27:43

Son más fáciles que no tiene cachés las

27:45

Los programas son más pequeños tipo de inicio de ser

27:47

Liw trabajó allí no funcionó para el general

27:49

propósito así que lo que queda

27:49

arriesgarse a quien hubiera adivinado 35 años

27:52

más tarde sigue siendo las mejores ideas bien

27:54

con eso etiquetaré a John y él lo hará

27:57

tomar el control

27:59

[Aplausos]

28:04

bien y ahora para algo realmente

28:06

diferente bien, así que lo que vamos a hablar

28:12

sobre ahora es lo que algunos de los actuales

28:13

Los desafíos son que sé que la mayoría de ustedes son

28:15

familiarizado con ese tipo de cosas

28:17

Cambios tecnológicos estamos en una era somos

28:20

de un montón de cambios en este momento al final de

28:22

Escalamiento Dennard significa que el poder y

28:24

La energía se convierte en las restricciones clave de diseño

28:26

y el final de la Ley de Moore no es el

28:29

El final completo de todo como Gordon Moore dijo que

28:32

Todos los exponenciales han llegado a su fin.

28:34

derecho, así que estamos en la fase de desaceleración en

28:36

La ley de Moore, pero también nos enfrentamos a

28:39

tipos similares de desafíos en torno a nuestra

28:41

ideas arquitectónicas porque nuestra

28:43

Ideas arquitectónicas como las empujamos siempre

28:46

Más difícil se volvió cada vez menos eficiente.

28:49

si eran ideas sobre multi-core

28:52

y la ley de Amdahl o si había

28:54

Conceptos para explotar la instrucción.

28:56

paralelismo nivel estábamos empujando el

28:58

sobre cada vez más y como el

29:01

ineficiencias en los fundamentales

29:03

Las ideas arquitectónicas se hicieron más grandes y

29:05

más grande el hecho de que estábamos al final

29:07

de la escalada de Dennard y el final de

29:10

La ley de Moore lo hizo cada vez más.

29:11

difícil, así que lo que ha pasado en términos de

29:15

procesadores bien tenemos esto desde el principio  
29:19  
noción que estábamos llegando temprano sis  
29:21  
procesadores de un 22 por ciento  
29:23  
rendimiento por año, entonces tenemos en este  
29:26  
increíble fase alta donde estábamos  
29:29  
conseguir metas de rendimiento dramáticas 50  
29:31  
porcentaje de mejora por año, entonces nosotros  
29:34  
Algo así como el ILP.  
29:36  
Ideas al final de la escalada de Dennard llegó  
29:39  
a lo largo nos movimos a multi-core funcionó  
29:41  
bastante bien y luego las cosas se pusieron a punto  
29:45  
Más lento y por fin nos fijamos en el último.  
29:48  
pocos años estamos básicamente mirando  
29:50  
Tres por ciento de mejora en el procesador.  
29:53  
rendimiento por año al final de una  
29:55  
fase dramática tenemos que repensar lo  
29:57  
Lo estamos haciendo tenemos que abordar los problemas.  
29:59  
diferente si se rompe esto y  
30:01  
Empieza a mirar las cosas que puedes ver.  
30:03  
facturación en la ley de Moore con respeto  
30:05  
a D Rams por supuesto D Rams son una muy  
30:07  
tecnología particular en la que se basan  
30:09  
Diseño de condensador de zanja y por lo que es  
30:13  
básicamente visto la cola más rápido que  
30:16  
convencional pero incluso si  
30:17

nos fijamos en el número de procesadores del

30:20

número de fichas el número de

30:21

transistores en un procesador Intel que

30:24

Puede comenzar a ver el fin de la ley de Moore.

30:26

Justo allí un poco y luego

30:27

Volvimos a juntar vapor pero si miras

30:29

en esa curva desde aproximadamente 2000 estamos

30:33

cayendo así que estamos ahora si nos hubiéramos quedado

30:36

en esa curva tendríamos 10 veces más

30:38

Transistores en un microprocesador típico.

30:40

Tenemos hoy, así que realmente

30:42

diferenciado y separado de

30:44

esa curva que nos ha hecho a todos

30:46

Piensa cómo vamos a hacer a continuación.

30:48

Y, por supuesto, la escala de Dennard es similar.

30:51

tipo de problema aún más agudo creo

30:53

todos dirían todos los que lo hacen

30:55

diseño de chips diría que la energía es un trabajo

30:57

ahora enciende el trabajo uno para pensar en el

31:00

Diseño para que las tecnologías continúen.

31:02

para mejorar tenemos esta curva que es

31:05

El consumo de energía que va en el

31:07

otra dirección y nos fijamos en cómo eso

31:09

curva realmente despegó después de aproximadamente 2008

31:12

simplemente sube y sube y sube ahora de

31:15

Por supuesto que eso es poder

31:16

Cómo se traduce en energía depende de

31:20

¿Qué tan eficientes estamos usando el

31:23

transistores lamentablemente las técnicas

31:26

tenemos para usar transistores tenemos

31:28

pensar cada vez más ineficiente

31:31

sobre caches algo que todos amamos uno

31:33

De las verdaderas grandes ideas en informática.

31:35

derecho, pero por supuesto, el más grande y

31:37

más grande haces tu caché menos y

31:39

Menos efectivo es y acelerando el

31:41

programa por lo que es nuestro reto que hemos

31:43

Tengo que encontrar nuevas formas de pensar acerca de cómo

31:45

Para utilizar la capacidad que tenemos más.

31:47

eficientemente también estamos en un muy triste

31:50

Estado con respecto a la seguridad como nosotros

31:52

escuchado en el panel a la hora del almuerzo la

31:55

Lo simple que diría acerca de la seguridad es

31:56

Si los aviones no funcionan tan a menudo como

32:00

mal funcionamiento de las computadoras nadie estaría en

32:02

esta conferencia que no vivió en

32:03

El sur de California aquí todos estar en casa

32:06

porque nunca te subirías a un avión bien

32:08

Si los coches no funcionaran así, nunca lo haríamos.

32:11

entrar en nuestro coche bien tenemos una gran

32:13

problema ahora algunos de nosotros somos lo suficientemente viejos para

32:16

recuerda cuando había mucha

32:18

El énfasis en la seguridad en los años 70 o grandes.

32:22

proyectos que era multics había una

32:23

mucho enfoque en él inventamos dominios

32:26

suenan incluso capacidades algunas de las

32:28

Ideas que recién están regresando a

32:30

arquitectura de Computadores

32:31

Se pilotaron en la década de 1970 lo que sucedió.

32:35

esas ideas no fueron bien utilizadas también nosotros

32:38

Aún no se había desarrollado la arquitectura

32:40

Técnicas para hacerlos rápidos para que las cosas.

32:42

como la traducción del interruptor de búfer lookaside

32:45

hacer posible la memoria virtual

32:47

memoria virtual sin que todos y cada uno

32:49

acceso a la memoria cada acceso a la memoria solo

32:51

Requiere dos accesos a memoria principal por lo que

32:54

no teníamos formas de alto rendimiento para

32:56

hazlo las técnicas no parecían ayudar

32:59

tenían un montón de gastos generales eran

33:01

abandonado al mismo tiempo que pensamos

33:04

la verificación formal va a resolver

33:06

todos nuestros problemas

33:07

Vamos a verificar todo nuestro software en

33:10

De hecho recordé el surgimiento del núcleo

33:12

Sistemas operativos basados y microkernel.

33:14

el núcleo del sistema operativo que

33:16

cosas de seguridad controlada era solo

33:18

Va a ser un mil o 2.000 líneas de

33:20

código

33:20

no hay núcleo con menos de un

33:22

millones de líneas de código por ahí, así que

33:24

Básicamente no entendí la seguridad

33:27

Justo como pensábamos que íbamos

33:28

Para resolverlo casi todo el software tiene errores.

33:31

todos ustedes que compran una nueva pieza de

33:33

software obtener una divulgación de 15 páginas que

33:36

Básicamente dice que si este software no lo hace.

33:38

trabajar demasiado mal para ti justo eso es lo

33:41

dice y todos ustedes marcan la casilla que

33:43

Puede obtener el software por lo que el hardware

33:47

comunidad la comunidad de arquitectura tiene

33:49

para intensificar el trabajo de la mano con la

33:51

Arte con la comunidad del sistema operativo.

33:54

Con las personas que piensan en seguridad.

33:55

para realmente solucionar este problema si

33:58

no lo hagamos vamos a tener una comunidad

34:00

de los usuarios por ahí van a convertirse

34:02

Cada vez más infeliz con nosotros hay

34:07

un montón de agujeros en esto si nos fijamos en

34:08



Aquí hay un solo ejemplo simple Intel

34:12

Los procesadores tienen este motor de gestión.

