# Copy Modifications & Additions

## Landing Page

**The API for Unlimited Compute Power**

Build and scale massively parallel apps with ease.  
The future of high-throughput computing is **compute.for( )**

What would you build with a virtual datacentre?

**What Is It?**

The **Distributed Compute Protocol** is a lightweight, powerful framework for HTC applications. It is a cross-platform standard for computing that makes the same code run on all hardware.  
  
DCP lets users provision and share heterogeneous compute resources across any network, from the office to the globe. Even desktops, laptops, and smartphones can seamlessly power virtual clusters.

**What’s New About This?**

While there are many frameworks for distributed and grid computing, DCP brings this into the 21st century.

Previous solutions were not standardized for different hardware and networks, while DCP executes identical code anywhere. It gives developers a way to build arbitrary applications rather than confining them, and makes these shareable with the entire world. Finally, DCP combines both on-premises networks with the cloud to put more computing power in reach than ever before.  
  
DCP lets you scale vertically and horizontally in three ways:

**A picture containing clock, drawing

Description automatically generated** *<New Image>*

**Intuitive Programming**

DCP’s Compute API is the easiest way to express arbitrary parallel workloads. It is used everyday by the same developers who built it for AI, Computational Finance, Bioinformatics, and more.

**A picture containing clock, meter

Description automatically generated** *<New Image>*

**Fungible Infrastructure**

Compute and network resources are completely abstracted from the underlying hardware, so you always get seamless performance even across different clouds.

**A picture containing clock

Description automatically generated** *<New Image>*

**Virtual Clusters**

DCP uses ultra-lightweight web technology that gets closer to bare metal performance than any other. A workload run on DCP is commonly more efficient than the same run in a Container or VM.

**Compute Wherever, However.**

DCP allows you to connect and disconnect your own hardware to build high performance compute grids. Additionally, compute grids from datacentres and other institutions are publicly available through the Distributed Computer. These two models are known as Compute Group and the Distributed Computer.

**A picture containing monitor, street, computer, light

Description automatically generated** *<New Image>*

**Compute Groups**

Accelerate high-throughput computing jobs with underutilized hardware you already own. Any CPU or GPU can be connected to build a fully functioning compute grid across a single LAN or even an international WAN.

A picture containing light, green, monitor, sitting

Description automatically generated *<New Image>*

**The Distributed Computer**

The ultimate private cloud powered by a global federation of compute providers, the Distributed Computer finishes the toughest jobs in minutes. Access thousands of cores with a second’s notice, all without managing a single Container or VM.

**Be Bold, Be First**

Be one of the first to build cutting edge applications with free compute on DCP. The **First Devs Program** will send you all the latest updates, and give you privileged first access rights when you join today!

**Why Should I Build On DCP?**

From deep neural networks to multiphysics simulations, DCP powers cutting-edge applications.



**Accelerated Compute**

Access a larger pool of compute resources both on-premises and from the cloud without managing a single server.

 *<New Image>*

**Runs Everywhere**

Code becomes fully interoperable and portable with the first framework that runs everywhere, from single board computers and the IoT to enterprise servers.



**Improved Efficiency**

DCP is lighter-weight and far more responsive than Containers or VMs, and requires almost no programming experience to master.



**Effortless Orchestration**

DCP lets nodes easily join and move between software defined clusters. Its overlay network is always working hard to optimize your jobs.



**High Dependability**

Harness the advantages of a distributed system with uninterrupted service and security. DCP delivers results in real-time and with an integrated failover system.



**Global Applications**

Customize your workflow with DCP 3rd party apps, or build your own. Reach a global audience with your software and earn royalties for every use.



**Browser Based**

Build and deploy your application from DCP's browser-based IDE. Building powerful compute clusters is as easy as configuring a few settings in a web browser.

*<New Image>*

**Native Parallelism**

Effortlessly build applications to take advantage of thousands of cores. The compute.for( ) command has unmatched support for data parallelism at any scale.

*<New Image>*

**Microservice Ready**

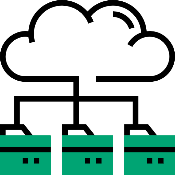
DCP is useful for fully integrated workflows, offloading compute-intensive steps, or handling requests from other apps. Every application bounded by HTC will benefit.