34:14

que ejecuta un montón de código antes de cualquier

34:18

corre el código del sistema operativo del núcleo y

34:21

configura la máquina que lee ese código

34:24

y asegúrate de que está haciendo lo correcto

34:26

cosa que ninguno de nosotros lea ese código por lo que hemos

34:30

Tengo problemas reales con grandes instrucciones

34:32

conjuntos de personas pueden probar códigos de operación aleatorios

34:35

encontrar agujeros en el conjunto de instrucciones

34:36

Definición de un montón de problemas que tenemos

34:39

para hacerlo bien y tenemos que repensar

34:41

Y por supuesto tenemos el espectro.

34:44

arquitectura de Computadores

34:45

como señalamos en el almuerzo esto va a

34:48

Requerimos que reconsideremos nuestra definición de

34:50

conjunto de instrucciones de arquitectura porque nosotros

34:52

Nunca pensé en el tiempo basado ahora estoy

34:55

edad suficiente para recordar cuando se basa el tiempo

34:57

los ataques fueron descubiertos por primera vez en

35:00

la década de 1970 hubo un tiempo basado

35:02

ataque en la parte superior 10 y tops 20 el OSS

35:06

que corría en sistema de cubierta decenas y

35:08

Veinte años atrás y luego sólo tipo

35:10

Se olvidó de ello y asumió que era una

35:12

problema del sistema operativo y puramente una  
35:14  
problema de software ahora es una  
35:16  
problema de arquitectura y es una  
35:17  
problema de arquitectura que ha estado allí  
35:18  
Durante 20 años y realmente ni siquiera  
35:20  
saberlo, así que tenemos que repensar lo que nosotros  
35:23  
significa acerca de la seguridad lo que nuestro papel lo  
35:26  
El papel de la comunidad arquitectónica es.  
35:28  
Cómo trabajamos en colaboración y creo.  
35:31  
Como hemos señalado en el almuerzo hay  
35:33  
Mucho más ataque de microarquitectura en  
35:35  
la forma mucho más tiempo basado  
35:36  
ataques de canal lateral están llegando esto es  
35:39  
no va a ser un problema fácil es  
35:40  
va a significar realmente repensar cómo  
35:42  
pensar en la seguridad bien, así que esto es una  
35:48  
nuevo momento para pensar en este problema y  
35:50  
vamos a tener que redefinir nuestra noción  
35:53  
de la arquitectura de la computadora así que aquí estamos  
35:57  
Suenan como una tragedia a punto de desenrollarse.  
36:00  
desacelerar en la ley de Moore no más Dennard  
36:03  
La seguridad del arquitecto de escala es un desastre  
36:06  
¿Qué vamos a hacer al respecto, así que tengo  
36:09  
Siempre tomada la vista hay una gran  
36:10  
cita que John Gardner dijo una vez que  
36:13

dijo que nos enfrentamos es un conjunto de  
36:15  
desafíos aparentemente imposibles  
36:18  
oportunidades disfrazadas de grandes  
36:20  
desafíos y ahí es donde estamos nosotros  
36:22  
tienen grandes oportunidades así que cuales son  
36:24  
esas oportunidades piensan en software  
36:28  
Maurice Wilkes me dijo hace unos 25 años.  
36:31  
Hace ya dije Maurice lo que pasa si  
36:33  
La ley de Moore siempre se ralentiza y él dice  
36:36  
entonces vamos a tener que pensar mucho  
36:37  
más cuidadosamente sobre cómo escribimos nuestra  
36:39  
Software y prestar mucha más atención a  
36:40  
eficiencia, así que tenemos software centrado  
36:43  
oportunidades correctas que todos escribimos en  
36:45  
estos lenguajes de scripting modernos son  
36:47  
interpretados son tipificados dinámicamente  
36:49  
fomentan la reutilización dan mucha  
36:51  
poder pero corren terriblemente son  
36:55  
increíblemente ineficiente ideal para  
36:57  
programadores malos para salir  
36:59  
Enfoque centralizado de hardware I  
37:01  
Creo que Dave y yo creemos que el único  
37:04  
camino a seguir y creo que mucha gente  
37:06  
en la comunidad de arquitectura toma esto  
37:07  
ver es algo que es más  
37:09

dominio específico, así que no trato de construir  
37:11  
un proceso de propósito general que hace  
37:13  
todo bien construir un procesador que  
37:15  
hace algunas tareas increíblemente bien y  
37:18  
averiguar cómo construir un heterogéneo  
37:20  
arquitectura utilizando esas técnicas y  
37:23  
Por supuesto, hay combinaciones de hardware  
37:25  
y el software van juntos tenemos que  
37:27  
piensa no solo en el dominio  
37:28  
Arquitecturas pero los lenguajes que son.  
37:30  
solía programarlos al mismo tiempo para  
37:33  
Aquí hay un gran gráfico que está fuera de una  
37:36  
papel llamado hay un montón de espacio en  
37:37  
la parte superior por la concesión de licencias y un grupo de  
37:40  
colegas en el MIT y se ve a una  
37:43  
un simple ejemplo es cierto que un simple  
37:44  
Ejemplo de matriz multiplicar por lo que vamos a tomar  
37:47  
La versión en Python cuanto más rápido  
37:50  
¿La versión acaba de reescribirse en C run?  
37:53  
cuarenta y siete veces más rápido 47 veces ahora  
37:57  
Estoy trabajando en compiladores antes de trabajar  
37:59  
en arquitectura de computadores un factor de dos  
38:02  
te haría una estrella en el compilador  
38:05  
comunidad si acaba de obtener la mitad que 47  
38:08  
Dejaste un factor de 23 serás un héroe.  
38:11

Ganarás el premio Turing entonces y luego

38:18

lo tomas y lo llevas a un 18

38:21

Intel Core a encontrar el bucle paralelo

38:23

Porque no hay forma de que nuestro software.

38:25

Los sistemas pueden encontrarlos automáticamente que

38:27

te da otro factor de ocho entonces

38:30

te llevas pon la memoria

38:31

optimización para que los cachés realmente funcionan

38:33

correctamente con una gran matriz que

38:36

típicamente no te da otra

38:39

Factor de 20 y finalmente tomas.

38:42

ventaja de hardware específico del dominio

38:44

Con las instrucciones de Intel AVX usas el

38:47

vectores y eso te da otra

38:49

factor de 10 mejora el rendimiento del

38:51

versión final 62 mil veces más rápido

38:55

que la versión inicial que es una gran cantidad de

38:59

trabajar para sacar eso de la Ley de Moore y

39:01

hardware, así que esto es sólo un gran

39:03

oportunidad de dominio específico

39:05

Las arquitecturas logran mayor eficiencia.

39:08

diciendo la arquitectura a la

39:09

características del dominio

39:11

y por dominio específicamente significa

39:13

Incluir una infraestructura programable.

39:17

derecho una máquina que hace un procesador

39:19

arquitectura que hace un rango de

39:21

aplicaciones que se caracterizan en una

39:24

dominio particular no solo un cableado

39:26

Redes neuronales de aplicación para máquina.

39:30

GPUs de aprendizaje para gráficos programables.

39:32

Los conmutadores de red son todos buenos ejemplos.

39:34

así que si sabes que Dave y yo sabemos que lo haríamos

39:37

me gusta ver una explicación cuantitativa

39:40

de por qué estas técnicas son más rápidas y yo

39:43

creo que eso es importante porque no lo es

39:45

La magia negra aquí es efectiva.

39:48

porque hacen un uso más efectivo de

39:50

paralelismo SIMD menos flexible que mi

39:54

MD pero cuando funciona más eficiente no

39:57

duda al respecto

39:57

derecho VL iw contra Adán especulativa

40:01

Orden más eficiente cuando BL iw

40:04

estructuras de trabajo más eficaz uso de

40:07

cachés de ancho de banda de memoria son grandes, excepto

40:09

cuando tampoco funcionan bien porque

40:12

de localidad espacial localidad temporal

40:14

Ventajas de streaming que el programador

40:17

puede tomar porque entienden la

40:19

patrones de acceso a la memoria eliminan

40:22

precisión innecesaria a la derecha reemplaza l Triple

40:24

E por punto flotante de menor precisión o

40:26

Enteros de 32 y 64 bits por 16 y 8 bits

40:30

enteros y luego programarlos usando una

40:33

lenguaje específico de dominio que puede obtener

40:35

mayores niveles de eficiencia para que

40:40

sobre estos lenguajes específicos de dominio l

40:42

Creo que la clave es que es demasiado

40:46

difícil de empezar con C o Python y

40:49

extrae el nivel de conocimiento que necesitas

40:52

para asignar al hardware de manera eficiente si

40:56

Usted tiene soporte específico de dominio en él

40:59

es simplemente un problema demasiado difícil un montón de

41:01

Nosotros y la comunidad de compiladores trabajamos en

41:03

ese problema es demasiado difícil para que

41:05

Necesita un lenguaje de nivel superior que habla

41:07

sobre matrices de vectores u otros

41:10

Estructuras de alto nivel y especifica.