**Get US$100 of free compute for your first workload on DCP.**

**How does DCP Work?**

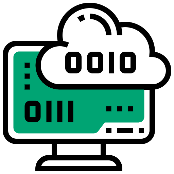
DCP is a complete distributed OS built to be standards-based, high performing, and future-proof.

**Data Parallelism**

 *<New Image>*

The Compute API uses a memory safe generator called compute.for( ) to express parallel workloads. A developer only needs to identify the ideal unit size of their data to process in parallel, and write the methods to execute.   
  
Each distinct parallel unit of work in DCP is called a 'slice', which contains both the data and code that is to be run. On a CPU, one slice generally maps to one core. These slices are then distributed to the proper hardware to be computed.

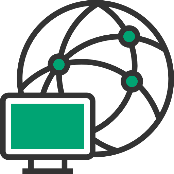
**Hardware Abstraction**

 *<New Image>*

DCP uses a lightweight JavaScript engine. It is more lightweight than Containers or VMs, and completely removes the complexity of managing infrastructure.

An instance of DCP on one machine is called a Worker. The Worker is a Node.js daemon that coordinates the developer’s workload and transmits the results back. Starting a new Worker is as easy as opening a web page, and has these benefits compared to Containers and VMs:  
  
> Lower memory overhead.  
> Faster response times, including for cold-starts and application switching.  
> Greater flexibility, being able to run across any fleet of hardware.  
> Improved security from having no I/O capability.

**Network Scheduling**

 *<New Image>*

DCP applies its own algorithms to solve the old problem of scheduling jobs on a distributed network.   
The DCP Scheduler is the primary module that does this, broadcasting both time and space-sharing commands while avoiding common issues like thrashing.  
  
The Scheduler takes multiple factors into account such as speed, energy use, and even the level of CPU and memory capacity it should not exceed. Because DCP is suitable for all environments, the Scheduler works across everything from bare metal to Kubernetes deployments.

**See how DCP is an accelerator for parallel and distributed computing.**

**A Global Supercomputer for**

**> Bioinformatics**

**> Computational Finance**

**> Big Data**

**> Monte Carlo Analysis**

**> Neural Networks**

**> Hyperparameter Search**

**> Computational Chemistry**

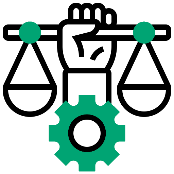
**> Edge Computing**

**> Hybrid Cloud**

The Distributed Computer is a global cloud federation powered by everything from universities to enterprise datacentres. It is as easy to access any of these resources through the Compute API as programming your laptop.  
  
**With the Marketplace, your code can:**

 *<New Image>*

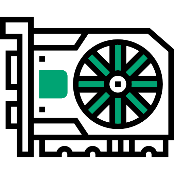
Power applications run by thousands of paying users

*<New Image>*

Load balance your apps during intensive operations

 *<New Image>*

Automatically scale across thousands of cores on-demand

 *<New Image>*

Leverage compute accelerators like GPUs & NPUs without needing a rewrite

**DCL Compute Grants**

Distributed Compute Labs is a global community of developers, cloud providers, and other institutions. It connects people with diverse mission statements to the resources they need through DCP.

If you are a researcher or non-profit who needs computing resources, we may be able to help. Please contact us and we will get in touch.

*<Testimonials>*

“The discovery potential of next-generation astronomical telescopes hinges on access to sufficient compute power to process the tremendous volume of data that will be produced. The Distributed Compute Protocol is an innovative approach to tackling this challenge that holds tremendous promise for my own research to understand the physical processes that drive galaxy evolution in the Universe.”

Dr. Kristine Spekkens, Royal Military College of Canada, Physics & Space Science

“Current advances in brain research are accompanied with an exponential growth in computation needs; Distributed Compute Protocol is an exciting new tool that will enable brain researchers and other scientists to get easy and cost-effective access to unprecedented compute power.”