41:13

operaciones en ese nivel más alto todavía

41:16

significa que hay compilador interesante

41:18

retos porque queremos una

41:20

programa específico de dominio para estar bien

41:25

Independiente de la arquitectura por lo que

41:28

quiero tener compilador interesante

41:30

Los desafíos son de mapeo de mapas

41:32

ese programa específico de dominio a una  
41:34  
arquitectura particular y la  
41:36  
La arquitectura puede variar de una versión a otra.  
41:38  
versión, así que creo que hay un montón de  
41:42  
excelentes oportunidades de investigación aquí  
41:44  
Haz programas de Python o, a diferencia de C,  
41:47  
ser un héroe si haces eso es una especie de  
41:50  
deja vu cuando lo pienso porque  
41:52  
Lo que estábamos tratando de hacer en la muñeca.  
41:53  
días era hacer lenguajes de alto nivel  
41:55  
correr eficientemente en el  
41:58  
arquitecturas de la época esta es la  
42:00  
mismo desafío mismo tipo de desafío  
42:03  
aplicaciones específicas de dominio lo que son  
42:06  
los objetivos correctos cuáles son los correctos  
42:07  
idiomas como piensas  
42:09  
tecnología de compilación ¿cómo construir  
42:12  
lenguajes específicos de dominio y  
42:14  
aplicaciones que pueden portar desde uno  
42:17  
generación a la siguiente, así que no estamos  
42:18  
software de reescritura siempre correcto  
42:21  
bueno, ¿en qué problema podrías trabajar bien?  
42:24  
hay un área donde el número de  
42:26  
Los papeles crecen tan rápido como la ley de Moore.  
42:29  
que se puede ver en esta trama y  
42:31



eso es aprendizaje de máquina por lo que hay una  
42:33  
área obvia es computacional  
42:35  
intensivo proporciona un montón de interesantes  
42:38  
cosas y el número de aplicaciones es  
42:40  
creciendo a pasos agigantados ¿por qué no  
42:42  
Trabajar en el aprendizaje automático, por supuesto  
42:44  
que la unidad de procesamiento de tensor es uno  
42:47  
ejemplo de esto que Google trató de hacer  
42:49  
Centrarse en la inferencia en redes neuronales.  
42:52  
Implementar estas máquinas y creo que desde  
42:55  
El punto de vista de la gente de Google su  
42:57  
vista es si no desplegamos esto  
42:59  
Tecnología que no podremos permitirnos  
43:02  
ejecutar muchas aplicaciones en la máquina  
43:05  
aprendiendo porque serán demasiado  
43:06  
costoso  
43:06  
Serán computacionalmente demasiado caros  
43:08  
Las arquitecturas se ven radicalmente.  
43:11  
derecho diferente por lo que en lugar de construir  
43:14  
grandes cachés y ocupan un montón de la  
43:16  
arquitectura para el control en lugar de eso  
43:19  
construir memoria que está dirigida a la  
43:22  
uso de la aplicación porque el acceso a la memoria es  
43:24  
un gran consumidor de energía que crees muy  
43:27  
Es difícil tratar de obtener acceso a la memoria  
43:29

en chip en lugar de en chip piensas

43:33

sobre la construcción de un montón de computacional

43:35

ancho de banda para que coincida con el tipo de

43:37

aplicación que estás ejecutando

43:38

En el caso de hacer red neuronal.

43:40

inferencia está haciendo mucha matriz

43:43

Se multiplica correctamente para que construyas una sistólica.

43:46

matriz aquí hay una vieja idea volver en

43:50

Es primordial 30 años después de su creación.

43:55

Y si nos fijamos en el rendimiento puedes

43:57

obtener mejoras dramáticas en el rendimiento

44:01

términos de rendimiento por vatio por supuesto

44:03

El reto uno de los retos.

44:05

Lo que tendremos aquí es qué aplicaciones.

44:08

¿Qué puntos de referencia y tal como lo hicimos en

44:11

época anterior inventamos espec. como una manera de

44:13

para normalizar y obtener maneras de comparar

44:17

diferentes arquitecturas que vamos a tener

44:18

pensar en como lo hacemos para

44:20

aprendizaje automático y otro dominio

44:22

ambientes así la única cosa que hemos

44:25

aprendido en la comunidad de arquitectura si

44:26

tenemos un conjunto de puntos de referencia que es una

44:28

buen conjunto de puntos de referencia y es

44:29

razonable y no puede ser manipulado

44:31

que proporciona estímulo para que todos  
44:34  
pensar en cómo sacar a la luz sus ideas  
44:36  
y ponlos a prueba contra eso  
44:39  
así que resumen un montón de oportunidades, pero una  
44:45  
nuevo enfoque a la arquitectura de la computadora es  
44:47  
Necesitamos necesitamos computadora renacentista  
44:49  
Arquitectura en lugar de tener gente.  
44:52  
que solo entienden una pequeña astilla de  
44:54  
la pila vertical tenemos que pensar  
44:58  
acerca de cómo construir equipos que ponen  
45:00  
juntos gente que entiende  
45:01  
aplicaciones personas que entienden  
45:04  
Idiomas y lenguajes específicos de dominio  
45:07  
y la tecnología de compilación relacionada juntos  
45:09  
con personas que entienden la arquitectura  
45:11  
y la implementación subyacente  
45:13  
La arquitectura para mí es un regreso a la  
45:16  
pasado es un regreso a un momento en que  
45:18  
las empresas de informática eran verticalmente  
45:20  
integrado en lugar de horizontal  
45:23  
descompuesto y creo que proporciona una  
45:26  
oportunidad emocionante para las personas tanto en  
45:28  
informática académica, así como en  
45:32  
industria así que gracias por su atención  
45:34  
gracias a todos por organizar esto  
45:36

Conferencia fabulosa y gracias a todos.  
45:39  
Los colegas que han colaborado.  
45:41  
Con esto a lo largo de los años y  
45:45  
dave va a terminar  
45:55  
Sí, me olvidé de agradecer a todos fue un  
45:58  
John Hill no lo hizo de mí, pero sí encontramos  
46:00  
fuera sobre el premio no hace mucho tiempo en  
46:02  
ACM Pat Ryan contactó a Tim Pinkston  
46:08  
y el comité organizador y dijo  
46:09  
¿Qué crees que puedes apretar un  
46:12  
Premio Turing en la conferencia y sobre qué  
46:14  
la demasiada lechuga y lo hicieron y así  
46:16  
Realmente lo apreciamos bien, así que esto es  
46:19  
la última parte antes de la pregunta y  
46:20  
responde y sabes si John y yo somos  
46:22  
Esperando por él, así que siempre  
46:24  
siempre celoso de mis colegas en  
46:26  
compiladores de sistemas operativos  
46:30  
sistemas operativos y compiladores de acuerdo  
46:33  
que podrían trabajar en  
46:35  
cosas de fuerza industrial hacen  
46:37  
contribuciones y todo el mundo podría  
46:38  
Úsalos porque la gente usa código abierto.  
46:40  
sistemas operativos así que ¿por qué no podemos hacer  
46:42  
que en arquitectura está bien, así que déjame decirte  
46:45

Usted acerca de los riesgos 5 y se llama riesgo 5

46:47

en la V porque es una especie de la quinta

46:49

proyecto de risc de berkeley por lo que básicamente

46:53

Era el momento en Berkeley Christa Tsarevich

46:56

estaba liderando el camino y teníamos que hacer

46:58

un conjunto de instrucciones y el problema con

47:01

Los candidatos obvios es si no solo

47:03

eran complicados nosotros simplemente

47:04

No se les permitiría usarlos Intel

47:05

fue uno de nuestros patrocinadores pero no lo hicieron

47:07

Queremos que usemos el x86 porque era

47:09

controlado por lo que Christa decidió que él

47:14

y los estudiantes de posgrado iban a hacer una

47:16

conjuntos de instrucciones de pizarra limpia vamos a empezar

47:17

Sobre consiguió una idea radical pero lo haría

47:20

Sólo tomar tres meses bien y luego estar

47:23

los estudiantes de posgrado líderes eran Andrew

47:25

Waterman sí, si tú y Andrew están aquí

47:27

así lo vi también si voy a

47:29

Hablar de eso así, pero que bien

47:31

Tomó cuatro años pero construyeron una gran cantidad de

47:34

fichas en ese momento y ayudé a algunos, pero

47:36

Es realmente Christa y Andrew y el

47:38

sospechoso que lo hizo entonces esta cosa extraña

47:41

sucedió sabes si eres un en si

47:43

estas en la academia siempre obtienes

47:44

quejas sobre lo que estas haciendo

47:48

Pero empezamos a recibir quejas sobre nosotros.

47:49

Cambiando internamente los detalles del riesgo.

47:53

cinco instrucciones establecidas desde el otoño hasta la primavera

47:54

bien, así que ustedes no apuntan sobre

47:58

todo por qué te importa cómo si nosotros

48:00

cambiar nuestro conjunto de instrucciones en Berkeley

48:02

Por nuestros cursos y lo que descubrimos.

48:04

en hablar con la gente había esto

48:06

sed de un conjunto de instrucciones abiertas que

48:08

miró a

48:09

grupo de ellos descubrieron riesgo cinco

48:11

y empezaron a usarlo eran

48:13

vamos a usarlo por lo que una vez que

48:14

escuchado y entendido hubo una demanda

48:17

para un conjunto de instrucciones abiertas pensamos

48:18

Bueno, eso es una gran idea, vamos a ayudar a hacer

48:20

eso sucede así que, ¿qué es diferente?

48:22

riesgo cinco es muy simple que está lejos

48:25

más simple que otra instrucción establece el

48:28

manual es de aproximadamente 200 páginas del Andrew

48:30

Escribió la mayoría de esos y entrando en Christa.

48:32

escribió la mayor parte de eso y sabes x86

48:34

es diez veces más que una pizarra limpia

48:37

diseño sabes que es más fácil si empiezas  
48:39  
Veinticinco años después para mirar todo.  
48:40  
Los errores del pasado para los MIPS.  
48:42  
y las arquitecturas SPARC no hacen esos  
48:45  
los errores no tienen el lazo  
48:46  
La microarquitectura en lo que es fácil de hacer.  
48:48  
En retrospectiva, es una instrucción modular.  
48:51  
establece y que hay una base estándar que  
48:54  
todo el mundo tiene que tener para todo esto  
48:56  
el software se ejecuta y luego hay opcional  
48:57  
extensión para que incluyas o no  
48:59  
dependiendo de su aplicación porque nosotros  
49:02  
Sabía que las arquitecturas de dominio específico  
49:04  
iban a ser importantes hay un montón  
49:05  
de espacio akkad para dejar de lado algunos  
49:07  
tipo de arquitecturas pierden eso con  
49:09  
campos de direcciones más grandes y el gran negocio  
49:11  
es que la comunidad diseñó la base y  
49:13  
extensiones estándar se terminan eso es  
49:15  
No voy a cambiar si quieres agregar.  
49:17  
extensiones es un esfuerzo de la comunidad para hacer  
49:20  
que a donde traemos los expertos.  
49:22  
de antemano típicamente lo que sucede en  
49:23  
La arquitectura informática anuncia un nuevo conjunto.  
49:26  
de instrucciones tienen una compañía  
49:27

y luego toda la gente del software dice

49:29

lo que estaba mal con eso tenemos esos

49:30

conversaciones en la delantera y luego es una

49:33

Fundación que lo va a mantener.

49:35

sabes que las universidades pierden atención y

49:38

No estoy emocionado, así que es una base.

49:40

fundación sin fines de lucro que lo ejecutará

49:41

y sabes que va a ser tecnología como

49:43

sistemas operativos compiladores adelantos

49:46

pasar por razones técnicas no no

49:47

Realmente razones de marketing por lo que hay

49:51

en realidad unas pocas instrucciones diferentes

49:52

establecer un 32 bits y un 64 bits y uno incluso

49:54

Para incrustar las extensiones estándar.

49:57

opcionalmente se multiplica y divide incluso

49:59

Instrucciones atómicas simples y dobles.

50:01

instrucciones comprimidas de precisión y

50:04

entonces vector cual es mas sabes

50:06

Elegante que los simons clásicos de simpatía.

50:08

Formato de instrucción simple soportado por

50:11

La fundación por lo que los miembros de la afición

50:14

están creciendo y hacia la derecha, de hecho,

50:15

así que hay más de cien de ellos

50:18

En un par de años Nvidia anunció en

50:22

un taller que iban a

50:23



Su microcontrolador con riesgo 5 por lo  
50:25  
Eso es lo que sé 20 o 40 millones al año  
50:28  
Western Digital anunció un taller  
50:31  
que lo van a poner en discos tan  
50:33  
van a llevar la computación a la  
50:34  
disco y y eso va a ser miles de millones  
50:36  
Por año y luego están en nuestra práctica.  
50:38  
Hablar en Stanford el jueves dos personas  
50:40  
se me acercó desde Chow Hong y en Kia  
50:43  
Y anunciaron que van a arriesgarse.  
50:45  
cinco y van a estar enviando 30  
50:46  
millones al año a partir del próximo año, así  
50:48  
Es que realmente está empezando a captar en  
50:50  
términos de los grupos de normas son  
50:53  
Las extensiones o las piezas de tu red.  
50:55  
gente creo que hay gente aquí  
50:56  
¿Quién ha trabajado en estas piezas pero conseguimos  
50:59  
Sabes que es bonito y lo consigues todo porque  
51:01  
Está abierto tienes todos los expertos.  
51:02  
juntos y ten estas conversaciones  
51:03  
algo así como un comité de normas  
51:05  
antes de abrazarlo en el  
51:07  
conjunto de instrucciones y riesgo cinco es justo  
51:10  
Un ejemplo en Vidya a su crédito.  
51:12  
no ha abierto justo lo que John estaba hablando  
51:15

sobre un acelerador específico de dominio y

51:17

Todo está abierto, la pila de software es

51:19

abrir la arquitectura del conjunto de instrucciones es

51:21

implementaciones abiertas están abiertas es una

51:23

Diseño escalable que puedes usarlo.

51:25

Y viene con el riesgo cinco

51:27

núcleo como anfitrión o no le toca a usted por lo

51:30

ese es otro ejemplo de abierto

51:31

arquitecturas y entonces sabes

51:34

motivados esta mañana son esto

51:36

La hora del almuerzo es seguridad le gusta esto

51:39

idea de arquitecturas abiertas no lo hacen

51:41

crear en la seguridad a través de la oscuridad

51:42

ellos creen en la apertura por lo que son

51:45

empresas preocupadas los países están preocupados

51:47

sobre las trampas es una seria preocupación

51:49

este documento que se hace referencia aquí

51:52

cambiado una línea de registro muy para insertar un

51:54

Trampa que podrían apoderarse de una máquina.

51:56

por lo que les gustaría estar abierto

51:58

implementaciones que se podrían ver

52:00

y luego la gran cosa es que sabes

52:02

claramente de lo que creo que elegirías

52:04

arriba de la conversación del almuerzo

52:05

La seguridad va a ser un gran desafío.

52:07

Para arquitectura de computadores necesitamos  
52:10  
todos los que trabajan que quieren  
52:12  
Ahora mismo con la propiedad nuestra  
52:13  
Criterios de arquitectura en los que tienes que trabajar.  
52:16  
para Intel nuestro brazo para aquellos propietarios  
52:17  
Los que están aquí todos pueden trabajar en ellos y  
52:19  
incluyendo a todos los académicos que tienen mucho  
52:22  
de valor para poder agregar y luego me  
52:24  
Piensa en lo emocionante que es la oportunidad.  
52:27  
Dados los grandes avances en FPGAs y  
52:29  
Implementaciones de código abierto y abierto.  
52:32  
software de pilas que puedes hacer novela  
52:35  
arquitecturas y poner  
52:36  
En línea puedes conectarte a internet  
52:39  
y usted podría ser capaz de someterse a la  
52:41  
ataques o podría ofrecer una recompensa por  
52:43  
ataques por lo que realmente tendría un activo  
52:45  
adversario en el otro lado no solo tu  
52:47  
Sé que se defiende sobre esta cosa e incluso  
52:49  
aunque se ejecuta a 100 megahercios eso es  
52:51  
lo suficientemente rápido para ejecutar software real  
52:53  
Podría tener usuarios y provoca un FPGA.  
52:55  
podrías iterar en semanas en lugar de  
52:56  
años con con hardware estándar  
52:58  
y así que mi conjetura es y hablé con el  
53:02