Dr. Gunnar Blohm, Queen’s University, Computational Neuroscience

"Harnessing the unused power of latent computer cores opens up real opportunity for Canadian researchers to access much-needed advanced computing resources. Advanced research computing is the backbone of innovation, and the potential of the Distributed Compute Protocol to complement and enhance existing resources is incredible."

Nizar Ladak, President & CEO of Compute Ontario

“I work on understanding the major mechanisms governing the transmission patterns of childhood diseases and the impact of public health interventions. This involves fitting complex stochastic models to data. The Distributed Compute Protocol is a very promising tool to aid in my research and other people's work that require large amounts of computing power.”

Dr. Felicia Magpantay, Queen's University, Mathematics & Statistics

**DCP Compute Grants**

DiskFit is a web app made with DCP that helps astrophysicists understand how galaxies form. Originally in Fortran, it now runs in Node.js with DCP.

Learn how what used to take 3.5 days on a single PC now takes minutes, and how DCP can do the same for your application.

**Patch Notes**

DCP is always improving with help from its community and core development team. The project has come a long ways, but there are many exciting developments yet to come. Stay tuned for stateful computing, support for popular integrations, and coarse-grained parallelism!

2020-04-27 - Portal v3.0.1 Brahe

**Features**

* Improvements around keystore handling and work
* Improvements in DCP-Client and job management APIs2019-12-19 - Portal v2.6.6

2020-04-02 - Portal v3.0.0 Brahe

**Features**

* Updated to DCP Protocol v4
* Improvements to dev environment, moving towards the First Dev release!

## The Technology

**Technology**

The DCP stack is built to improve efficiency, scalability, and simplicity. It reflects classic elements of distributed computing with a new vision to make compute resources fully fungible.  
  
Learn how DCP's architecture securely unites millions of heterogeneous CPUs and GPUs.

*A close up of a sign

Description automatically generated <New Image>*

**~~Simplified Programming~~**

DCP uses the JavaScript V8 engine to execute all computations. Unlike other frameworks, this enables the same code to run on any device.

**A picture containing drawing

Description automatically generated** *<New Image>*

**~~Fungible Infrastructure~~**

DCP leverages WebAssembly to provide multiple performance advantages future compatibility with languages besides Node.js like Python, C, and Rust.

**~~Virtual Clusters~~**

~~DCP integrates WebGL for rendering and general computing tasks. It enables GPU processing and, in the near future, General-Purpose GPU compute (GPGPU).~~

**The Tech Stack**

DCP is lightweight overlay network built on a few key software modules. Each module is primarily built on open-source frameworks, with a few novel tweaks.

*<Same Diagram & Copy>*

**Why JavaScript**

Although not often thought of as a tool for compute-heavy jobs, there are several reasons JavaScript is the ideal choice for a distributed compute network:

* It is nearing the speed and effectiveness of low-level languages like C++.
* JavaScript is constantly being improved by Google, Mozilla, Apple, and Microsoft!
* WebAssembly allows high-level JavaScript to have the performance of low-level code.
* JavaScript has the largest and youngest development community, and is the most popular language - NPM is also the largest software repository in the world.

## The Distributed Computer

**A Global Compute Supercluster**

The Distributed Computer is a cloud federation with thousands of devices connected via DCP. This open platform lets workloads run anywhere on any hardware, for optimal scale and performance.  
  
There is no cloud as powerful as every cloud.

**Benefits of the Distributed Computer**

As a global multi-cloud with a unified, secure access point, DCP lets you build for a scale previously limited to large corporations and government.



**Massively Parallel Execution**

Access thousands of CPU and GPU cores for high-throughput computing. DCP's serverless architecture lets you scale up and speed up with a few keystrokes.



**Write for Anywhere**

Traditional microservices and serverless compute lock you into a specific architecture. The Distributed Computer lets you skip across different clouds with the same code.



**Hybrid Environment**

With identical implementation across different architectures, DCP lets you seamlessly burst to a WAN compute cluster from a local one. Different clusters can also seamlessly combine and separate.



**An Open App Economy**

Developers can make the most of applications with embedded distributed compute. The DCP application market provides pre-made software modules as well as SDKs.