gente en la mesa una charla como probablemente  
53:03  
muñecas 5 será el ejemplar probablemente  
53:06  
La gente usará 5 si hacemos Co diseño.  
53:08  
Con arquitectos y personal de seguridad.  
53:10  
adelanta que eso probablemente pasó  
53:11  
allí primero ahora otras personas podrían usar  
53:13  
las ideas que probablemente sucedieron primero  
53:15  
y responder de manera resumir lo abierto  
53:18  
arquitectura submarina libre es  
53:20  
absolutamente libre cualquiera puede usarlo  
53:23  
No hay contratos ni nada parecido.  
53:24  
al igual que Linux es más simple de la  
53:28  
puerta y no va a estar allí no lo hará  
53:30  
Ser razones de marketing para expandirlo cuando  
53:32  
hablar con personas o empresas comerciales  
53:33  
¿por qué haces eso bien sabes que es  
53:35  
Más fácil vender una nueva instrucción que  
53:37  
es que sabes una mejor aplicación que  
53:39  
hace una gran diferencia en el extremo inferior  
53:41  
te has sorprendido de cuan minucioso es eso  
53:45  
Quieren lo pequeños que quieren ser.  
53:47  
en el extremo superior sabes el  
53:48  
La arquitectura no importa tanto pero  
53:50  
puede ser que no hay razón por la que no puede ser  
53:52  
tan rápido en la gama alta podemos apoyar  
53:54

Los DSA tenemos el espacio de código de operación para

53:56

Eso y creo que con más gente

54:00

la construcción de procesadores que va a ser una

54:01

mercado más competitivo que probablemente

54:03

significa una innovación más rápida y creo que como yo

54:06

dijo que creo que los expertos en seguridad van

54:08

para reunirse en torno al riesgo 5 y nuestra modesta

54:11

objetivo es la dominación del mundo bien, así que

54:14

No podemos pensar en ninguna razón por la que

54:16

saber un solo conjunto de instrucciones

54:18

La arquitectura no funcionaría bien

54:19

Es pequeño y trabaja con un por qué

54:22

sabes por qué no sabes que eso es todo RISC

54:23

procesadores así que esperando que sea el Linux

54:25

de procesadores por lo que para la última parte de

54:28

la charla bien es hardware ágil

54:33

Desarrollo tan poco más de 15 años.

54:37

Hace tiempo hubo un gran avance en

54:38

ingeniería de software en lugar de la idea

54:40

Ese software sería mejor con

54:41

elaborada planificación y fases llamadas

54:44

modelo de cascada esto fue una rebelión y

54:47

La rebelión fue que lo vamos a hacer.

54:49

ágil vamos a hacer

54:50

desarrollo del trabajo que vamos a hacer

54:51

Prototipos de trabajo que están incompletos.

54:53

ir al cliente y ver lo que

54:55

Queremos a continuación y vamos a hacer esto rápido

54:56

iteración y esto ha sido una revolución

54:59

en ingeniería de software una de las

55:01

modelos es lo que se llama un scrum

55:03

La organización es del rugby y eso.

55:05

Tendrás un pequeño equipo y lo harían.

55:07

hacen el trabajo y ellos hacen estos sprints

55:09

donde consigues construir el prototipo y

55:12

luego haz una pausa y ve lo que haces a continuación, entonces

55:14

Trabajar más duro que la buena noticia de Sprint es

55:17

El moderno software CAD nos permite utilizar

55:20

algunos de estos desarrollo de software

55:21

Técnicas y para que pequeños equipos puedan.

55:23

hacer algo de eso con la abstracción y

55:25

reutilizar aquí hay un ejemplo de Berkeley en

55:27

términos de tres diseños diferentes allí

55:31

es la columna de la izquierda estos son todos

55:34

Se arriesga cinco ejemplos es un tres etapas

55:36

32 bits de una tubería de tres etapas de 32 bits

55:39

diseñar el medio es el clásico

55:40

Diseño de cinco etapas de 64 bits y el derecho.

55:43

una es una máquina fuera de servicio de 64 bits

55:46

diseñar las líneas únicas de código de todos

55:49

ellos menos de la mitad del código es

55:51

Único que puedes compartir a través de ellos en el

55:53

Diseños de cohetes casi ninguno soy yo

55:55

y por lo que es enorme cantidad sabes que es

55:57

sabes el nivel de elevación de

55:59

La abstracción reduce las líneas de código.

56:00

Pero lo importante es conseguir la reutilización del código.

56:03

ahora parece que sabes como haces esto

56:06

un mes de vuelta cuando estás haciendo

56:07

Hardware bien lo que el grupo evolucionó fue.

56:10

este modelo iterativo en primer lugar si

56:12

puede hacer su innovación por lo que el simulador

56:14

hazlo allí, entonces es como un software

56:16

ahora pero no puedes correr tantos relojes

56:19

ciclos que no puedes hacer miles de millones o

56:20

billones de ciclos de reloj si vas a

56:22

El FPGA se puede así que si quieres ver

56:24

como realmente funciona

56:25

A continuación, realice los cambios del FPGA el

56:27

una gran noticia que ha pasado en el último

56:29

pocos años es que hay casos de

56:32

FPGA en las nubes que ni siquiera tienes

56:33

comprar hardware para hacer FPGA para que pueda

56:35

solo ve a la nube usa ese alguien

56:37

otra cosa lo configura todo y lo mantiene y

56:38

Puedes usarlo y con esos ejemplos.

56:40

en esta conferencia así, pero tú

56:43

Sabemos si realmente vamos a hacer energía.

56:45

Y realmente nos va a importar el costo

56:46

Tenemos que llegar al diseño correcto y por lo tanto

56:48

Esa es la siguiente iteración una vez que las ideas

56:50

Trabajar en el simulador y FPGA en el

56:52

siguiente diseño ahora los flujos de diseño

56:55

te da algunas estimaciones hay más

56:56

Trabajo que tienes que hacer para ser realmente

56:59

listo para grabar lo que llamamos una cinta

57:00

en pero entonces es que es bastante bueno ahora

57:03

podría detenerse ahí

57:04

podríamos detenernos allí porque realmente

57:06

Puede estimar el área y el poder con

57:07

buena precisión en la velocidad del reloj y todo

57:09

esas cosas así que ¿por qué no parar allí es

57:12

porque somos la gente de hardware más

57:15

La diferencia es que tenemos más

57:17

la gente de software como están atrapados en

57:19

El ciberespacio sabes que el programa es

57:21

a su manera nunca es no físico

57:22

construimos cosas obtenemos algo

57:25

espalda física y ahí está la emoción

57:27

cuando el chip regrese se va a

57:28



trabaja qué tan rápido va a funcionar cuánto

57:30

poder y esas cosas vuelve por lo que la

57:32

razones para construir un chip es recompensar a

57:35

emoción para todos involucran la

57:37

estudiantes de posgrado o la empresa para obtener

57:38

Las fichas de vuelta que debe ser realmente

57:40

caro no, no es realmente caro

57:43

hemos tenido chips de prueba para siempre que estamos

57:45

barato para que puedas conseguir cien por uno

57:48

uno sabes un milímetro cuadrado

57:50

chips en 28 nanómetros por \$ 14,000 por qué

57:54

esto es emocionante ahora porque con

57:56

La ley de Moore donde estamos es millones.

57:58

De los transistores se puede obtener un riesgo de cinco.

57:59

Núcleo en la Nvidia DLA en un pequeño cofre

58:02

nave y es sólo \$ 14,000 por lo que

58:04

Todos pueden pagarlo ahora si quieren.

58:06

para construir un chip realmente grande que sería

58:07

El último paso y por supuesto que será

58:09

más caro pero todos pueden pagar

58:10

para construir fichas hoy como ejemplo en

58:15

Berkley liderado por christos te amo

58:17

Sé que construyeron diez fichas en cinco años.

58:20

y así con este modelo ágil derecho

58:22

No esperes un poco como yo

58:24

Lo hice y luego lo saque y vea qué

58:26

sucede que sólo sabían bien la próxima

58:27

iteración de cinta para que el graduado

58:29

Desde que salió programé sentí muy

58:30

Confiando en que los chips van a funcionar.

58:32

porque construyeron muchos de ellos en cada uno

58:34

Una guía un poco mejor la

58:35

modelo ágil es superior para envolver

58:39

Antes de hacer preguntas, John y yo pensamos.

58:41

estamos entrando en una nueva edad de oro en el

58:44

final de la escalada de Dennard y la ley de Moore

58:45

significa que los arquitectos es donde vamos

58:48

Innovar si quieres hacer cosas.

58:50

ir mucho mejor cuando no va a

58:51

pasar en la microarquitectura

58:53

seguridad de nivel claramente necesita innovación

58:55

También si solo tenemos soluciones de software.