**Cross-Platform Data Security**

DCP's data sharding algorithms mean that sensitive data is sent to verified data centers and kept in secure isolation. Unlike the central cloud, data is not vulnerable at rest.



**Get More for Less**

The Distributed Computer matches tasks with different requirements to the right hardware to ensure that minimal memory and compute capacity goes wasted.

**Get US$100 of free compute for your first workload on DCP.**

**What Hardware is Supported?**

The Distributed Computer unifies everything from the cores on a Raspberry Pi all the way to Tesla GPUs (and even the GPUs on a Tesla car!). Whatever your application, go from experimentation to production with the same API.

*<Four Elements Copy Remains the Same; New Images>*

**A close up of a sign

Description automatically generated A picture containing object, clock

Description automatically generated A picture containing clock, drawing

Description automatically generated A picture containing object, clock

Description automatically generated**

**How does it work?**

* **Networking:** The Distributed Computer is a network overlay that abstracts and routes data across heterogenous topographies. Every type of connection from Ethernet to 5G is abstracted in the same way to improve ease-of-use.

Every device connected to DCP is given an identifier tag along with data regarding its bandwidth, security, and owner. When a developer deploys their job through the protocol, all data is sent to the central Scheduler, is fragmented, and transmitted to the ideal device to be computed.

* **Storage:** The user of DCP must currently provide their own storage platform to read and write data. This storage can be either on-premises or cloud-based.

All data is transmitted and computed as JSON objects. DCP Workers have no I/O capability with the underlying hardware, and the Scheduler does not store data that passes through it.

* **Compute:** The Distributed Computer is a pool of serverless compute resources across various nodes and machines. When a developer initiates a workload through this network, it is characterized and then matched with the appropriate type of hardware.

A single workload can be any size, from a single virtual thread running inside an isolated sandbox to multiple datacentres in different cities. All compute is executed inside the same secure JavaScript V8 engine as local DCP instances.

* **Memory:** DCP uses the memory of the underlying infrastructure, which is abstracted away for the developer. A user may however specify a minimum amount of memory that is needed for their application.

DCP instances require almost no memory from the hardware because it uses the lightweight V8 JavaScript engine. Every core on a machine running DCP uses the same overhead, leaving substantially more room for your dataset itself.

* **Providers:** Any person or institution can provide compute for the Distributed Computer as easily as signing in to a web account. The array of hardware that can be accessed is vast, ranging from workstation CPUs to the GPUs in a corporate datacentre.

It is possible for providers to build their own private cloud using a DCP Compute Group and also supply the Distributed Computer. Some may also consume resources from other providers during periods of high traffic, or donate spare cycles for non-profit initiatives.

* **Security:** There are multiple levels of security in DCP. Developers can choose to send sensitive workloads to datacentres with appropriate security and location certificates. Compute Groups multiply the cycles available on-premises so sensitive data never has to leave the four walls.

Workloads from different developers running on the same device are secure from each other as well as the hardware providers, because the DCP Sandbox does not permit I/O. In the near future, different degrees of Homomorphic Encryption will also be integrated.

**The Cost of Compute**

There is tremendous competition for scarce resources at a time when demand is rising exponentially. The Distributed Computer flips the economics of computing so that both the developer and the corporation can win by:

*A close up of a sign

Description automatically generated <New Image>*

**Increasing the Supply**

DCP combines public clouds with previously inaccessible private networks like universities to access more cores, all while recapturing huge amounts of wasted cycles. No cloud is as powerful as every cloud.

*A close up of a sign

Description automatically generated <New Image>*

**Creating Perfect Competition**

Perfect competition is the opposite of a monopoly where price is set by pure supply and demand. Since compute is fungible with DCP, providers are nearly equal from the developer’s point of view and therefore the market is productively efficient.

**A picture containing clock

Description automatically generated** *<New Image>*

**Transparent Price Discovery**

DCP's scheduling algorithms quantify the true economic costs of compute, and put participants on an equal playing field. Because of this, developers can be confident knowing they are not overpaying while providers are not underselling their infrastructure.

**The Supply & Demand Economics**

These economic elements can be visualized in the following chart. At present, there is only a small quantity of core years provided at a low price which are primarily reserved for researchers. The vast majority of public cloud CPU and GPU resources are priced high above what most need.

The Distributed Computer encourages low cost providers of compute to run your workloads, and also encourages existing cloud companies to sell excess capacity at a discount. DCP enables all kinds of developers to experiment and scale big.

From hyperparameter searches to bioinformatics and everything beyond, build it on the Distributed Computer.

Distributed Compute Labs partners with several institutions to bring accessible compute to the people who need it.

Please contact the core developer team with details about your project. We may be able to find incredibly discounted or even free hardware for you.

**Contribute Compute Power**

You can join and earn compute as easily as opening a web page. DCP workers operate both through the browser or as standalone Node.js daemons for Linux, Windows, and OS X.  
  
By powering someone else's applications during your hardware's idle period, you could be accelerating your own when you need it most. You may operate on an entirely volunteer basis, or set an amount to cover the cost of your network and compute resources.

* **Access Criteria**

DCP does not interfere with core operations on your hardware. You can set it to use only a portion of idle capacity, work only during certain hours, or both.

* **Control by Preference**

You can choose which applications to contribute compute power to. Favor computational research from a certain institution, only process Canadian datasets, and more.

* **Local Payouts**

DCP measures compute in currency. You can use the value that is accumulated to accelerate your own work on the Distributed Computer, or request a payout to bank or virtual wallet.

## For Enterprise

**DCP For Enterprise**

The Distributed Compute Protocol provides many benefits for future-thinking businesses:

* Serverless infrastructure to greatly reduce OpEx.
* Secure on-premises computing.
* Powerful no-code applications for non-developers to use.
* Complete system mobility and zero cloud lock-in.

Kings Distributed Systems Ltd. is the official partner of DCL that deals with all enterprise users of DCP. Their team provides the dedicated support and knowledge to help achieve your business goals.

Daily business operations are increasingly impacted by the need for large amounts of compute. With OpEx and data security concerns growing, the current cloud architecture may be unsustainable.

To reduce this uncertainty and provide abundant compute resources, KDS builds enterprise-grade networks built on DCP. Its mission is to allow business of any size to meet increasing processing requirements while keeping their data secure.

## Features

**Feature Spotlight**

DCP offers flexible ways to perform high-throughput computing. Built on proven technology, its many features provide performance, security, and affordability for almost any need.  
  
Learn more about how DCP benefits everything from local clusters to complex deployments.

* **Write once, run anywhere**
* **Effortless migration & compute foraging**
* **Low-code & no-code deployments**

*<Icons to place at the top of the page where you can select the section to go to>*

*A picture containing drawing

Description automatically generated A close up of a sign

Description automatically generated A picture containing drawing

Description automatically generated A picture containing clock

Description automatically generated A picture containing clock, sign

Description automatically generated<New Images>*

**Serverless Compute**

Serverless computing dynamically manages the compute and network resources that your application requires. Instead of you managing the fine details, DCP matches your workload with the right infrastructure.

* **API Centric Framework**

DCP provides a unified API framework for your code to run anywhere, built to be intuitive for all developers. These can be called anywhere from a CL to a website.

* **Run Serverless Anywhere**

DCP lets you run serverless applications on your own hardware, meaning fewer servers you have to manually configure.

* **Zero Lock-In**

Unlike other serverless solutions, DCP is entirely agnostic to the hardware owner. There is no requirement to migrate your data or build for a specific cloud platform.

* **Performance Management**

Adjust your application in real time, so you can scale up or down without restarting work in progress. You can also get fine-grained runtime metrics to gauge and tune performance.

* **Fast Cold Starts**

Hardware is always ready to execute your functions and begins working the instantly it receives instructions. For experienced serverless users, DCP solves the "cold start" problem.

* **Smart Data Parallelism**

Compute.for( ) simplifies your development workflow by automating how work is parallelized and distributed. By coordinating these tricky details, DCP is a point and click solution.

**Data Security**

Not all data is created equal, and some is critical to protect. Therefore, DCP provides new tools for developers to keep data and algorithms safe given the unique nature of its distributed platform.