58:58

No van a conducir a más seguro

59:00

Los sistemas de la principal específicamente que

59:02

está elevando el nivel de abstracción a

59:04

Que sea más fácil para el programador también.

59:06

facilita a los arquitectos

59:07

Innovar y el principal específico.

59:09

Las arquitecturas están recibiendo estos factores.

59:10

de 20 o 40 no no 5 o 10% el abierto.

59:15

La arquitectura es un implemento abierto  
59:18  
Reduce el esfuerzo por involucrarte.  
59:20  
Puede entrar y gustar a su compilador y  
59:22  
compañeros de trabajo y hacer  
59:23  
mejoras a estos dispositivos y  
59:25  
todo el mundo se pone a trabajar en ello es una  
59:28  
inclusiva democrática la nube FPGA es  
59:31  
Haz que sea aún más fácil para todos  
59:33  
construir cosas y lo que parece personalizado  
59:35  
Hardware y este ágil desarrollo de Ford.  
59:39  
para hacer chips así como John dijo como nuestro  
59:42  
época en que éramos jóvenes arquitectos en  
59:45  
la década de 1980 es un buen momento para ser un  
59:48  
Arquitecto tampoco que después del inmenso.  
59:50  
Árbol y con eso tendremos preguntas.  
60:01  
no estoy bien esa fue la charla más rápida que  
60:06  
Alguna vez escuché a Dave pensé que pensaba John  
60:09  
me iba a cortar, pero no está bien  
60:12  
hay un micrófono oh aquí vamos así  
60:15  
solo llamaremos a todos por su nombre Joseph  
60:17  
Charles gracias por el pensamiento que quiero  
60:21  
volver a lo que John dijo que es muy  
60:23  
Fácil de conseguir 62,000 de aceleración desde  
60:26  
lenguajes de scripting en realidad un sistema  
60:29  
como el compilador v8 de Google es extremadamente  
60:34

eficiente tiene millones de líneas de  
60:36  
Código y si nos fijamos en el clima  
60:38  
rendimiento perdido no hay plata  
60:40  
bala está por todas partes  
60:42  
hay sistema de tipos hay múltiples  
60:47  
capas de compiladores la resolución en la mia  
60:49  
almacena cachés recolección de basura sip  
60:52  
regular y cachés no funcionan, así que nadie  
60:56  
dijo que iba a ser fácil, creo que él  
61:00  
lo que dijo era el potencial está ahí  
61:02  
para 60 mm sí no tienes que conseguir todo  
61:04  
de eso si obtienes un factor de mil  
61:08  
eres un héroe correcto y obtienes uno  
61:10  
sesenta de lo que hay disponible creo  
61:12  
tienes razón, se extiende alrededor de su  
61:15  
voy a creo que lo interesante  
61:17  
La pregunta será cómo combinar el compilador.  
61:21  
enfoques con quizás nuevos tipos de  
61:24  
Soporte de hardware que te ayudará a poner.  
61:26  
los dos de nuevo juntos recordar algunos de  
61:28  
tienes edad suficiente para recordar la edad  
61:30  
de maquinas Lisp y espuelas y cosas  
61:32  
así tuvimos un montón de ideas allí  
61:34  
Bueno, ahora tenemos apalancamiento de programación  
61:37  
eso es sustancial cuáles son los correctos  
61:39

ideas de arquitectura para que coincida con eso

61:41

la tecnología del compilador no separa la

61:43

Dos no ponen a los arquitectos aquí.

61:45

Y encerrarlos lejos de la arquitectura.

61:47

La gente se mezcla y tu compilador

61:49

gente que tienes que mezclar juntos solo

61:51

Finalmente señalando que es como el

61:54

El desarrollo tomó millones de ingeniería

61:56

horas de ingeniero ya sí, pero mira

61:59

Somos investigadores, ¿verdad, John, si lo sabes?

62:03

Algo podría habernos dicho que cuando

62:05

Estábamos sabes en nuestros años 30 es correcto

62:08

que no se puede construir un microprocesador

62:09

Eso te lleva a saber cientos de Intel

62:11

ingenieros no puedes no puedes hacer eso

62:13

Esta es una oportunidad que no tenemos otra opción.

62:15

derecho si

62:16

La ley de Moore seguía vigente.

62:17

microprocesadores se duplicaban cada 18

62:19

meses no sé si las personas que trabajan

62:20

en esto pero tenemos que hacer algo -

62:22

Sí, y aquí hay una oportunidad y tú

62:24

debería pensar en lo que realmente

62:25

sucedió en toda la era VLSI la

62:29

Comunidad académica construyó un conjunto de diseño.

62:31

herramientas que no eran herramientas de diseño fuera

62:33

allí recuerdo haber bajado y visto

62:35

Zilog diseñando Z 80s y hubo una

62:38

pedazo de mylar lo pegó en la pared

62:40

Eso fue alrededor de 20 pies por 20 pies y

62:42

Así es como estaban haciendo el diseño si

62:44

Habíamos adoptado esa metodología de diseño.

62:47

nunca hubiéramos podido vestirnos

62:49

esto pero de todos renunciar

62:51

todavía serían estudiantes de posgrado, sí todos

62:56

bien también un micrófono micrófonos van

62:57

adelante me inclino cómo canción soy un doctorado

63:00

estudiante frente amigo hay esta cosa

63:05

Llamado tragar el micrófono para que

63:07

Tengo que en realidad sí, así que di una vez más lo

63:09

lo hace, así que estoy siendo alojado en un doctorado

63:11

diente frontal del estudiante y trabajando en una

63:14

Thomas Pacifica la arquitectura por lo que para

63:16

Thomas computadores portátiles aquí

63:19

En realidad estamos pidiendo que tengamos que acercarnos.

63:22

Una es que correrá una nueva terminal.

63:26

aplicación poco común convencional

63:29

ecosistemas informáticos para que podamos encontrar que

63:32

¿Dónde está la restricción y el socio y

63:36

luego mejorar eso por lo que un ejemplo es el

63:40

esteroides internos mal dije que el

63:43

Primero y encontrar que la computación.

63:47

curador es el socio por lo que la propuesta

63:50

diseñar el acelerador de servicios TNR y

63:54

entonces encuentran que el conjunto de instrucciones

63:57

La arquitectura es y otros socios por lo

63:59

El propuesto para el Cabaret con esto es

64:02

La instrucción es para el general.

64:05

Servicio y otro enfoque para el

64:08

Thomas Patil Computing que esto es como

64:11

un enfoque de diseño de trabajo por lo que en este

64:15

Acércate y espera solo una propuesta.

64:18

solución directamente desde el software

64:21

framework eligiendo otro hardware

64:25

arquitectura así que un ejemplo está bien

64:27

cuál es la pregunta

64:29

¿has dicho antes cuál es la pregunta

64:31

así que la pregunta aquí es la pregunta

64:33

así para Thomas percibir y computar

64:36

gente que conoces tenemos que acercarnos tan

64:40

¿Tiene algún comentario o sugerencia?

64:42

sobre esto es que te acercas gracias

64:45

Creo que el supuesto enfoque creo que ambos

64:50

Los enfoques tienen mérito, creo que decir uno

64:53

El enfoque dominará todo el espacio.

64:55

Probablemente esté equivocado y al final el  
64:59  
Ventajas de los puntos de referencia para varios  
65:02  
Las áreas confirmarán qué enfoque funciona  
65:05  
mejor que eso te importa si bien una  
65:09  
enfoque cuantitativo exactamente muy agradable  
65:17  
hablar  
65:18  
por lo que Jason Mars Michigan, así que estamos tipo de  
65:23  
en una era en este momento donde vamos  
65:26  
derecho especializado para que puedas pensar en ello  
65:28  
como un Sisk extremo donde estamos realmente  
65:30  
Especializada para usos complejos particulares.  
65:33  
casos y sabes por lo que ustedes son  
65:36  
Los hardcore son defensores de un punto de riesgo.  
65:39  
De vista en el mundo y por eso me pregunto  
65:41  
¿Qué tipo de lecciones podemos tomar de  
65:43  
tu experiencia teniendo esta batalla  
65:46  
antes de Sisk versus arriesgar qué tipo de  
65:48  
lecciones de riesgo podemos tomar ser que la  
65:51  
La comunidad en general ha estado tan enfocada.  
65:53  
en la aceleración de especialización, etc, pero  
65:59  
El hardware y el software juntos.  
66:00  
no separe el hardware del  
66:04  
software pensar en el problema en una  
66:06  
La moda integrada creo que esa es la  
66:08  
Lo más importante es que nos referimos al riesgo  
66:11