* **Keep Data Local**

DCP expands your local capacity with secure clusters so you may never need to process in the cloud again. This keeps your data and algorithms within your LAN while benefiting from a serverless architecture.

* **Verified Infrastructure**

For compute-intensive workloads with sensitive code or data, DCP lets you connect to trusted third party sites. Whether your data must remain in a jurisdiction or meet other security requirements, DCP will distribute it accordingly.

* **Secure Overlay Network**

Protect data with secure SSH tunneling between servers when scaling beyond your LAN. Further, DCP implements cryptographic signatures on every message for tracking and nonrepudiation purposes.

* **Secure Code Execution**

By executing in a secure sandbox within the V8 JavaScript engine, code cannot interfere with the underlying device. This protects against XSS and CSRF vulnerabilities, and ensures workloads are completely isolated.

* **Client-Side Verification**

In addition to standard password and IAM methods, DCP enables Workers to request special identifiers from people accessing their hardware. This protects compute nodes from unauthorized users and applications.

* **Homomorphic Encryption**

DCP will implement native support for computing on encrypted data using the Microsoft SEAL library. Developers will be able to specify an appropriate level of encryption even though it may degrade compute performance.

**Application Ecosystem**

Platforms have revolutionized the world, and many provide support for user made applications. DCP fully supports this, allowing users to build, share, and use 3rd party apps from finance to AI.

* **Engaging User Interface**

DCP apps can take on almost every frontend imaginable, from beautiful HTML5 pages to mobile apps. Your users can run advanced computations without a single line of code.

* **Interactive Computing**

Build parameters into your apps that can be adapted for different experiments and datasets. DCP apps also allows others to back-check computations for effective collaboration.

* **Seamless Backend**

Every app integrates with the DCP compute platform, whether on-premises or in the cloud. DCP also implements a deterministic math engine so results are identical across operating systems.

* **Installation-Free**

Apps are accessible on any device and require no download or installation. In the case of web apps, getting your audience set up is as easy as sharing a URL.

* **Potential to Monetize**

Apps can be free to use, require a license, or require payment based on compute consumption. Developers can seamlessly monetize their area of interest and make significant income.

* **Largest Dev Community**

DCP is built on JavaScript, with native support for NPM. With over 1m packages, the largest software registry in the world is ready to be used in your app.

**Complete Distributed OS**

Distributed computing must improve the performance of jobs while minimizing overhead. DCP solves the tough problems to make any collection of computers greater than the sum of its parts.

* **Optimized Scheduling**

DCP's algorithms automatically optimize time- and space-sharing allocations for different jobs. This dynamically updates as jobs are added or change nature.

* **Confirms Work Profiles**

Easily estimate the size and compute requirements of your workload to accurately estimate its profile. DCP cross-validates results from different Workers for a high degree of confidence.

* **Arbitrary Compute Grids**

Build and manage your own personal compute cluster with spare servers and IoT devices. Access to these grids can be given on a permissioned basis, and they may be sub-divided into hyper-local networks.

* **Highly Elastic**

Scavenge compute and memory from any machine with control over the time of day, type of application, and more. When resources are required locally, DCP instantly scales back.

* **Manages Network Traffic**

The network is finally homogeneous with DCP. From 5G and SSH to Ethernet and open WiFi connections, DCP's overlay network accounts for differences in bandwidth and latency.

**Flexible Deployments**

Developer needing control over how and where their applications are run. Therefore, DCP lets you simply manage how your code is executed under the same serverless framework.

* **Hybrid Deployments**

Combine local Compute Groups on your LAN with ones in the cloud. DCP allows you to control remote nodes as easily as your local computer. You can even try different combinations with just a few clicks.

* **Flexible Edge Computing**

Process data close to the source. Anything from a smart home monitor to a connected vehicle can run your code, giving you almost infinite choice to build an edge-ready cluster.

* **High Fault Tolerance**

Workloads adapt in real-time to the underlying compute nodes. This allows work to shift around when a node drops a connection, while livestreamed results mean a lost node does not mean lost time.