Las ideas no habrían venido sin el  
66:14  
ideas sobre qué tecnología compilador  
66:16  
Necesitaba lucir bien y esa es la  
66:19  
llave interior por lo que es lo mismo aquí  
66:20  
no los separe y optimice  
66:23  
a través de ese límite y pensar cómo  
66:25  
encajan y como dijimos que sabes  
66:27  
Es que es un derecho fortuito que es  
66:32  
lenguajes específicos de dominio están siendo  
66:34  
Creado para hacer más programadores.  
66:36  
eficiente en estos dominios y eso es  
66:38  
Independiente de todos nosotros pero qué bueno.  
66:41  
Noticias para nosotros elevar el nivel de  
66:43  
abstracción  
66:43  
Acción que hacen fácil el programa también.  
66:44  
plantea donde podemos innovar a la derecha y  
66:47  
así que sí, eso es lo que dijo John  
66:49  
aplicación de personas integrada verticalmente caso  
66:51  
significa más de nosotros en la vieja especificación  
66:53  
días no creo que ni siquiera pensé  
66:54  
sabes lo que era, sabes, no sé  
66:56  
Su programa para nosotros lo estudiamos para hacerlo.  
66:59  
corre más rápido ahora en él porque no lo hiciste  
67:01  
no has ganado nada si estudias el  
67:03  
Programa Specht porque no quieres  
67:04

cambiarlo, esto es realmente diferente

67:05

derecho del mundo vas a la

67:07

Creo que las oportunidades que vemos aquí son

67:09

por eso está sucediendo en Burley

67:11

empresas integradas estas empresas

67:12

tener las habilidades en muchos niveles diferentes

67:14

y están encontrando que sabes grande

67:16

oportunidades gracias

67:18

bien bien primero de todo gracias

67:22

mucho por tu contribución realmente

67:24

apreciado gracias por lo que la pregunta que

67:27

tener es realmente acerca de dominio específico

67:28

arquitecturas no sé todo el

67:30

comunidad de arquitectura en este punto de

67:32

el tiempo está hablando de dominio específico

67:33

siendo las arquitecturas la solución para una

67:35

Muchos de los problemas que hemos trabajado.

67:38

en la arquitectura de la computadora móvil de

67:39

Louis a ellos mi tiempo en el pasado

67:41

Varios años significaron este dispositivo que

67:43

aquí tiene más que una CPU

67:46

y GPU o que estas personas han sido

67:48

haciendo arquitecturas específicas de dominio para

67:50

más de una década en este punto algunos curiosos

67:53

mientras estamos buscando en el espacio del servidor que

67:56

hemos pasado de CPU GPU a TP

67:58

ayer o dice que estas cosas tienen 10

68:01

veces más en ese número en términos de

68:03

integración que el SOC

68:05

Entonces, ¿cómo estás buscando dominio específico

68:07

Arquitecturas en este dominio y dicen.

68:10

eso es nuevo o es realmente lo que

68:13

son las lecciones que podemos tomar, sí voy a

68:15

Intento y hago más corto, por lo general también voy

68:17

largo pero básicamente la diferencia es

68:19

programación correcta como yo estaba

68:22

involucrado con el proyecto que es

68:25

lo que se llama el núcleo virtual de píxeles en

68:28

Google en mi sabático y hubo una

68:31

hardware que se llama ISPs simplemente lo hacen

68:33

todo esto en hardware era un fijo

68:35

línea de hardware la idea era hacer esto

68:36

Más programables y en realidad el

68:38

Se llama lenguaje específico de dominio

68:39

halide que ataron en eso, pero yo

68:41

piensa que esa es la diferencia

68:42

Ciertamente hay un montón de especial

68:45

aceleradores de propósito que son muy

68:46

energía eficiente que tipo de hacer uno ish

68:48

cosa y así podemos mantener programable

68:51

¿Podemos mantener el software y la aceleración?

68:54

derecho eso es exactamente correcto

69:01

so Vivek Sarkar Georgia Tech so David

69:05

John me gustaría aprovechar tus 40 años

69:08

de experiencia en educación y pensamiento.

69:11

sobre cómo ha sido la informática

69:13

enseñado durante este período de 40 años y usted

69:16

saber a medida que te acercas a buscar

69:18

oportunidades en esta era dorada

69:20

¿Ves lagunas en la preparación o

69:22

estudiantes de pregrado y posgrado

69:25

para hacer frente a esta era brechas en el

69:33

currículo el campo se amplía la gente

69:35

han caído así como cuando primero

69:36

Empezaron todos los compiladores estudiados que

69:38

no pasa

69:39

Yamo es, por ejemplo, tan claramente

69:42

Me refiero a la seguridad claramente no estamos

69:44

enseñando a nuestros estudiantes lo suficiente sobre el

69:46

importancia de la seguridad como pensar

69:48

sobre eso, había más si vas

69:50

Volver a los libros de texto de hace 20 o 30 años.

69:53

probablemente encontrarás más cosas en

69:55

Seguridad que en algunos modernos.

69:56

Los libros de texto que necesitamos para arreglar eso es

69:58

obviamente obviamente crucial creo que para  
70:03  
mejor peor nuestro campo se expande por lo  
70:05  
dramáticamente que es imposible para una  
70:08  
estudiante universitario o incluso un estudiante graduado  
70:10  
para dominar todo bien cuando era un  
70:12  
estudiante de posgrado que podría ir a básicamente  
70:14  
cualquier defensa de doctorado y entender lo que es  
70:16  
pasando podría ir a cualquier charla de trabajo y  
70:18  
Entiendo lo que estaba pasando para decir eso.  
70:20  
puedes hacer esto hoy es solo  
70:22  
imposible el campo ha crecido a pasos agigantados  
70:23  
y bajadas por lo que tenemos, excepto que nosotros  
70:26  
Tenemos que enseñar a nuestros estudiantes algunos conceptos básicos.  
70:29  
algoritmos de computación complejidad, pero  
70:32  
También tenemos que enseñarles algo.  
70:33  
conceptos importantes del sistema correcto  
70:35  
paralelismo seguridad cosas como esa  
70:37  
y tenemos que darnos cuenta de que estamos  
70:40  
Tendré que enseñarles a ser de por vida.  
70:42  
aprendices porque van a tener que  
70:44  
volver a aprender y volver a participar porque nuestro campo  
70:47  
va a seguir expandiéndose y  
70:49  
flor creo que soy la otra cosa es  
70:51  
debido a esta emoción sobre los datos  
70:53  
ciencia y pensar estadísticas ahora es  
70:56

dependiendo de la escuela que puedan obtener

70:57

Un montón de estadísticas o puede que no, pero

70:59

Creo que la informática se acerca.

71:01

a las estadísticas y viceversa va a

71:03

ser una cosa para que sepa que supongo que

71:05

mira tu currículum y mira

71:07

Gracias Rita Sarita lo siento muy guardia.

71:11

Universidad de Illinois, así que gracias um

71:14

así que hablaste de entrar en un dorado

71:16

edad de la arquitectura de la computadora y usted

71:18

También se habló de cómo va a ser

71:19

importante para hardware y software

71:21

la gente trabaje bien y yo

71:24

Creo que este grupo tiene ese derecho obtenemos

71:27

que tenemos que trabajar con gente de software

71:28

En realidad estamos acostumbrados a hacer eso, pero el

71:30

de otra manera no está aquí a la derecha

71:33

En general, culturalmente creo que el software

71:36

la gente no consigue el hardware no es

71:38

parte de los departamentos de informática

71:40

cultura, así que mi pregunta es ¿qué podemos hacer

71:44

mover esa aguja así que lo primero

71:45

es decirle que sí chicos que hemos estado

71:47

dándote un viaje gratis por 30 años bien

71:51

usted escribe su software de mala calidad y nosotros

71:53

lo hizo más rápido

71:54

Está bien, eso es todo lo que sabes que

71:57

Quiero decir, recuerdo que sabes ed ed

72:04

Berkeley en como 2003 fui a la

72:07

el almuerzo de la facultad es que sí voy a dar

72:08

una charla el futuro es solo paralelo

72:12

Los núcleos no se están volviendo más rápidos que yo no

72:14

saber cuántos de mis colegas creen

72:16

yo cuando lo dije sabes todos ellos

72:18

todo el mundo tiene una opinión pero creo

72:19

lo que nuestro trabajo es decirles que se acabó

72:22

el viaje gratis ha terminado no hay hay

72:24

No hay magia, no hay ceño fruncido, es

72:27

No viene por la colina la oportunidad

72:30

ir hacia adelante va a ser

72:31

hardware / software de diseño de carbón y

72:33

Los arquitectos van a jugar este gran papel

72:35

Si quieres cosas más rápidas o más bajas.