* **Lightweight Footprint**

DCP is an ultra-lightweight framework with resting memory requirements as low as 0.1% of popular Container and VM services. Nodes can join and leave DCP clusters in an instant.

* **Microservice Enabled**

Offload computationally-intensive but stateless work onto DCP through your other applications. The API is built to be used in loosely coupled systems while adding almost no overhead.

* **Deploys Anywhere**

Manage DCP with popular deployment options like Kubernetes, Containers, and VMs. DCP can scavenge unused compute from any hardware virtualization image, even when fully dedicated.

DCP is a unique tool for data parallel workloads of all shapes and sizes. Built by and for developers, its platform provides abundant compute from devices all around you.

The future of distributed computing is compute.for( ).

## For Research

**Accelerate Your**

**> Bioinformatics**

**> Computational Neuroscience**

**> Deep Neural Networks**

**> Astrophysics**

**> Weather Mapping**

**> Fourier Transformations**

**> Differential Equations**

**> Markov Processes**

Many researchers have already accelerated their work with thousands of additional core hours, and you can too.

Whatever your requirements, DCP is a friendly way to deploy your code as fast as possible. It reduces time spent writing tedious code, and lets you focus more on experimentation. Plus, the community is there to help you every step of the way!

**Global Research Runs on DCP**

**Researchers Deserve Better, That’s Why DCP Was Made**

Whether you are a college student or an international taskforce, DCP makes collaborating on compute easier than ever before. Besides speeding up your workloads, there are multiple benefits to using DCP:

**** *<New Image>*

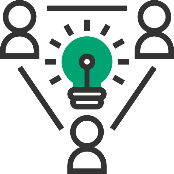
**Interactive Computing**

Share your research as gorgeous websites with interactive parameters and native compute. Compatible with CSS themes and more, making science interactive has never been this easy.

 *<New Image>*

**Computational Peer Reviewing**

DCP apps implement deterministic math libraries. For the first time, your audience can back-check computational results and get the same answer regardless of their operating system.

 *<New Image>*

**Dynamic Resource Sharing**

Share your underutilized compute with colleagues around the world, or leverage open university grids. Leading institutions are connecting machines with the Distributed Computer to accelerate science & innovation.

The first user of DCP, Dr. Daniel Desjardins, had compute-heavy research requirements in electrodynamics. His work involved differential equations and mathematical solvers.  
  
Listen to Dr. Desjardins explain how the protocol has accelerated his research 100x compared to mainstream tools.

**Innovation, Your Way**

DCP is ideal for the kinds of data parallel workloads common in research computing today, such as:

 *<New Image>*

Artificial Intelligence

The compute needed for AI is doubling every 3.5 months. Fortunately, most of the fundamentals like hyperparameter searching are parallel and can be turbocharged with DCP.

 *<New Image>*

Edge Computing

DCP is completely device agnostic, making it the ideal tool for lightweight IoT deployments. It can make use of compute on any network, including 5G.

 *<New Image>*

Bioinformatics

From BLAST searches and genetic algorithms to protein folding and genome sequencing, DCP cuts down the time to novel insights.



Computational Finance

From securities analysis to tracking the market, massive compute power can be an invaluable tool in financial analysis. DCP accelerates it all.



Computational Physics

Many of the most common methods in mathematics are well suited to parallel execution, from finite element analysis to partial differential equations.

 *<New Image>*

Simulation & Modelling

Monte Carlo analysis is a perfect application for DCP, as are other stochastic simulation methods. Many more types can be modelled and run as parallel components.

**A Supercomputer Programmed Like a Laptop**

Computational science is tough enough without having to manage Containers and VMs The Compute.for( ) function abstracts away the tedious parts, so you only have to worry about your own code.  
  
Unlike other platforms, DCP lets you spend less time worrying about technical difficulties and more time making breakthrough discoveries!

**DCL Compute Grants**

Unfortunately, many people with brilliant ideas for cutting edge research are held back by a lack of resources. Distributed Compute Labs works with its partners to scavenge compute and donate it to ambitious projects.  
  
As a community, DCL wants to see a world where no good idea has to be scaled back because of a lack of hardware. If you need help, contact us today!