72:36

energía que va a este es el único camino

72:38

te dejé saber y creo que es nuestro yo

72:40

Creo que en realidad es todo tu trabajo para

72:42

Explica eso a tus colegas y al

72:44

facultad que consigue que van a ser una

72:46

daño derecho porque estamos justo a la derecha

72:48

no va a pasar nada mas

72:50

nunca nos equivocamos no nosotros la arquitectura  
72:55  
comunidad sabes que sabemos lo que es  
72:56  
pasando en hardware y cosas como esta  
72:57  
y eso es lo que sabes que sabes que puedes obtener  
72:59  
una pregunta bien la computación cuántica no ir  
73:02  
en y la computación es una muy emocionante  
73:03  
cosa que no es mañana, no es  
73:05  
va a pasar mañana así que no es el  
73:07  
mismo horario que la fusión, ¿quién es yo, así que me sale  
73:10  
ese video está en vivo y ahora  
73:12  
todos te escuchan oh veo  
73:14  
eso está bien, puedes subestimar, puedes, puedes  
73:16  
clip de video clip tanto sí hey marca marca  
73:20  
marca Tommy Microsoft tan claramente que hay  
73:23  
Un renacimiento de arquitectos y microempresas.  
73:25  
Arquitectos de construcción muy innovadores.  
73:27  
chips para la IA algunos de ellos que te requieren  
73:30  
saber miles si no decenas de miles  
73:33  
de núcleos lo que no veo tanto como una  
73:35  
Renacimiento de la gente del compilador que conoces  
73:38  
hay tu comentario o vas mas  
73:39  
vertical pero se han ido los días que nos  
73:42  
necesaria para compilar a un simple riesgo es una  
73:45  
Ahora se está compilando para estos gigantescos  
73:47  
máquinas distribuidas  
73:49



viste crees que estamos

73:50

alimentando la pipa o el talento en ese

73:53

espacio bien creo que es una oportunidad

73:56

Mi punto de vista es que golpeamos la pared

74:00

En la compilación empujamos la tecnología del compilador

74:04

por lo que podría llegar si estuviéramos hablando

74:07

sobre compilar desde el nivel más bajo

74:10

idiomas de nivel algunos escriben C Fortran

74:11

cosas como esas empujamos eso

74:13

Tecnología en la medida de lo posible

74:14

no podríamos empujarlo más me refiero

74:17

entonces mucha gente que trabajó en ello

74:19

tipo de renunciar a hacer cosas como

74:22

reorganizando la memoria para mejorar la memoria

74:25

rendimiento que llegó tan lejos como pudimos

74:26

conseguir programas simples puedes hacerlo

74:29

Multiplicar la matriz puedo hacerlo automáticamente

74:31

pero toma una aplicación grande o

74:33

Matrices algo dispersas lo olvidan

74:34

sistema se rompe por lo que ahora hay una

74:36

oportunidad de volver a participar y renovamos

74:41

Toda esa zona y rejuvenecerla y yo.

74:44

Creo que es una oportunidad emocionante y

74:45

Creo que esa gente obtendrá más

74:47

apalancamiento si están trabajando de la mano

74:50

con personas de hardware con arquitectos y  
74:52  
Usted sabe y esta crianza en la abstracción  
74:54  
para los lenguajes específicos de dominio son  
74:56  
esperanzador derecho sí matrices son  
74:59  
primitivos derecho y eso lo hace  
75:01  
más fácil de hacer hay un proyecto en Google  
75:04  
Xoa llamado que está tratando de hacer  
75:06  
optimización de este nivel muy alto  
75:09  
nivel que es un Newton bien a mi yo  
75:12  
Creo que es un nuevo reto para el.  
75:13  
enfoque comparativo así que sí, sí por lo  
75:17  
Sí, también podríamos saber que estamos  
75:18  
Probablemente no voy a hacer todo el trabajo John  
75:21  
Lo que podemos hacer es establecer una especie de  
75:23  
oportunidades y esto es esto es  
75:25  
claramente una oportunidad y ustedes saben esto  
75:27  
vamos de una era muy ajustada  
75:29  
ya sabes  
75:29  
papeles con pequeñas mejoras  
75:32  
estamos hablando  
75:34  
números gigantes están ahí y sabes  
75:36  
como John es así con John y tú  
75:39  
No puedo trabajar en los nuevos premios de maestros.  
75:41  
exactamente hay un solo requisito que  
75:44  
Tengo que tener más pelo que David.  
75:50

Moochi de - Hola Universidad, soy tu  
75:53  
Estudiante de doctorado en tu charla hablas  
75:56  
acerca de que estamos en la Edad de Oro de  
76:01  
Arquitectura que podemos desarrollar.  
76:03  
lenguaje específico del dominio para neural  
76:05  
redes y como piensas tu  
76:08  
red afectará a nuestra comunidad para  
76:11  
ejemplo sin red  
76:13  
tal vez cactus tomo que bien como  
76:15  
¿Puedo ejecutar una cartografía tan joven acantilado  
76:17  
aquí dio una dirección magistral en una  
76:19  
Taller llamado ADA en el que fue como  
76:24  
¿Podemos usar el aprendizaje automático para diseñar?  
76:26  
computadoras correctas y creo que eso es una  
76:29  
idea muy interesante, ¿verdad?  
76:30  
Algo llamado aprendizaje automático.  
76:32  
auto aprendizaje automático donde usas  
76:34  
Aprendizaje automático para diseñar los modelos para  
76:36  
sabes que esto está revolucionando a  
76:39  
aprendizaje automático de muchos campos sería  
76:41  
Interesante para usarlo aquí, así como  
76:43  
Los predictores de rama deben basarse en  
76:45  
Principios de aprendizaje automático.  
76:47  
de aprendizaje automático es algo más  
76:48  
como pregunta qué tan bien sería qué tan bien  
76:51

podríamos diseñar máquinas mejor

76:53

Si usamos el aprendizaje automático eso es

76:55

otro sabes potencialmente emocionante

76:57

Idea revolucionaria y vería a Cliff's.

77:01

nota clave mañana

77:03

bueno hola soy Lee del pasillo en el que trabajo

77:06

Los aceleradores de uso de hidrógeno por lo que acabo

77:10

quiero mirar al nivel del sistema como

77:12

Tienes todos estos dominios específicos

77:14

poderes así que ahora estamos hablando de

77:18

aplicaciones en la nube en todos

77:21

estos campos necesitamos algo para

77:24

agregue todas estas horas podría usted

77:26

Comento esto porque pienso en mi

77:28

opinión hay una que hay mucho como

77:30

mayores lagunas de sitio soberano y en nuestra

77:33

sociedad para la agregación hay una gran

77:36

brecha hay una gran brecha en tanto quiero decir que

77:38

creo que eso es lo que lo hace un emocionante

77:39

tiempo hay muchas oportunidades de tipo

77:42

de repensar cómo programamos cómo nosotros

77:46

organizar nuestra arquitectura

77:48

Y eso proporciona un por eso pensamos

77:51

Es una nueva edad de oro que hemos tenido.

77:53

esta carrera nos encontramos con toda esta curva

77:55

cuando empujamos todo por ahí

77:56

Esas ideas se hacen tiempo para repensar el

78:00

problema y eso significa que hay una

78:02

gran oportunidad tal vez deberíamos hacer

78:04

es claro lo que queremos decir con la edad de oro por lo

78:06

Somos una especie de investigadores de la derecha y qué

78:09

Podría ser realmente aterrador para las empresas

78:11

lo que hace mucho oro no es lo que

78:15

estamos hablando de esto no como Oh

78:16

dinero en el suelo esto va a Wow nosotros

78:18

No se que hacer

78:19

para que la gente como nosotros tenga una

78:21

oportunidad de liderar el camino cuando cuando

78:23

es muy claro que las empresas pueden

78:25

Sigue haciendo lo mismo y gana mucho.

78:27

de dinero sabes que es muy reconfortante

78:29

para las empresas pero no tan emocionante para

78:31

investigadores, así como sólo hay una

78:33

entorno rico en objetivos para los investigadores

78:35

y la arquitectura para hacer grande

78:37

Contribuciones a que conoces la sociedad.

78:39

bien nos han dicho que la recepción

78:42

está a punto de empezar, así que tenemos que agradecer

78:45

todos por todas sus grandes preguntas

78:46

y Vicky quiere revelar todo bien

78:57

gracias solo quería tener a todos

78:59

darles otra gran mano gracias por

79:01

esa maravillosa charla

79:16

y sí, ahora hay